An IoT Platform for Occupancy Prediction using Support Vector Machine

By

Alec Parise

BScE. Geomatics Engineering, University of New Brunswick Fredericton, 2017

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Master of Science in Engineering

in the Graduate Academic Unit of Geodesy and Geomatics Engineering

Supervisor: Monica Wachowicz, PhD, Geodesy and Geomatics Eng.

Examining Board: Shabnam Jabari, PhD, Geodesy and Geomatics Eng.

Jennifer McArthur, PhD Architectural Science,
Ryerson University

This thesis is accepted by the
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

November, 2019

ABSTRACT

The Internet of Things (IoT) is a network of devices able to connect, interact and exchange data without human intervention. Most of today's research focuses on collecting indoor sensor data with the purpose of reducing costs of operation facilities management. Innovative approaches ranging from context aware sensing platforms to dynamic robot sensing have been proposed in previous research work, but the challenge remains on understanding how sensor data can be used to predict occupancy usage patterns in smart buildings. This research aims at developing a non-intrusive sensing method for gathering sensor data for predicting occupancy usage patterns in indoor environments. There are several potential applications ranging from that can benefit from occupancy prediction. Smart building management systems; establishing communication with the HVAC system when an accurate occupancy classification and prediction for optimization of energy consumption. Towards this end, an IoT platform based on an open source architecture consisting of Arduino and Raspberry Pi 3 B+ is designed and deployed in three different environments at two University campuses. By utilizing temperature and humidity for observing indoor environmental characteristics while combining PIR motion sensors, CO2, and sound detectors a robust occupancy detection model is created, and by applying Support Vector Machine, occupancy usage patterns are predicted. This IoT platform is a low-cost and highly scalable both in terms of the variety of on-board sensors and portability of the sensor nodes, which makes it well suited for multiple applications related to occupancy usage and environmental monitoring.

DEDICATION

I am dedicating this thesis to four beloved people who have meant and continue to mean so much to me. Although my grandfather, Camille Parise, who is no longer of this world, his memories continue to regulate my life and has taught me value of hard work. Next, I would like to thank my grandmother, Thelma Parise who has taught me the virtue of compassion and had showed the importance of enjoying the little things in life. I also want to thank my father, Terry Arsenault who has shown his support and pushed me to become a better version of myself. Finally, I want to extend my gratitude and appreciation to my mother, Rachel Parise, her strength, empathy for others and selflessness has made a tremendous impact on not only my life but has inspired the lives of many.

ACKNOWLEDGEMENTS

I would like to thank the all of the people involved in this project and that ultimately made this a reality. I would like to extend my gratitude specifically to Miguel Ángel Manso Callejo who was a key contributor for programming the sensors and other hardware components. He continued to dedicate his time and effort to answer the many questions or concerns I had when given the task to program my own sensors. Secondly, I would like to thank Marco Mendonça who aided in programming the Raspberry Pi and was a tremendous help for sensor deployment and monitoring the sensor data. I would also like to thank Hung Cao. Throughout this research he was an amazing mentor and played a large role for managing the cloud which served as the repository for the sensor data. I would like to express my gratitude to the ITS department that approved the deployment all of the sensors and allowed me to continue with my experiment. In addition, I would like to thank my fellow PhD, MSc. Students, and RA's for their feedback, cooperation and of course friendship. I am also grateful for the support, mentorship and valuable experiences that Monica Wachowicz provided for me as my supervisor. I was uncertain about my place at the People in Motion Lab but was welcome with open arms and gained the confidence with my academic ability and made an amazing group of friends along the way.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Indoor sensor network technologies have been a relevant research topic in academia for decades ranging from designing short range network infrastructures for data transmission and sensor APIs for performing common programming tasks to indoor localization, occupancy detection, ambient intelligence, and context-aware services (Ferdoush, 2014; Lim, 2007; Anastasi, 2003). With advances in technologies such as the Internet of Things, indoor sensor networks are enabling devices to communicate any sort of" sensor data" through network carriers. In particular, the Internet of Things (IoT) aims to establish interoperability and interconnectivity among sensors to enhance human experience, especially those pertaining to physical contexts (e.g., home, office, and workshop) and the amenities contained within (Laput, 2017). The purpose of developing an IoT platform is to create an interoperable sensing environment able to establish communication between devices (i.e. many sensors that are needed to collect data) having different data rates, latency and bandwidth requirements. However, commercially available sensors are usually expensive and data access is restricted to the companies' cloud and network infrastructures, which hamper their use due to their lack of interoperability.

Current research in IoT platforms has pointed out an emerging interest in occupancy detection for Building Management Systems (BMS). Canada's built environment consists of approximately 14.1 million households and 482,000 commercials or institutional buildings. In 2018, buildings in Canada accounted for 12% of total greenhouse gas (GHG) emissions, or 17% if emissions from generating the

electricity used in buildings are included (Schulte, 2018). The percentage of Canada's GHG emissions from the building sector has remained approximately the same since 1990. As Canada pursues its COP21 commitment to reduce greenhouse gas emissions to below 30% by 2030, there is a critical need not only to ensure that new buildings are designed to minimize energy consumption, but to address the performance of the much larger area of existing buildings.

The central task for efficient energy management is to reduce the energy consumption while not impeding on the comfort of the occupants. This may include the temperature, lightning and air quality. According to Ortega et. al (2015) Current BMS, especially those related to building efficiency, are usually operated based on a fixed seasonal schedule, maximum design occupancy assumption, and pre-defined comfort levels (constant) to ensure satisfactory temperatures, ventilations and luminance at all times; but fail to capture dynamic information. This is both costly and inefficient. For example, a lighting system would operate a fixed 8-hour schedule expecting that the occupant will be present through all that period (Ortega et al. 2015).

Several occupancy detection models have been proposed in the literature for lighting and heating control systems and optimization of HVAC systems that were based on a relatively small number of mathematical models such as Markov Chain, genetic algorithms, and fuzzy inference algorithms (Ryo et al. 2016; Wang et al. 2018). However, previous occupancy detection models have focused on monitoring extensive simulation and scenario derived data, rather than real-world sensor data, mainly because they require costly platforms to monitor occupancy or because they inaccurately monitor the occupancy due to irregular historical patterns.

Therefore, this thesis proposes an IoT platform that has the ability to efficiently collect time series as well as event-triggered sensor data obtained from a variety of on-board sensors measuring environmental information for occupancy prediction. Towards this end, the Support Vector Machine (SVM) is used to build a predictive model since it is a well-known method developed by Vladimir Vapnik and Corinna Cortes in 1989 and it has been applied to many applications including classification of daily activities based on health sensor data (Fleury, 2010). SVM is a non-probabilistic binary classifier that relies on separating features via a hyperplane and later classifies those features based on the distance of the maximum margins of the hyperplane.

My empirical approach consists of deploying an open source IoT platform capable of streaming non-intrusive data in near real-time to a private cloud service where the data is to be stored and then analyzed. By running an SVM prediction model in this IoT platform, it is possible to train the models by collecting sensor data based on time series and event-triggered data in an indoor environment. Combining time series and event-triggered data can be used to detect trends and outliers, but it can also be used to recognize disruptions when an event occurs do to, for example, an occupant is entering a room or a lecture is happening in the same room.

The data is collected from PIR (Passive Infrared) motion, temperature, CO2 concentrations, humidity, sound detection and luminosity sensors, all of which combined generates sensor data to be handled by a five-layer architecture of an IoT platform. The layers are: Sensing Layer, Edge Access and Processing Layer, Network Layer, Cloud Access and Processing Layer, and Application Layer. Three experiments were conducted using four general purpose sensing units distributed in an indoor environment. Each

environment was selected to encompass three different occupancy patterns, controlled, semi-controlled and completely and will hypothetically yield different diverse results, which will be used to evaluate the accuracy of the proposed predictive model.

There are several potential applications ranging from that can benefit from occupancy prediction. Smart building management systems; establishing communication with the HVAC system when an accurate occupancy classification and prediction for optimization of energy consumption. There are also similar location-aware services that business can also optimize based on detecting levels of occupancy. Occupancy classification and prediction can also play a key component for emergency response operations where a crowd needs to be directed for evacuation purposes.

1.2. Scientific contributions

The main scientific contribution of this research work is to contribute with empirical evidence for understanding how IoT platforms should be developed for predicting indoor occupancy using non-intrusive sensing technologies. This research also paves the way in avoiding simulated data and mathematical models because real-world data-driven models such as SVM predictive models have the potential of improving our current understanding on occupancy patterns in indoor environments.

.

1.3 Research Questions and Objectives

The overall research goal is to develop and deploy an open source, non-intrusive IoT sensor platform capable of collecting sensor data that will be used to predict occupancy using a computationally inexpensive SVM algorithm This in turn will be applicable in the future for enhancing living experiences, promote healthier environments and reduce energy consumption.

The research questions of this research work are:

1. Would a single sensor or a set of sensors be more adequate to detect occupancy meanwhile preserving privacy?
2. What is the expected accuracy of SVM prediction models which are incorporated into IoT platforms?
3. Would validation of the proposed SVM model have to include alternative, more intrusive sensing units?

The research objectives are:

1. Select a number of sensors that can be used to indoor occupancy detection, aiming at retaining the privacy of occupants by not relying on intrusive information and mobile devices.
2. Develop a generic sensing approach for measuring the dynamic changes caused by occupancy.

3. Develop a communication protocol that can stream data in near real time to the cloud.
4. Study the best location of the sensor nodes in a room. Apply SVM for classifying and predicting indoor occupancy.
5. Evaluate the results.

1.4. Organization of this Thesis

This thesis is organized in five chapters as described as follows:

- Chapter 2 provides a background of IoT devices along with a hardware comparison describing available sensors and which sensors were chosen for the experiments.
- Chapter 3 summarizes the related literature and impacts for the decisions in terms of algorithms and sensors used in this thesis.
- Chapter 4 describes the methodology including the IoT platform and the proposed SVM prediction model.
- Chapter 5 provides an in-depth analysis of the results and the accuracy of the proposed approach..
- Chapter 6 outlines the conclusion and future work research work.

CHAPTER 2: BACKGROUND

The International Telecommunication Union (ITU) defines the IoT as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies (Djelouat, 2018). The primary components of IoT consists of sensors, gateways, communication protocols and data processing technologies that interact with each other to provide a specific application. For instance, the applications pertain to, experimental monitoring, automated industry, connected healthcare, smart buildings, Intelligent Transportation Systems (ITS), and smart grid (Djelouat, 2018). IoT platforms have evolved rapidly in the last decade by leveraging the interactions among different types of sensors in smart devices, such as vehicles, medical devices, cameras and RFID systems. The continuously expanding sensing and hardware components available in IoT platforms demand considerations regarding the appropriate selection of sensors. Off-of-the-shelf sensors from suppliers such as Estimote and Texas Instrument were avoided in this thesis as they restrict data access and do not allow for the interoperability of their sensors into other IoT platforms.

This Chapter compares a variety of available sensors which can be used for the selection of the sensors for an IoT platform. To address the security of the data centric communication a description is given regarding the network communication protocol which integrates a secure network to publish and subscribe data tuples using Wi-Fi. Finally, cloud computing is introduced as it is a computational component for performing the SVM analytical tasks in an IoT platform.

2.1 Sensors

Three main criteria are usually used for selecting a sensor. They are accuracy, range and cost. Table 1 describes each sensor based on these criteria.

Table 1: Comparison of popular sensors used in indoor IoT.

Sensor	*Accuracy*	*Range*	*Cost (Excluding Shipping)*
Temperature Sensors			
DHT11	+/-2 degrees	0 to 50 degrees Celsius	$12.35
DTH22	+/-0.5 degrees	-40 to 125 degrees Celsius	$12.99
Motion Sensors			
AM313	+/- 1.5 m	3 m	$11.99
Robojax HC-SR501	+/- 2 m	7 m	$8.25
CO2 Sensors			
CCS811	N/A	CO2: 400 to 60,000 (ppm)/ TVOC: 0 to 60,000 (ppb)	$20.95
Adafruit SGP30	N/A	CO2: 400 to 60,000 (ppm)/ TVOC: 0 to 60,000 (ppb)	$19.95
Luminosity Sensors			
Robojax Photoresistor	+/- 10 lux	0-1000 lux	$9.87
SunFounder Photoresistor	+/- 10 lux	0-1000 lux	$9.99
Sound Sensors			
SparkFun	N/A	50Hz – 20Khz	$11.95
Sound Detector B00TM7R21W	N/A	N/A	$6.99

These sensors can be deployed for measuring indoor environmental characteristics such as: ambient temperature, humidity, luminosity, motion using a passive infrared (PIR), and CO2.

Temperature and Humidity: DHT22 Digital Input Sensor

The DHT22 sensor is capable of measuring temperatures ranging from -40 to +125 (+/- 0.5) degrees Celsius and is also efficient in measuring humidity ranging from 0 to 100%. Even though this sensor is more expensive than its predecessor, the DHT11 senor, it is substantially more accurate and has significant temperature range improvements; the DHT11 sensor ranges from 0 and 50 (+/-2) degrees Celsius. The vast ranges in temperature and humidity along with a 0.5 Hz sampling rate permits the DHT22 to quickly detect small or large discrepancies, which, in an indoor environment is advantageous for potentially detecting a fire to ensure emergency protocol. Further specification comparisons between the two sensors can be seen in Table 2.

To measure temperature this sensor utilizes an NTC "Negative Temperature Coefficient" (which means the resistance decreases with the increase of temperature) thermistor. The thermistor is a variable resistor that changes its resistance as modest temperature changes occur, therefore allowing it to accurately detect minute temperature variations. The ambient humidity is measured by two electrodes with moisture holding substrate between them. As the humidity diversifies, the conductivity of the substrate creates a resistance between these electrodes that permits resistance measurement, later processed by the NodeMCU (Dejan, 2016).

Table 2: A comparison between the DHT11 sensor and currently used DHT22 sensor (Dejan, 2016).

DHT11		DHT22
0 - 50°C / ± 2°C	*Temperature Range*	-40 - 125 °C / ± 0.5 °C
20 - 80% / ± 5%	*Humidity Range*	0 - 100 % / ± 2-5%
1Hz (one reading every second)	*Sampling Rate*	0.5 Hz (one reading every two seconds)
15.5mm x 12mm x 5.5mm	*Body Size*	15.1mm x 25mm x 7.7mm
3 - 5V	*Operating Voltage*	3 - 5V
2.5mA	*Max Current During Measuring*	2.5mA

Robojax HC SR501

The HC SR501 PIR motion sensor is equipped with digital intelligent infrared technology capable of detecting motion with a range of 7 meters. Internally, this device has a 2 second time delay between events to reduce the impacts of small triggers. Another consideration is that this sensor must be placed away from any conventional heat source or window because false triggers may occur due to the sensitivity of the device. On the other hand, an advantage of having a high sensitivity, is that when placing the PIR motion sensor in proximity of a doorway it can accurately depict entry or exit of a room. These events are important when determining the correlation between occupancy triggers and how the ambient environment is affected upon entry and exit (Sunrom, 2018). The specifications of the Robojax HC SR501 PIR motion sensor can be seen in Table 3.

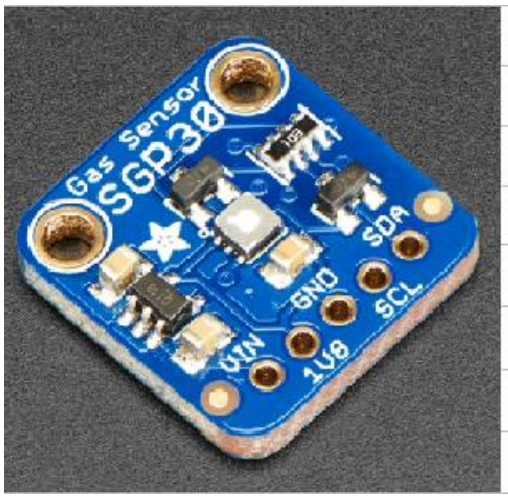
Table 3: Specifications for the Robojax PIR motion sensor (Sunrom, 2018).

Operating power: 3.3V
Range adjustment: 3m – 7m
Sensing range: 120 degree, within 7 meters
Temperature: – 15 ~ +70
Delay time: Adjustable (.3->5min)

SGP30 CO2 Air Quality Sensor

This sensor from AMS is a gas sensor that can detect a wide range of Volatile Organic Compounds (VOCs) and is intended for indoor air quality monitoring. When connected to the NodeMCU it returns a Total Volatile Organic Compound (TVOC) reading and an equivalent carbon dioxide reading (eCO2). The CCS811 sensor can measure the 400 to 60,000 parts per million (ppm), and TVOC (Total Volatile Organic Compound) concentration within a range of 0 to 60,000 parts per billion (ppb).More specification can be seen in Table 4. One of the main purposes of this sensor is to collect data on indoor air quality, but more specifically monitor the indoor CO2 levels for interpreting densely occupied rooms (Adafruit, 2018).

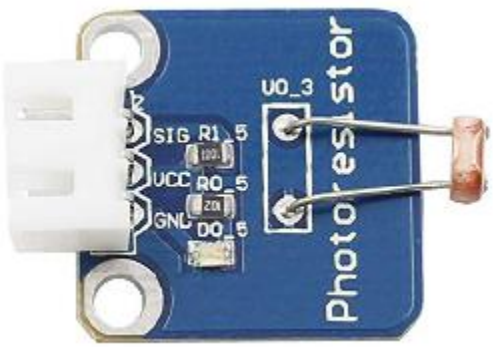
Table 4: Specification for SGP30 CO2 Air Quality Sensor.

	Total Volatile Organic Compound (TVOC) sensing from 0 – 1,187 ppb
	eCo2 sensing from 400 – 8,192 ppm
	Fiver operating Modules
	Integrated MCU
	Onboard processing
	Standard I2C Digital interface
	Optimizes Low-Power Modes
	Optional NTC Thermistor Pins

Sunfounder Photoresistor Module: Luminosity sensor

The Sunfounder Photoresistor module is an analogous sensing unit capable of measuring room light intensity using a photoresistor and a 10 kΩ in-line resistor. This sensor is used to determine when the lights in a room are on or off. However, an important consideration should be the impedance sunlight may have on the accuracy of the sensor and if thresholds can mitigate the effects of natural light. . Table 5 summarizes its main characteristics.

Table 5: Specifications for the Sunfounder Photoresistor Module.



Drive Voltage: 3.3 V
Analog Output
Operating Temperature: -40 to 85 Degrees Celsius
Model: Sunfounder
On Board LED

Sparkfun Object Sound Detector

The KY-038 object sound sensor has three main components on its circuit board. First, the sensor unit at the front of the module measures the area physically and sends an analog signal to the second unit, the amplifier. The amplifier amplifies the signal, according to the resistant value of the potentiometer, and sends the signal to the analog output of the module. The third component is a comparator which switches the digital out and the LED if the signal falls under a specific value (SensorKitSupport, 2018).

Essentially the digital output is programmable threshold that you can control by adjusting the potentiometer. The frequency response for this sensor can range from 50Hz – 20 kHz, however the data does not show absolute values like the temperature sensor, therefore it does not have the capability of voice recognition. This feature is critical for retaining user privacy, but also provides context by means of number of event triggers per period. Table 6 shows the specifications for this sensor.

Table 6: Sparkfun Sound detection specifications.

Adopt LM393 Main Chip
Electric Condenser Microphone
Features Single Channel signal output
Low lever output signal used for sound control light
Working voltage: DC 4-6V
Frequency Range: 50Hz – 20Khz

2.2 NodeMCU

Consisting of 64kB of instruction RAM, 96 kB of data RAM, the NodeMCU is able to quickly process sensor data by utilizing an event driven Nodejs Network API. Wi-Fi connectivity is established through a low-cost Wi-Fi ESP8266 chip which adheres to the IEEE 802.11 b/g/n protocol, a widely used wireless computer networking standard. Even though the NodeMCU requires a constant 5V power supply, it is still a versatile spatial context measurement unit that can be displaced in any room or environment that has an outlet and Wi-Fi connectivity. Full specifications of the NodeMCU can be seen in Table 7. The GPIO (General Purpose Input/Output) can be configured as either Input or Output for execution time. By configuring the GPIO as input it allows us to read the state of each sensor, and output authorizes sensor configuration (NodeMCU, 2018).

Table 7: NodeMCU specifications

	Specifications
	•32-bits RISC CPU 80MHz
	•64KB of instruction RAM, 96 KB of data RAM
	•4MB Flash
	•IEEE 802.11 b/g/n WiFi
	•13 GIPO pins (only 11 available in the board)
	•SPI, I2C
	•Only 1 10-bit ADC (0-1V)
	•Pins work with 0 - 3,3 V
	•CP2012 USB or CH340G (without patent)
	•Can be programmed with Lua, C/C++, Python, JavaScript, Arduino IDE
	•Micro USB (5V power supply)

2.3 Raspberry Pi 3 B+ Gateway

The Raspberry pi 3 B+ is the latest product in the Raspberry pi 3 range. It includes dual-band 802.11ac Wi-Fi, meaning it offers the versatility of 2.4GHz and 5GHz wireless communication and has a 1.4GHz 64-bit quad-core processor—a 200MHz improvement over the 1.2GHz processor in the Pi 3 Model B. The new board is also equipped with the latest version of Bluetooth (Bluetooth 4.2), 4 USB ports, a Gigabit Ethernet port (maximum throughput 300 Mbps), a full-size HDMI for efficient desktop compatibility, and requires 5V/2.5 DC power input. The increased processing power, Wi-Fi protocols, lower power input requirement and programming versatility of the Raspberry pi 3 B+ made it a preferred gateway over previous Raspberry Pi models and the Arduino Uno Rev3. Table 8 provides the specifications of each gateway (Maker.IO,

2018). Both gateways have been known to have time delays if the internal clocks are not synchronized. To bypass time delays of tuple, which would equate to approximately 1 second every 20 seconds, the gateway will be synchronized to a UTC online time clock to ensure that it is not susceptible to drifts.

Table 8: Specifications of the Raspberry Pi 3 B+ and Arduino Uno

	Arduino Uno	Raspberry Pi Model B+
Price	$30	$35
Size	7.6 x 1.9 x 6.4 cm	8.5 x 5.6 x 1.7 cm
Memory	0.002MB	512MB
GPIO	14	40
Clock Speed	16 MHz	700 MHz
On Board Network	None	10/100 BaseT Ethernet socket
Multitasking	No	Yes
Input voltage	7 to 12 V	5 V
Flash memory	32KB	Micro SD card
USB	One, input only	Four, peripherals OK
Operating System	None	Linux distributions
Integrated Development Environment	Arduino IDE	Scratch, IDLE, anything with Linux support

2.4 Wireless Networks

There are several existing communication networks available, such as: Wi-Fi Bluetooth low energy and ZigBee. This section outlines the specifications of each network and its feasibility for an indoor environment. Bluetooth Low Energy (BLE) is often referred to as a personal network, therefore its range is much shorter than both ZigBee and Wi-Fi. However, there is a tradeoff in terms of data rate, since BLE has a much higher data rate than ZigBee. Traditional BLE has a data rate of 1 Mb/s for short bursts, compared to ZigBee which has a data rate of 250 kb/s. Wi-Fi on the other hand

also has a high data rate but much of its other parameters are undefined because it depends on the locations of the access points within a building. Usually if there is wide coverage in the building more often Wi-Fi is the preferred solution. Wi-Fi is capable of reaching up to 250 Mb/s if there is a direct connection to the Internet, and it has a three-channel bandwidth of 2.4, 3.6 & 5 GHz. ZigBee also has a bandwidth of 2.4GHz, 915MHz & 868 MHz, and would be useful for applications where there is sparse Wi-Fi connectivity.

Below Table 9 provides a comparison between existing networks based on their primary specification entailing: range, power, latency, bandwidth and transmission rate. The range of each sensor is arbitrary and is completely dependent on obstruction, such as walls ceilings, it is not an absolute range, simply an estimated range. The power of the device reflects its consumption. BLE is the most energy efficient, however is limited in terms of capability for a larger scale indoor IoT platform. Latency reflects the delay between data tuples. Bandwidth refers to the range of frequencies available for transmitting data. Lastly the transition rate is how quickly that signal can transmit the data tuples.

Table 9: Comparison between wireless LAN, Bluetooth and Zigbee. Source: Shahzad et al. (2014)

Standard	ZigBee	BLE	Wi-Fi
IEEE Specs	802.15.4	802.15.1	802.11 b/g/n
Frequency spectrum	868/915 MHz; 2.4 GHz	2.4 GHz	2.4 GHz, 5 GHz
Topology	Star, mesh, cluster tree	Star, point-to-point	Start, point-to-point
Network size	65536	Not defined	32
Data rate (Mbsps)	0.02 – 0.25	1	11/ 54/600
System resources	4 kB – 32 kB		1 MB+
Range (m)	< 100	< 50	< 100
Number of channels	1/10/16	40	11-14 (3 orthogonal)
Security	128-AES	128-AES	SSID

2.5 MQTT Protocol

MQTT is a machine-to-machine (M2M) connectivity protocol, designed to publish/subscribe messaging transport which contains packets of data. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium. For example, it has been used in sensors communicating to a broker via Raspberry Pi, over Wi-Fi connections within facilities management or healthcare scenarios, and in a range of home automation and small device scenarios (HiveMQ, 2019). To further elaborate, MQTT consists of Clients and Brokers. An MQTT client consists of publisher/subscriber labels that refer to whether the client is currently publishing messages or subscribing to messages. In other words, an MQTT client is any device (from a micro controller up to a full-fledged server) that runs an MQTT library and connects to an MQTT broker over a network (HiveMQ, 2019). An MQTT Broker is the counterpart of the client and is essential for any publish/subscribe

protocol. The broker is responsible for receiving all messages, filtering the messages, determining who is subscribed to each message, and sending the message to these subscribed clients. Therefore, it is important that any broker to be highly scalable, and easy to monitor when multiple sensor networks are collecting indoor environmental data over multiple levels (HiveMQ, 2019). Some of the most scalable and robust open source MQTT brokers are Mosquito, and HIVEMQ.

2.6 Computing Environments

According to the international Data Corporation (IDC), the amount of digital data generated surpasses 1 zettabyte in 2010, and 2.5 exabytes of new data is generated each day since 2012 (Yousefpour et al., 2019). One of the main contributors for the exponential growth of digital data is caused by sensing technologies, specifically IoT. Cisco estimates that there will be around 50 billion connected devices contributing to IoT by 2020, which have the potential of generating massive amounts of data (Evans, 2011). As the data velocity and volume increase, moving the big data from IoT devices to the cloud will create increased bandwidth ultimately causing latency (Yousefpour et al., 2019). Currently, three main paradigms can be found to reduce the latency impacts of big data. They are described as Edge Computing, Fog Computing, and Cloud Computing, shown in Figure 1.

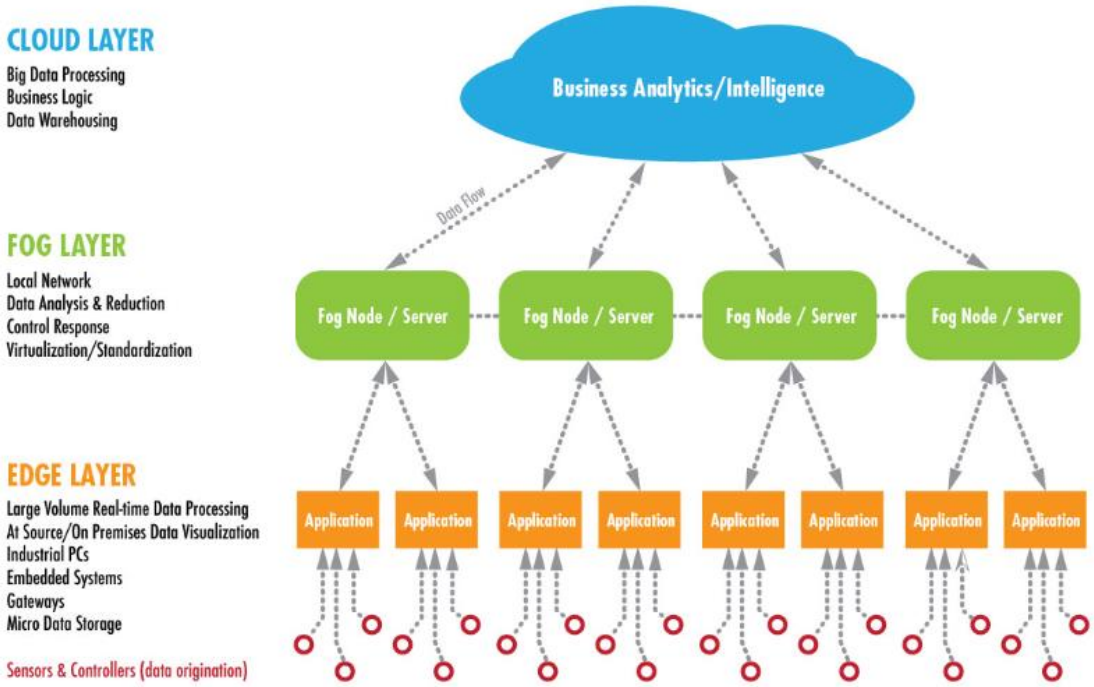


Figure 1: Overview of the computing environments.

Edge computing

Edge computing refers to the enabling technologies allowing computation to be performed at the edge of the network (Shi et al., 2019). For example, edge devices can include gateways which are in proximity of sensors collecting ambient information. Edge computing offers many advantages over traditional architectures such as optimizing resource usage in a cloud-computing system. Performing computations at the edge of the network reduces network traffic, which reduces the risk of a data bottleneck. Edge computing also improves security by encrypting data closer to the network core, while optimizing data that's further from the core for performance (Satyanarayanan, 2017).

Fog Computing

Fog Computing acts as a bridge between edge devices and the cloud. It supports a distributed computing model that provides services at highly geographically distributed fog nodes such as access points, switches, and routers. It is defined as "a scenario where a huge number of heterogeneous (wireless and sometimes autonomous) ubiquitous and decentralized devices communicate and potentially cooperate among them and with the network to perform storage and processing tasks without the intervention of third-parties (Cao, 2018). Fog computing uses edge devices and gateways with the Local Area Network (LAN) to provide processing capability. These devices need to be efficient, meaning they require little power to receive real-time data such as response time (latency), security and data volume, which can be distributed across multiple nodes in a network.

Cloud Computing

Cloud computing is a network of remote servers hosted on the Internet to store, manage and process large amounts of data. Cloud computing is defined by the National Institute of Standards and Technology (US. NIST) as:

"A model for enabling convenient, on demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." (Mell, 2009)

This definition can further describe cloud computing based on five principle characteristics: on demand self-services, heterogeneity of networks, resource pooling, rapid elasticity and measured services.

On demand self-service provides an instantaneous feedback that can avail computing resources (such as CPU time, network storage, software use, and so forth) in an automatic (i.e. convenient, self-serve) fashion.

Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations) (Mell, 2009).

Resource pooling. A cloud service provider's computing resources are 'pooled' together in an effort to provide a secure network for multiple consumers using either the multi-tenancy or the virtualization model. Consumers are not able to tell where their data is going to be stored in the Cloud.

Rapid elasticity: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand (Mell, 2009). To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

Measured Service: Cloud systems automatically control and optimize resource use by leveraging a metering capability1 at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service (Mell, 2009).

Cloud computing offers robust services ranging from cloud architectures, security, and deployment strategies for sensor networks in IoT. It allows for heterogeneous platforms, such as mobile phones and web interfaces to interact with users

and can provide insightful information in terms of energy consumption while prohibiting the traceability of where the data is stored in the cloud. Cloud computing will be an essential component moving forward for developing IoT platforms.

All these three paradigms aim to reduce the latency of sending the data from the edge devices to the core network, ensuring highly efficient network operation and service delivery as well as providing the edge analytics to offload the burden at the core network. But they have few different characteristics that play an important role in the selection of an appropriate computing platform for edge analytics.

CHAPTER 3: LITERATURE

The Internet of Things (IoT) relies on platforms that can orchestrate many IoT devices containing several sensors that utilize the Internet as the backbone of communication to allow a virtual interaction between people and their surrounding IoT devices (Gubbi, 2013). According to Sundmaeker (2010), "things" are active devices in business, information and social processes where they are enabled to interact and communicate among themselves and with the environment by exchanging sensor data about the environment, while reacting autonomously to real-world events and influencing them by triggering actions and creating services with or without direct human intervention.

Many short-range networks exist in today's networking infrastructures to transport sensor data using IoT platforms. Some of the most commonly used ones are Wi-Fi, ZigBee, Wireless HART, Bluetooth Low Energy and 6LoWPAN. There are also a variety of indoor IoT platforms which rely on Wireless Sensing Networks (WSN's) for collecting sensor data (Gungor, 2010). According to Rawat et al. (2014), indoor WSN's have several distinctive features such as availability and easy-to-deploy capabilities that facilitate the widespread use of WSN technology in indoor IoT applications such Building Management Systems (BMS) (Agarwal, 2010; Jin, 2018; Plageras, 2017) and occupancy detection monitoring (Ferdoush, 2014). However, WSNs for large-scale and continuous monitoring still have technical challenges that are raised due to the constraints imposed on the sensor components: limited power, limited communication, bandwidth,

limited processing capacity, small storage capacity, and level of distinctive measure (Gungor, 2010).

Indoor sensor data is being generated by static indoor sensors, remotely controlled robots and wearable sensors. In general, sensing can be categorized as distinctive or non-distinctive sensing for indoor environments. Distinctive sensing requires the sensor to be physically attached to a person in the indoor environment. Non-distinctive sensing, on the other hand relies on monitoring an indoor environment, rather than the person. In this Chapter, an overview of current state-of-the-art in IoT platforms, indoor WSN's for occupancy detection methods, and general purpose sensing for supporting SVM models.

3.1 IoT Platforms

Building IoT platforms requires a deep understanding of the applications requirements for determining which types of sensors are needed, what type of gateways are available, making sure the standards of communication protocols are adopted, and guaranteeing that the data processing power is capable of handling the IoT data streams in order to avoid latency. Therefore, different layered architectures have been previously proposed in the literature to build an IoT platform.

Wu et al. (2010) proposes a five-layer IoT platform that consists of Perception Layer, Processing Layer, Transport Layer, Application Layer and the Business Layer. The main task of the Perception Layer is to retrieve the physical properties of "things" (e.g. temperature and location) using various sensors (e.g. infrared sensors, RFID, 2-D barcode), and convert this information into digital signals which are more convenient for network transmission. The Transport Layer, also known as the Network Layer, is mainly

responsible for transmitting sensor data from the Perception Layer to the Processing Layer through various networks, such as WSN or cable network, as well as the enterprise Local Area Network. The main techniques in this layer include FTTx, 3G, Wi-Fi, Bluetooth, ZigBee, UMB, and infrared technology.

The Processing Layer is mainly used to store, analyze and process large quantities of sensor data and the mass information received from the Perception Layer. For carrying out these tasks, Wu et al. (2010) proposes cloud computing to handle the volume of data. The task of the Application Layer is based on the data processed. The function of this layer is providing all kinds of applications for each industry including intelligent transportation, logistics management, identity authentication, Location-Based Service (LBS), and safety. Lastly, the Business Layer plays the role of a manager of the Internet of Things, including managing the applications, the relevant business models and profit models. The Business Layer not only manages the release and charging of various applications, but also the research on business model and profit model.

In contrast, Liu et al. (2014) considered an IoT platform of four layers, namely Physical (PHY/Mac), Transport, Middleware and Application layers (Figure 2). The PHY/Mac Layer consists of three temperature sensors, one humidity sensor, one alarm, and one camera, all wirelessly connected by ZigBee using a central ZigBee coordinator. Behind the access network (ZigBee based sensor network in this case), a WiMAX subscriber station is deployed in the same room that connected to the ZigBee coordinator via USB cable. The Transport Layer utilizes a software defined radio system to connect and HTTP request/response protocols that can send sensor data from the WiMAX subscriber station to the WiMAX base station. The Middleware Layer (WiMAX base

station) supports device naming, application addressing and profile storage and look-up services in the IoT platform. Towards this end, deviceID, device type and device group profiles can be registered and stored in the National Adaptation Plans (NAPs) repository which allows search services to retrieve a list of devices and device groups with certain geographical, domain, and device name information. This web portal feature is considered to be Application Layer of the proposed IoT platform.

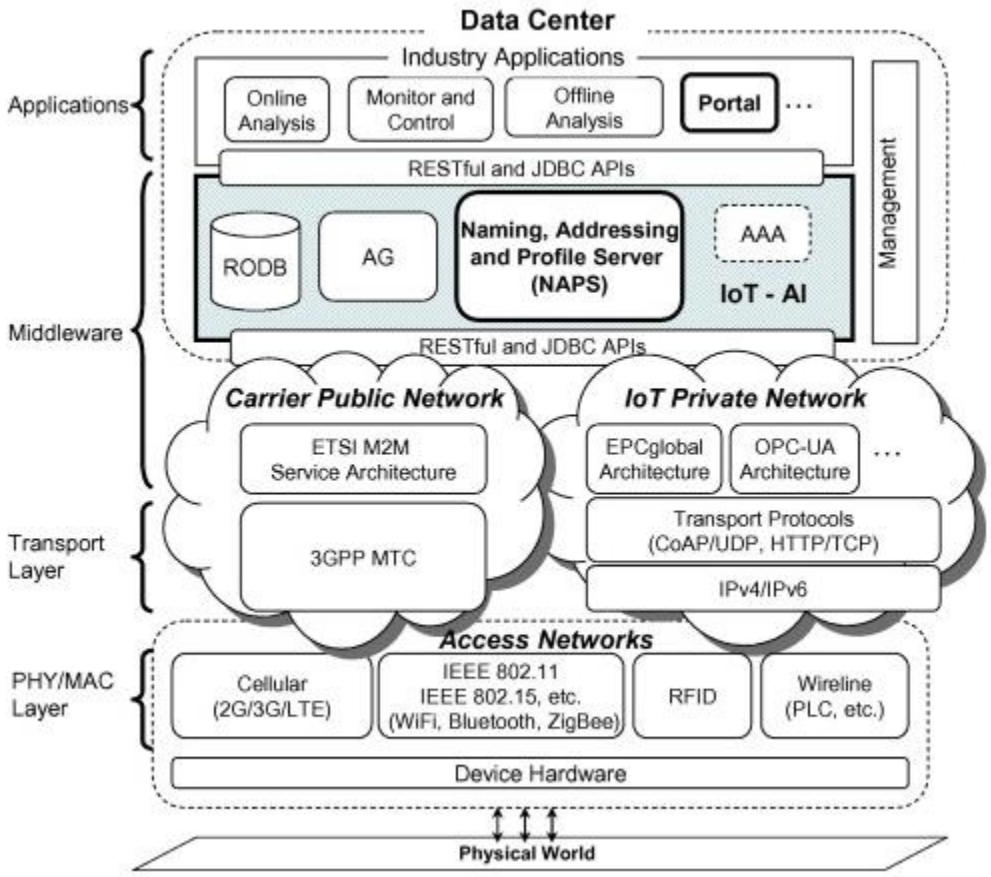


Figure 2: Example of four-layer IoT platform proposed by Liu et al. (2014).

More recently, Djelouat (2018) proposes a three-layer IoT platform consisting of a Sensor Layer, a Processing Layer and an Application layer. The Sensing Layer collects sensor data from temperature and ECG blood pressure sensors and converts the collected

measurements into digital signals. The Processing Layer utilizes edge computing and provides transient data storage for the data received from the sensors and performs local data processing, executes real-time actions, and sends the data to the cloud. Lastly, the Application Layer performs analytical tasks on the information routed from the Processing Layer. Figure 3 provides an overview of the proposed three-layer IoT platform.

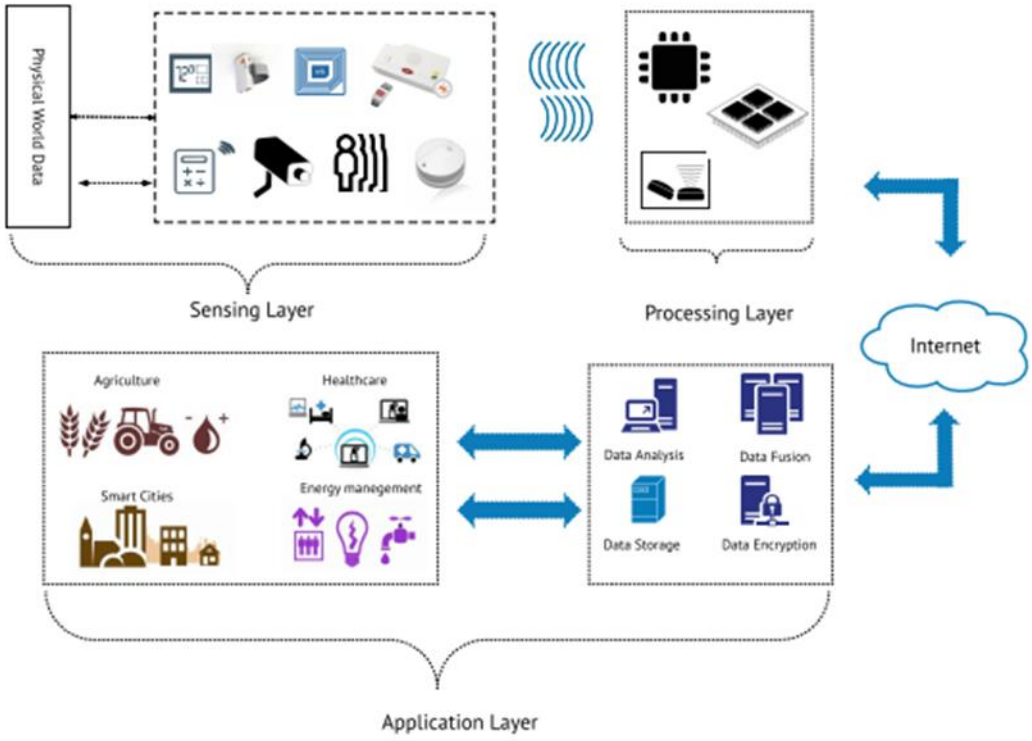


Figure 3: IoT platform architecture (Djelouat, 2018)

This section briefly described three different platforms that illustrate the diversity in IoT. Although theses platforms have different architectural layers, three main components of an IoT platform usually remain the same. They are the sensors, network communication, and cloud storage.

3.2 Occupancy Detection Methods

There are a variety of occupancy detection methods which rely on collecting occupancy information through distinctive or non-distinctive sensing. Huh and Seo (2017) propose a system capable of estimating indoor position by utilizing Bluetooth Low Energy (BLE). More specifically the system implemented in this experiment relies on trilateration techniques of multiple BLE beacons. To estimate indoor position the receiver signal strength index (RSSI) values are recorded from each beacon while occupants moved throughout an open laboratory. This technique also relies on other devices with Bluetooth capabilities such as a laptop of a smartphone. To analyze the data Huh and Seo considered two types of dynamic measurements: acceleration and velocity in the XYZ coordinates. The acceleration and velocity in turn would yield a distance measurement able to estimate the distance a person was from the beacon. By combining the distance measurement from multiple beacons, this method was able to estimate the occupant's location within the laboratory. Because this experiment relied on a mathematical model which used a range-average algorithm for measuring the shortest distance, there are some limitations:

- The measurement results depended on the sample size.
- The sample efficiency of localization depends on sampling speeds and environmental changes.

In addition, this experiment was performed in an open laboratory with few obstructions (i.e. walls or dividers). RSSI values are usually impacted by obstructions, much like any signal, when depending on RSSI values for distance or localization sensors

must be placed with a direct line of sight, otherwise false readings may occur. Therefore, placement of sensors and gateway is a key consideration for optimizing results for an IoT platform.

Depatla et. al (2015) propose a framework for estimating the total amount of people in an indoor and outdoor environment. This framework is based on the integration of Wi-Fi RSSI measurements between a pair of transmitter /receiver antennas. The authors have developed a mathematical model to determine a probabilistic distribution of the receive signal amplitude as a function of the total number of occupants-based estimation using Kullback-Leibler divergence. The results of this experiment show that this technique is applicable for indoor and outdoor environment if there is a clear line of sight between the transmitters and antennas. Wi-Fi based localization can cause inaccuracies when faced with complex indoor environments because the fabric of the environment consists of many obstructions, resulting in noise bursts (Melamed, 2016).

Several efforts can be found in the literature to deal with these resource constraints of indoor WSN. Ferdoush (2010) developed a low cost indoor WSN open source architecture utilizing a Raspberry Pi and an Arduino micro-controller. Temperature and humidity sensors were connected to an Arduino Uno microcontroller which sent time-series data to a Raspberry Pi 3 acting as the base station (Figure 4). As a result of poor Wi-Fi connectivity this experiment used Xbee Wi-Fi modules (conforming to Zigbee networking protocols) as a means for sparse Wi-Fi connectivity and for data transmission. When the data was received by the Base station it was stored locally in a MySQL relational database (RDSMS). This experiment proposes a pragmatic solution for

personalizing an indoor WSN using a Raspberry Pi and an Arduino micro-controller, however, it relies on a limited sensor network for collecting real-world sensor data.

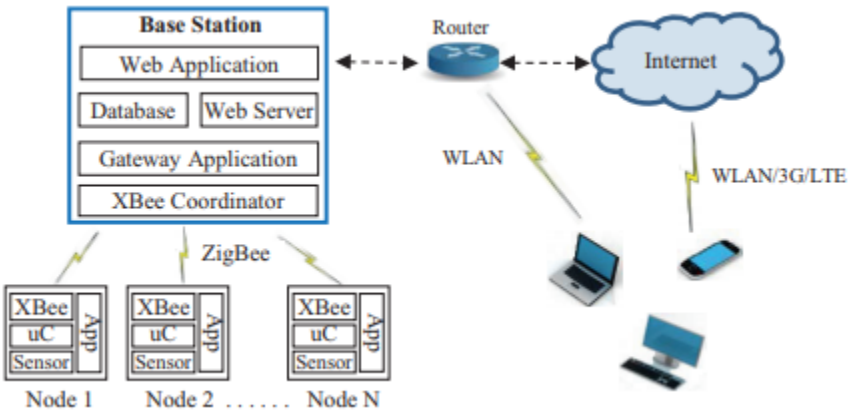


Figure 4: Network architecture from Ferdoush (2010).

More recently, Jin (2018) presents a novel indoor dynamic WSN which collects environmental air quality using a controlled robot. This experiment relies on dynamically monitoring CO2 changes in a building, while hypothesizing that mobile sensor units cover the whole space with higher granularity, therefore enhancing Indoor Environmental Quality (IEQ), such as indoor air quality, thermal comfort, lighting and acoustics. The remotely controlled robot has four functioning sensors measuring: CO2, Organic Volatile Compound (VOC), temperature, luminosity, and a web camera which is directly connected to an Arduino Uno microcontroller that sends and stores sensor data to their cloud service through Wi-Fi. The experiment was conducted in ten different environments in a building for a duration of one week. The sensor data was interpolated using ST binning to reduce the 3-D measurement errors. This was achieved by dividing 3 dimensional images into 3-D cubes based on the spatial temporal resolution. After, global trend analysis was applied to find trends in the data by creating scatter plots and using

traditional regression techniques. Lastly based on residues, a local variation function was used to compute the local variation. The results proved that a dynamic sensing unit is a pragmatic solution for monitoring dynamic indoor environment fluctuations, however consideration needs to be taken for when deploying the robot in a crowd room.

Agarwal (2010) proposed energy management monitoring platform concentrating on occupancy to optimize the HVAC operating schedules. Multiple read switches and the PIR sensors were both connected via WSN to interrupt enabled GPIO pins on a CC2530 micro controller. To detect occupancy a rule associated algorithm was implemented based on possible scenarios. When a room was considered occupied the reed switch would have to detect that a door has been opened. This assumed that for a typical office building an open door denotes the occupant being in the office or being somewhere nearby. Similarly, when people leave for an extended period (such as the end of the day) they typically close the door to their offices for security reasons. When a door closing event happens, there are two assumptions. Either the person closed the door and headed out (room unoccupied) or the person just closed the door and is still inside the room (room occupied). To disambiguate between these two cases the PIR sensor was used to determine if someone walked near the door. If the PIR sensor is triggered, then it is assumed there is still a person in the room. If the PIR sensor does not detect motion, then there is no occupant in the room. Based on this assumption and the simulated experiment using EnergyPlus, they concluded that HVAC system operation costs were reduced by 10-15%. An issue with this assumption, however, is that a PIR motion sensor may not detect occupancy if the occupant is remaining stationary for long periods of time, therefore based on this experiment's assumptions they will obtain erroneous data. Since

PIR motion sensors only detect explicit movement past the sensor, an integration of other sensors such as sound and luminosity should complement the PIR sensor to negate false triggers.

Having the same purpose of optimizing HVAC consumption, Plageras (2017) proposed a similar indoor WSN design that collects data produced from each sensor in a smart building and then pushed and stored that data into a cloud server. The proposed WSN platform consisted of collecting temperature, humidity, motion using PIR, and luminosity. The sensors were selected to facilitate building management with provisions of reducing energy consumption caused by inaccurate HVAC schedules. The operating system Contiki and an emulator Cooja were used when conducting the simulated experiment, while Wireshark served as the analytical tool. The results show that with the proposed system architecture, they obtained enough ambient observations in order to observe environmental quality to optimize the HVAC system and reduce energy consumption. Even though this experiment proved the usefulness of this integrated WSN platform, this experiment has used simulated data and a small sample of sensor data to build their building energy consumption model.

Another study on deploying a static sensor network consisting of infrared cameras, a microphone, door contacts, temperature and humidity sensors, and a wearable kinetic sensor was proposed to predict Activities of Daily Living (ADL) by Fleury (2010). The experiment was conducted over a two-week span with 13 participants for monitoring: hygiene tendencies, dressing and undressing, eating habits, resting and sleeping periods, and toilet use. By analyzing data from ambient room sensors and

passive infrared (PIR) motion sensors using SVM, he classified the ADL to improve the health of participants.

3.3 Non-intrusive General-Purpose Sensing and SVM

General-purpose-sensing relies on collecting ambient and dynamic environmental information using non-intrusive means for detecting occupancy. Few attempts have been found in the literature where SVM is applied to classify types of occupancy based on general-purpose sensor data. One example includes Laput (2017), where a general-purpose sensing unit was implemented for gathering sensor data from an accelerometer, microphone, temperature, humidity, a 2.4 GHz Wi-Fi chip, and PIR motion sensor. This approach mainly relies on classifying microphone data for distinguishing the length of events. The length of events would distinguish which type of event occurred. For example, the microwave would have three sound events when the food is ready, the oven would have one sound event and the blender would have a continuous sound event. Based on this criterion the authors were able to manually label events which were used to classify the type of appliances being used by the occupant. While this approach utilized an SVM model to classify specific simulated interactions such as: turning on a blender, microwave stove and faucet, the classification accuracy of events achieved 85 %.

Abade et al. (2018) proposed an open source framework for non-intrusive occupancy detection in an indoor environment based on four essential sensing units; Temperature, CO2, sound and luminosity. The IoT platform for this experiment consisted of an Arduino Yun micro controller which would send data every ten seconds to a raspberry pi then aggregated means were taken of six values from each sensor and stored

in a MySQL database. Logistic regression, SVM and Neural Networks were compared to individual sensor values to determine the accuracy of occupancy for each sensor. The results of this experiment concluded that logistic regression, followed by SVM yielded the most accurate results. The results of the temperature exceeded 89 % for both SVM and logistic regression while CO2 and Noise sensors received accuracies of 6 % and 1 % for logistic regression and SVM. The luminosity was the best indicator of when the room was occupied as it obtained an occupancy detection accuracy of 95.60 % for both machine learning algorithms. Due to sensor events being caused by different activity, some sensors would have far fewer events then other, for example luminosity would have far more events, especially if there is light pollution occurring from the windows. Therefore, having an accuracy of above 95 % for occupancy detection may have been caused by external noise. Combining time series and event-triggered data allows us to determine the trends of the room and combing the sensor data can mitigate sensors which are triggered less often, which can hinder the accuracy for occupancy detection and prediction. More importantly amalgamating the sensor data from not only just one node can create a more robust experiment, correlating occupancy with changed values from each sensor in an organized trigger sequence.

Brennan et al. (2018) suggested utilizing CO2, Temperature and humidity to estimate the occupancy by obtaining activities of daily living gathered by sensor nodes in a complex room inside a building. In total, six sensor nodes were placed in two laboratories with different layouts. Four were placed in a larger laboratory, two nodes were placed with effective coverage of four desks and two were positions at the entrance of the lab. The remaining nodes were place in the smaller laboratory. To filter and label

the data the maximum and minimum distribution of observations from each sensor node were taken. For handling redundancy, the authors eliminated observations if the threshold of the CO2 values did not exceed 5ppm. Samples were normalized on a per minute basis by collecting samples and taking the median of the observations for a 24-hour time window over a span of a week. Manual occupancy labelling was done based on the distribution of observations and peak measurements from CO2 and humidity values. Brennan concluded the study by evaluating the correlation of CO2, humidity and temperature and through indoor localization using sensor nodes and from the distribution of measurements were able to estimate occupancy within a room for smart systems.

3.4 Summary

In summary, this Chapter has elaborated on the current methods of indoor WSNs for IoT. These platforms usually employ an open source interoperable sensing platform capable of measuring indoor ambient environmental measurements. Existing architectures allow for IoT devices to be connected through Wi-Fi or BLE. Large scale deployments for BMS have also been made simulating occupancy towards optimizing HVAC schedules (Agarwal, 2010; Depatla, 2015 Plageras, 2017), while Laput (2017) explored combining accelerometer, gyroscope, magnetometer and microphone to classify human-object interacting using SVM.

Laput's research work has motivated my research on exploring SVM for predicting occupancy behavior, because it utilized a data driven event-triggered approach. However, my research hypothesis is that by combining event triggers and times series

data, an IoT platform will be able to distinguish occupancy trends of a room and determine the outliers caused by individual occupancy.

Furthermore, previous data driven approaches have shown promising applicability for measuring non-intrusive occupancy information using general-purpose sensing units (Abade, 2018; Brennan et al, 2018; Laput, 2017). This research motivates to collect non-intrusive data used to predict occupancy patterns in a real-world scenario which avoids simulating events or using mathematical models. SVM has a strong potential to be used to predict occupancy, as it is a versatile machine leaching algorithm due to its capability of predicting binary outcomes (i.e. occupied or not occupied) and the fact that it is computationally inexpensive and can be executed at the gateway or in the cloud.

CHAPTER 4: METHODOLOGY

This Chapter describes the proposed architecture for our IoT platform consisting of five layers, named as Sensing Layer, Edge Access/Processing Layer, Network Layer, Cloud Access/Processing Layer, and Application Layer. The Sensing Layer handles the increased sensor data generated by generic sensing approaches. The Edge Access and Processing Layer is designed for processing the sensor data before it is stored in the cloud. Here the data can be processed at a gateway in order to reduce the amount of incoming data into the cloud, avoiding latency and bottleneck issues. The Cloud Access and Processing Layer provides feedback of occupancy and occupancy trends towards reducing energy consumption, feedback for building management, and contributes to a comfortable living environment. However, in this experiment the processing layer will be introduced to create a four-layer IoT platform. In the future the application layer will be introduced to create a five-layer IoT platform. The role of the Application Layer is to provide an interface capable of facilitating building management, energy reduction, space optimization, indoor quality, and reduce costs of operation.

This IoT platform is unique in supporting an SVM model for predicting occupancy patterns in indoor environments. The mathematical formulation of SVM models is also given in this Chapter, and how to optimize the accuracy of SVM by tuning its three hyper parameters to best fit an IoT dataset.

4.1 Overview of the proposed IoT Platform

Figure 5 illustrates the five layers of the proposed IoT platform for supporting SVM models to predict occupancy patterns in indoor environments.

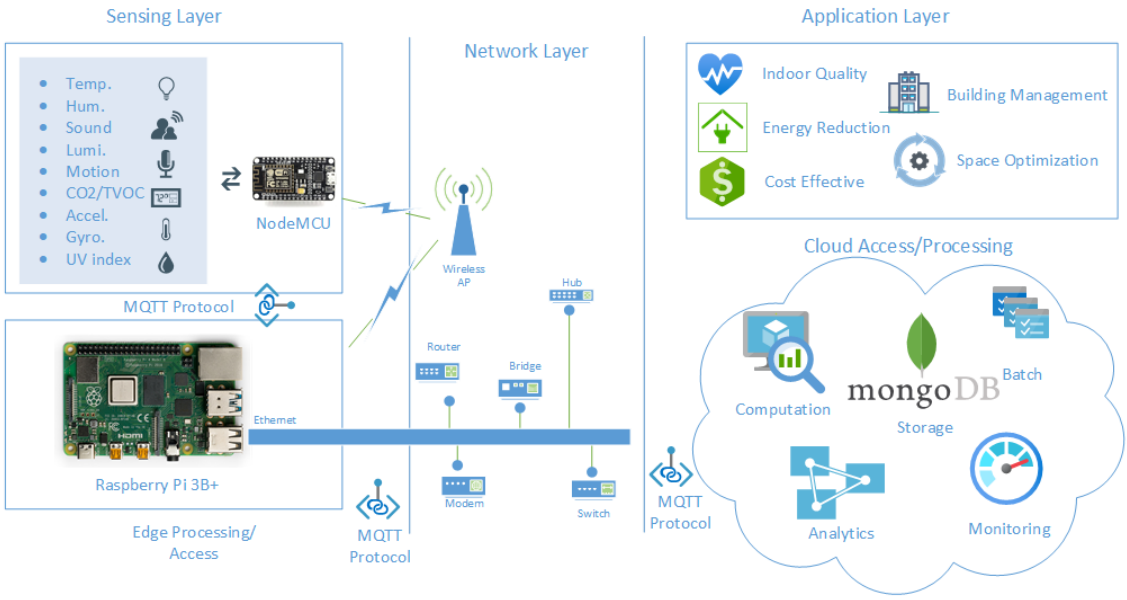


Figure 5: Proposed five-layer IoT platform

Sensing Layer

The Sensing Layer is responsible to convert the digital signals into collected data from 5 up to 10 different types of sensors. The hardware selected for this layer consists of two main elements: the sensors and the NodeMCU. The sensors are used to measure non-intrusive information about an individual or a group of individuals by being able to collect sensor data about the room and their dynamic changes caused by human occupancy. The sensors include: temperature, humidity, motion using a passive infrared (PIR) sensor, object sound detection, indoor CO2/Total Volatile Organic Compounds (TVOC) levels, and luminosity. Each sensor has a direct wired connectivity to a

NodeMCU, an open source IoT platform module, programmed in Arduino IDE. The NodeMCU is then configured to send every sensor data using Wi-Fi connectivity to a Raspberry Pi B+ gateway.

The sensor data consists of 2 types of data tuples which can be described as follows:

- Event triggered data: these are data tuples consisting of all measurements from the sensors. For example, this includs motion, sound event, temperature, CO2 events.
- Time series data: these are data tuples consisting of all measurement from the sensors that are collected with a fixed data rate.

Edge Access and Processing Layer

The Edge Access and Processing Layer provides transient data storage for the data received from the sensors and it also performs computing tasks in real-time, and up-links the data to the cloud. This layer is at the edge of the network. At the edge, local data processing tasks can be assigned which leverage the powerful local processing units to shift the data computation and storage from the cloud. In the proposed IoT platform, edge computing is accomplished using Raspberry Pi gateways, and Arduino NodeMCUs. The data fusion runs at the edge. As the data arrives from multiple NodeMCUs, the Raspberry pi 3B + is programmed to fusion sensor data into one 24-hour file containing the event and time series data from the deployed sensor nodes. When the data exceeded 24 hours other files can be created and stored locally and later pushed to the cloud.

Other tasks can also be developed such as response, low-cost management as well as fewer data transferred to the cloud. The increased processing power, Wi-Fi protocols,

lower power input requirement and programming versatility of the Raspberry pi 3 B+ made it the preferred gateway over the Arduino Uno micro-controller.

Cloud Access and Processing Layer

This layer in responsible for data storage, batch processing tasks, data monitoring and performing the analytical pipeline for creating the SVM predictive model. As the sensor data arrives in the cloud, it is stored in MongoDB. Data stream monitoring is performed to ensure that the data is still being collected from the sensor nodes. Later the data is extracted from MongoDB for running cleaning tasks and transforming the data for training the predictive model. Section 4.3 describes in detail the analytical pipeline developed for the SVM predictive model.

Networking Layer

The network communication between the sensor layer and both access and processing layers at the edge and cloud can be configured using a variety of networks. The work of Ferdoush (2010) proposes to establish communication using ZigBee as can be an alternative to Wi-Fi when the Sensor Layer is deployed in areas with sparse Wi-Fi connectivity. Other methods include Wi-Fi (Laput, 2018; Brennan, 2018; Agarwal, 2010; Abade 2018; Depalta, 2015) and BLE (Huh, 2017). Based on the literature review and Table 9 in Chapter 2, Wi-Fi connectivity, operating in 802.11 protocol with Wi-Fi frequency band of 2.4 GHz was selected due to its wide availability and ability to transfer large amounts of data. A widely used IoT communication protocol to exchange data tuples was used. This protocol relies on a publish/subscribe method to publish data to a

topic in the MQTT broker and then subtribe to that topic by another devices, such as a Raspberry pi 3B+. In other words, the NodeMCU client would publish the data to the MQTT broker and the gateway would subscribe to retrieve the data. Direct wired connection to the Internet is used for the gateway for time synchronization of the Raspberry Pi, and for lower latency when transmitting larger amounts of sensor data. To send the data tuples from the Edge Layer (i.e. gateway) to the Cloud Layer, the publish/subscribe protocol is utilized once again to send and retrieve data from the HiveMQ broker.

MQTT offers flexibility because it supports three levels of quality enforcement which can be described as one of the following:

- Message sending without an acknowledgement request;
- Message sending once with an acknowledgment request,
- Message sending through a handshake mechanism.

Application Layer

The role of the application layer is to support an interface capable of facilitating building management, energy reduction, space optimization, indoor quality, and reduce costs of operation. This layer was not developed in this research work due to time constraints.

4.2 Fundamentals of SVM

SVM is a supervised learning algorithm which relies on separating objects via hyperplane and selecting the appropriate event by successively shrinking or reducing a dataset to determine which objects are closest to the hyperplane(Joachims et al, 1998) In this algorithm, each data item (i.e. measurement) is plotted as a point in n-dimensional space (where n is the number of features in the dataset) with the value of each feature being the value of a particular coordinate. Support Vectors are simply the coordinates of individual observation and Support Vector Machine is the point which best segregates by the two classes via hyper-plane, as shown in Figure 6. Classification occurs by inputting a training sample set consisting of 60% of the original dataset. Then the algorithm can be parameterized to find the best fit hyper-plane that differentiates the two binary classes.

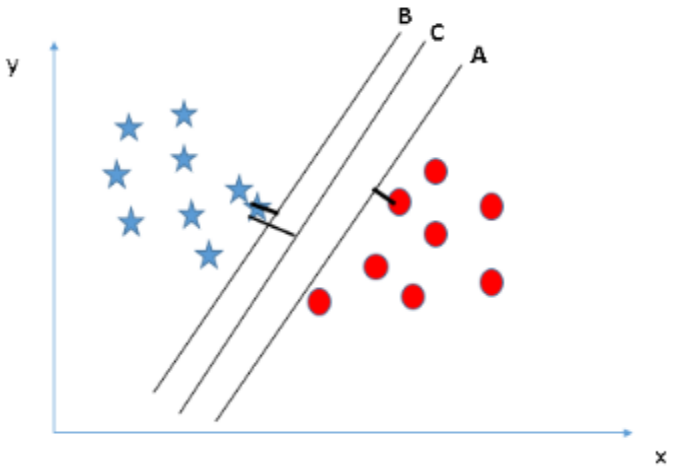


Figure 6: Establishing the hyper-plane when using SVM.

To find the hyperplane f(x), a category of "−1" need to fall into the range of "y" defined by f(x) < 0, a category "+1" needs to fall into the range of " y" defined by f(x) >

0. Therefore, we can distinguish the categories according to the sign of f(x). The hyperplane equation of "f(x)" can be expressed as:

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b \qquad (1)$$

Where w is the normal vector of this hyperplane, x is the input vector, −b/w is the distance from the origin perpendicular to the hyperplane. If w is a unit vector, this distance is −b. Thus, w and b are the parameters to search for. To find the maximum margins and minimum square error to solve the hyper-plane the objective function must be used as is defined as follows:

$$E(\mathbf{w}, b) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i \left(y_i - f(\mathbf{x}_i)\right)^2 \qquad (2)$$

Where $\alpha = \{\alpha_1, \ldots, \alpha_m\}$ is the coefficient of Lagrange and $\alpha_i > 0$, i = 1, 2,... ,m. Maximizing (2), the following equation can be obtained:

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0. \qquad (3)$$

Substitute the aforementioned equation into (2). The original objective function is converted into a dual objective function, and the objective function can be redefined as:

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathrm{K}(\mathbf{x}_i, \mathbf{x}_j). \qquad (4)$$

It also satisfies:

$$\sum_{i=1}^{m} \alpha_i y_i = 0; 0 \leq \alpha_i \leq C, \ \forall i = 1, 2, \ldots, m \tag{5}$$

Where $\alpha = \{\alpha_1, \ldots, \alpha_m\}$ is the coefficient of Lagrange, $K(x_i, x_j)$ is a kernel function, m is the number of training examples, and C is an adjustable positive parameter. For separable linear problems, C value is infinite. On the contrary, for non-separable linear problems, C value is a positive integer. This equation is a quadratic equation, so it can be solved by using the optimization algorithm. By substituting the obtained value of $\alpha = \{\alpha_1, \ldots, \alpha_m\}$ into (3), the w vector is obtained. By substituting w into (1), b value can be obtained. $K(x_i, x_j)$ is a kernel function of positive number. In this paper, a Radial basis Function (rbf) was used as the kernel function, defined as follows:

$$\mathbf{K}(\mathbf{x}, \mathbf{x}_i) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{x}_i\|^2}{\sigma^2}\right) \tag{6}$$

Where σ acts as a variance parameter.

4.3 The analytical pipeline for the SVM predictive model

The analytical pipeline that performs the SVM predictive model is relatively simple as illustrated in Figure 7. The initial phase is to input the data tuples to further perform pre-processing tasks such as cleaning, normalization, transformation (e.g. categorical values to numerical), and labeling tasks. Data cleaning involves detecting any missing, noisy or incomplete data. If duplicated, noisy and unreliable data is used during

the training phase, the accuracy of the SVM model is expected to be low. Therefore, data cleaning tasks can take considerable amount of processing time.

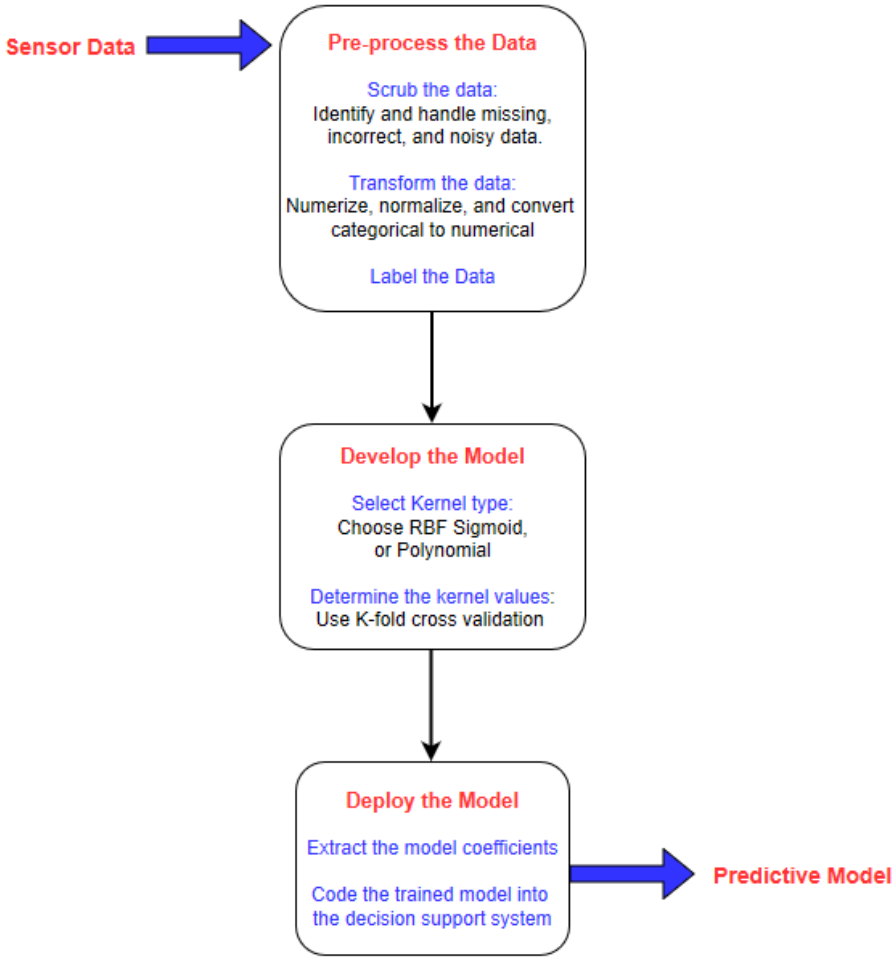


Figure 7: Process for creating an SVM predictive model.

Moreover, the labelling task is also a time-consuming task because it requires to label the data tuples that have attributes such as sensor ID, timestamp, and the measured sensor values from the on-board sensors. For optimizing the labelling task, a simple rule is created where the label Occupied is annotated to each event. This included associating triggered events with occupancy. If there was an event that exceeded its threshold the

room was labeled as occupied. The product of data preprocessing is then passed to the next phase; building an SVM model using training and testing data sets.

To build the SVM model using a training data set, three key hyper-parameters must be considered to determine the optimal location for the hyper plane as described by Cheng (2013). They are: the kernel, gamma and C parameters. Kernels are functions which take a low dimensional input space and transform it into a higher dimensional space. There are three potential kernel functions: sigmoidal, radial basis functions (rbf), polynomial and linear. The gamma parameter is the coefficient of the kernel. If gamma has a large value, then variance is small, implying the support vector does not have wide-spread influence. A large gamma leads to high bias and low variance models, and vice-versa (Girard, 2016). Penalty parameter C, or the cross-validation score is considered to be the error term. It controls the trade-off between smooth decision boundary and classifying the training points correctly. Figure 8 and Figure 9 illustrate the effects that the Gamma and C coefficients have when using a rbf kernel.

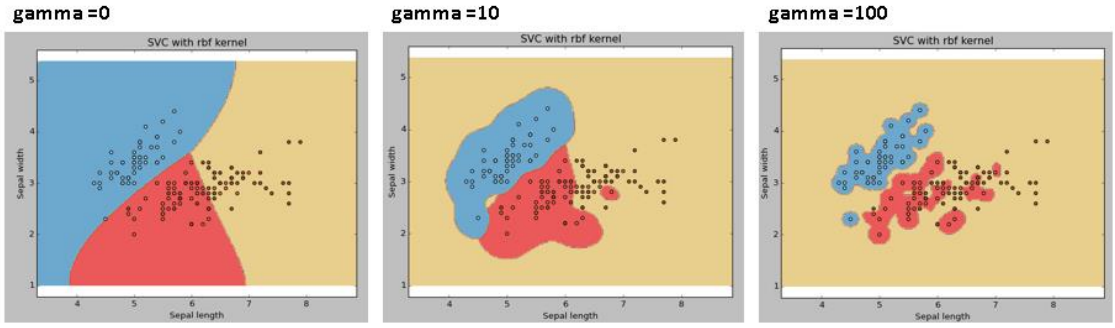


Figure 8: Effects of a higher Gamma coefficient.

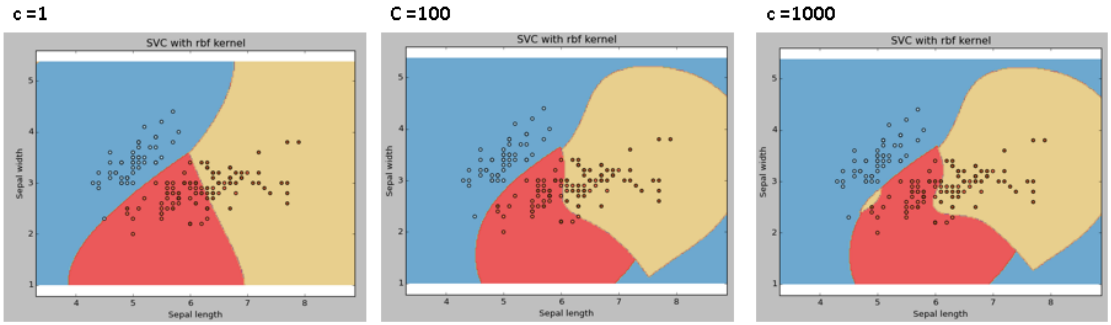


Figure 9: Effects of a higher penalty parameter C.

A common method for determining the selection of the hyper-parameters is to perform a grid search. Grid searching is a process of scanning the data to configure the hyper-parameters for a given model. Grid-Search builds a model on each parameter combination possible. It iterates through every hyper-parameter combination and stores a model to determine which combination of hyper-parameters yields the optimal results (Bergstra, 2017). Following the grid search to validate the training towards building the predictive model a k-fold cross validation is performed. Cross-validation is a validation technique for assessing how the results of a statistical analysis generalizes to an independent data set (i.e. the testing data set). It is mainly used in settings where the goal is to estimate how accurately a predictive model performs in practice (Drakos, 2018). To execute cross validation the sensor data needs to be separated into a training/test split. This strategy is simply having the training and test datasets, so they do not overlap as shown in Figure 10.

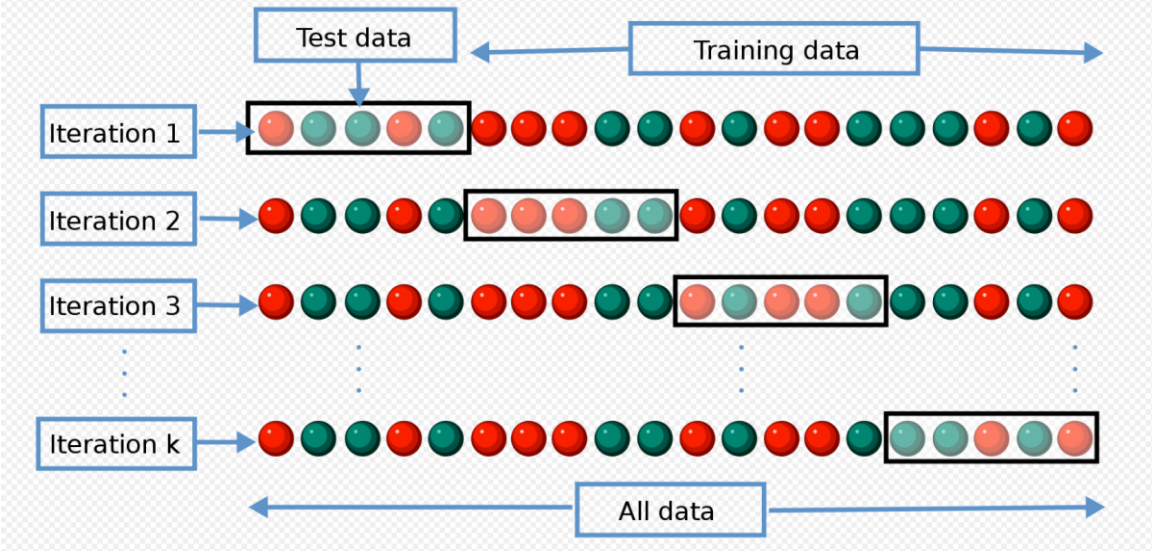


Figure 10: Process of cross validation (Drakos, 2018).

The goal of cross-validation is to test the model in order to limit problems such as overfitting, underfitting. Underfitting refers to not capturing enough patterns in the data while overfitting has the potential to capture noise and patterns which do not generalize well the unseen data (Drakos, 2018).

To assess the performance of the SVM predictive model, three integral measurements must be considered. The precision recall and F1 score. These values are generated by comparing the true values of the labeled class with the predictive values. These significant measurements ensure that the predictive model accurately represents the ground truth values and that class imbalance has not occurred. The Class Imbalance Problem occurs when the class distributions are highly imbalanced. In this context, many classification learning algorithms have low predictive accuracy for the infrequent class (Longadge, 2013).

The cross-validation reflects the precision ratio of the correctly predicted positive observations to the total number of positive observations (True positives/True positives +

False positives). Having a high precision relates the low false positive rate. The recall is a sensitivity ratio, it is the ratio of correctly predicted positive observations to all the observations in the actual class (True positives/True positives True negatives. F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution (Joshi, 2016).

CHAPTER 5: EXPERIMENTS AND DISCUSSION OF RESULTS

This Chapter describes the experiments developed for the implementation of the proposed IoT platform. It also provides an in-depth analysis of the results for each experiment and elaborate on the previous hypothesis when cleaning the data and training the SVM prediction model.

5.1 Description of the experiments

The implementation of the developed IoT platform consisted of deploying three experiments in three unique environments to encompass different occupancy patterns. They include:

- A Classroom, representing a controlled environment where it occupancy is a-priori known.
- An Office, representing a semi-controlled environment when a period of time is expected to be occupied.
- A Hallway representing a completely random environment where occupancy is not a-priori known.

The sensor data was collected with a 10 seconds data rate for all the experiments. The first experiment consisted of deploying two sensor nodes, in the E-11 classroom, each designed to capture facilitate to initial occupancy and prolonged occupancy seen in Figure 11. The sensor node at the entrance captured initial entry and exit events, containing motion, sound, luminosity, CO2/TVOC sensors while the second node place

at the front of E-11 captured when lectures were in process. The on-board sensors for the node placed at the front contained temperature, humidity, pressure, luminosity, motion, heat index and sound.

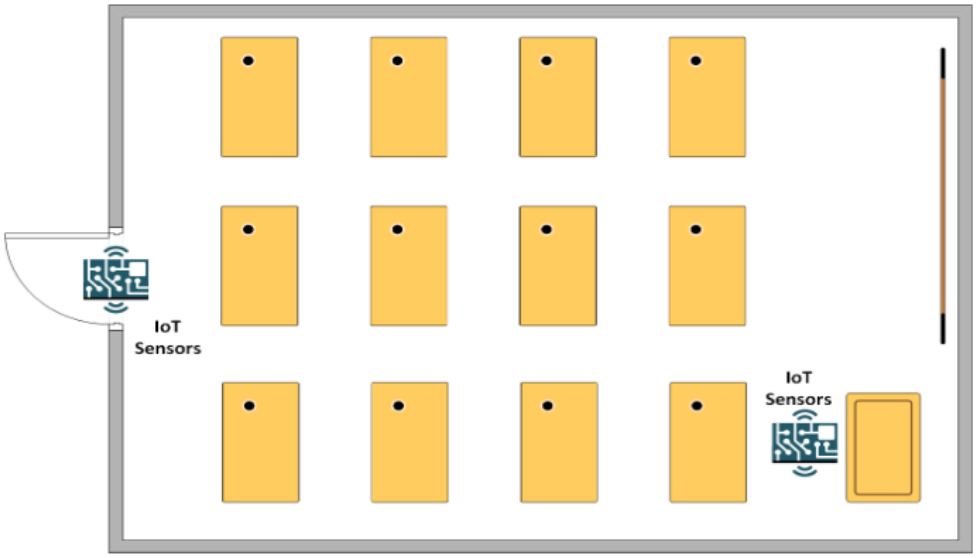


Figure 11: Sensor node deployment in E-11.

The hypothesis for designing sensors nodes for prolonged and initial occupancy was to determine the correlations and impact that an occupant has upon entry to the ambient environment and that having sensor nodes tailored to capture specific events would best represent when a people have entered the room. However, it created issues when consolidating the data from the two sensor nodes, which is further elaborated in Section 5.2 (Data-preprocessing).

The duration of the experiment in the E-11 classroom was two weeks and was selected to represent a controlled environment due to scheduled instances of occupancy from classes. Communication between devices operated in 802.11 with Wi-Fi frequency band of 2.4 GHz. A bandwidth of 2.4 GHz was preferred due to its longer range which

allowed for a lager coverage of the gateways, which is an important consideration for scalability for future experiment. The network consisted of utilizing Wi-Fi and MQTT protocols to publish the data tuples to HiveMQ, a MQTT broker, then allow the subscription to the gateway to pass data tuples which were then stored locally, seen in Figure 12. In total 247,695 observations were collected and pre-processed in order to create the occupancy prediction model using SVM.

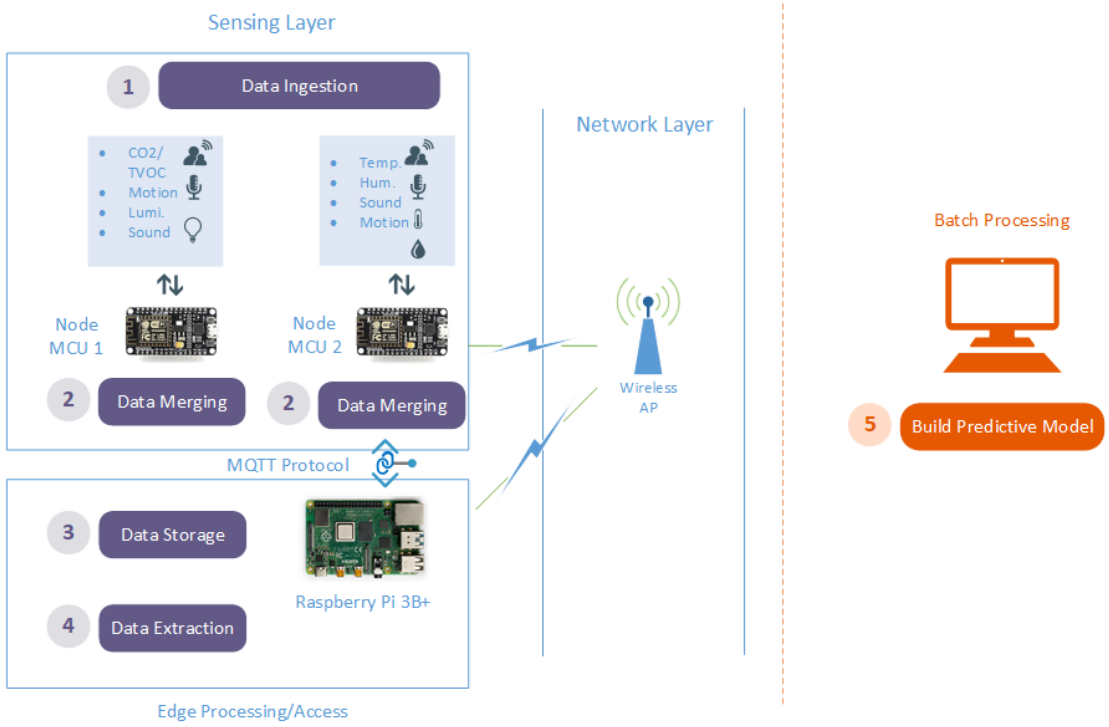


Figure 12: IoT platform for Experiment 1.

The second experiment was deployed following the summer semester in the head hall complex on E-level during the month of June and lasting until July. This experiment was conducted in a completely random and a semi-controlled environment where occupancy is sporadic and is not scheduled like a classroom. To begin collecting occupancy information in a random environment the first sensor node was deployed in a

hallway leading to E-54 GGE department office, while the second, which was collecting occupancy information in a semi-controlled environment was place in E-54 on the middle of the GGE department office.

The locations of the two sensor nodes were selected to facilitate to group monitoring in hallways and for recognizing patterns of events due to people entering the department offices, as there are many intersections between multiple offices and laboratories. The data from the two sensor nodes were combined as the data arrived at the gateway, based on sensor ID and MQTT timestamp. The data was initially combined for determining correlations between the events occurring and the time-lapse in between the events. A direct wired connection to the internet was used for the gateway for two reasons: firstly, for time synchronization of the Raspberry Pi to ensure there was no drift caused by the internal clock on the Raspberry Pi, and for lower latency for transmitting the larger amounts of data from multiple sensor to the cloud for storage.

To transmit the data from each device, HiveMQ was utilized once again to publish the combined sensor node data. The cloud then subscribes to the MQTT topic to retrieve and store the data, where batch-processing tasks occur for eventually building the SVM prediction model. Figure 13 illustrates the IoT platform implemented for this experiment. The location of the sensor nodes for experiment 1 and 2 are shown in Figure 14.

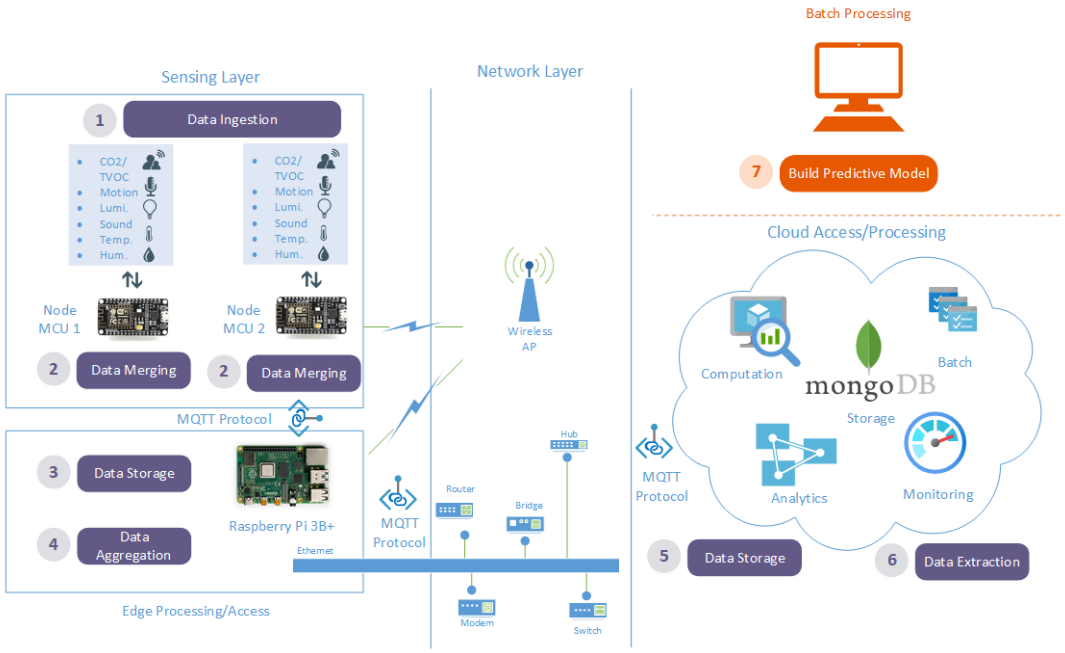


Figure 13: IoT platform for Experiment 2.

Figure 14: Sensor node deployment and gateway locations for Experiment 1 and 2.

The last experiment was deployed at the Polytechnic University of Madrid, consisting of one sensor node located in the middle of the 113 classrooms, during the winter semester, from April 1st until April 24th. The sensors contained on the sensor node was temperature, humidity, pressure, luminosity, CO2/TVOC, motion, sound, accelerometer, gyroscope, UV index and heat index. This sensor node was designed to test the viability of including additional environmental information; accelerometer, gyroscope and UV index. The additional measurements were collected and compared using a correlation matrix to determine if the any of the measured variables were correlated or if they were independent variables which can be used for strengthening the predictive model.

The communication network architecture was similar to the first experiment. It relied on using a Wi-Fi bandwidth of 2.4GHz to connect to HiveMQ which published and subscribed topics to a data repository, however unlike the last experiment the data was sent directly to the cloud rather than to a gateway. This communication network was experimented with as a means for having a direct data stream to the cloud in the hopes of an achieving lower latency for real-time analytics as shown in Figure 15.

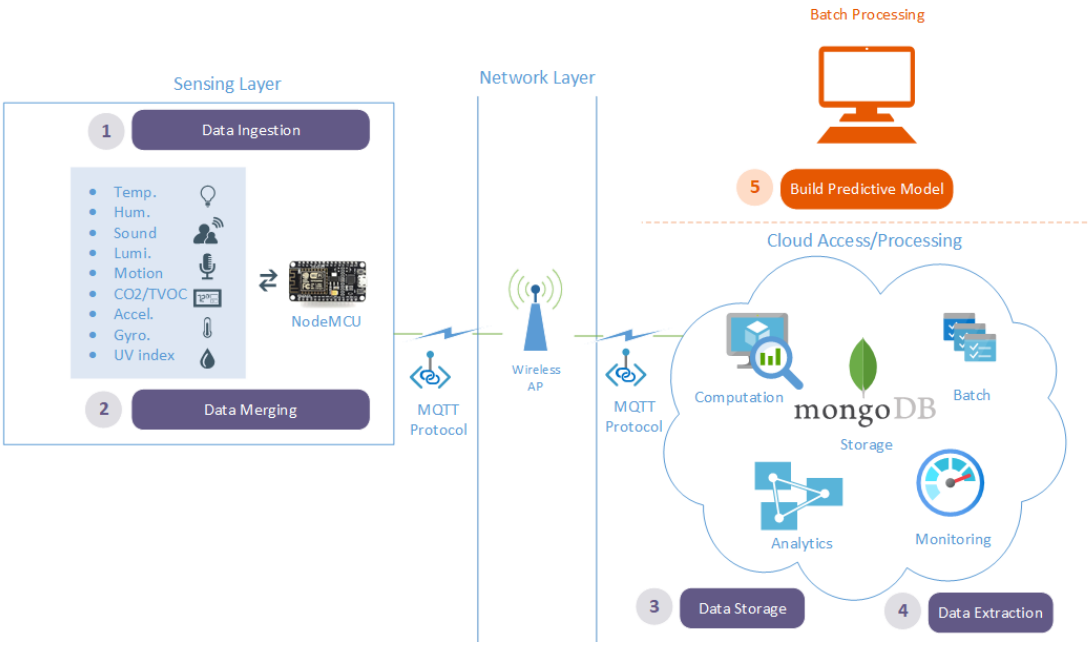


Figure 15: IoT platform for Experiment 3.

Over a 24-day span, a total of 452,599 observations were collected using the same data rate as the previous experiment of 10 seconds. A detailed description of the experiments is found in Table 10 and the sensors used for each experiment are described in Table 11.

Table 10: Overview of the experiments.

#	Environment	Location	Duration	Sensor Nodes	Network Connectivity	Total # of observations
1.	Controlled	E-11 Front E-11 Entrance	2 weeks	2	Sensor Nodes → Raspberry Pi	247,695
2.	Semi-Controlled Random	E-49 Office E-49 Hallway	1 Month	2	Sensor Nodes → Raspberry Pi → Cloud	1,616,362
3.	Controlled	113 (Madrid)	1 Month	1	Sensor Nodes → Cloud	452,599

Table 11: On-Board sensors for each sensor node.

Sensors	*Sensor Nodes*				
	E-11 (Front)	E-11 (Entrance)	113 (Madrid)	E-49 (Office)	E-49 (Hallway)
Temperature	✓	✗	✓	✓	✓
Humidity	✓	✗	✓	✓	✓
Pressure	✓	✗	✓	✓	✓
CO2/TVOC	✗	✓	✓	✓	✓
Motion	✓	✓	✓	✓	✓
Sound	✓	✓	✓	✓	✓
Luminosity	✓	✓	✓	✓	✓
Heat Index	✓	✗	✓	✓	✓
Accelerometer	✗	✗	✓	✗	✗
Gyroscope	✗	✗	✓	✗	✗
UV Index	✗	✗	✓	✗	✗

5.2 Data pre-processing

During the first experiment two Arduino scripts were programmed to collect event and time series data from the two sensor nodes. If the ambient data exceeded a threshold, manually programmed in the Arduino script, the type of event would be registered in a separate column in the dataset and sent to the Raspberry Pi. If no event occurred, data was sent to the gateway every 10 seconds. Since there were two nodes measuring different environmental characteristics the types of events were unique to each node. The node placed at the entrance was programmed to collect, motion, sound, luminosity, and CO2/TVOC, meanwhile the sensor node place at the front was collecting motion, sound, luminosity, temperature, humidity and pressure. As the data arrived at the gateway it was stored in two separate files, one file contained the data from the sensor node at the entrance while the other file contained the incoming data from the sensor node at the front of the E-11 classroom.

The data was then extracted from the Raspberry Pi, separated into columns and manually labeled headers were created to describe each columns in the dataset. By utilizing Python's global merge function, the data from both sensor nodes was concatenated based on timestamp and sensor ID. However, to reiterate the two sensor nodes did not contain the same sensors, which in turn created an inconsistent dimensionality of measurement. To address this issue weighted means replaced missing values to create a consistent data frame, which was used to build the predictive model using SVM.

Another pre-processing task included converting all of the categorical data into a numerical values. Then "Event column", which contained a categorical description of each of the events, was converted as a series of numbers each representing a unique event as shown in Table 12. The measured values for motion and sound were converted to a binary set of values, 0 signifying the sensor did not detect motion or sound and 1 meaning that motion or sound has been triggered. Following this pre-processing task the cleaned data was stored in a separate value and used to train the SVM predictive model.

Table 12: The numerical values assigned to the categorical descriptions of events.

Event	***Numerical value***
Time	0
Motion	1
Sound	2
Luminosity	3
Temperature	4
Humidity	5
Pressure	6
CO2	7
TVOC	8

The second experiment was designed to include a consistent dimensionality of measure, therefore the same sensors were implemented on the two sensor nodes. The network communication for this experiment also included sending the data to the gateway however the gateway would publish data tuples to HiveMQ and subscribed to by the cloud. Prior to sending data to the cloud however, an automated pre-processing task was implemented on the raspberry pi. The data was merged into one file and assigned headers to describe the measured values. Like the previous experiment, the categorical data was converted to numerical. The same values, from the previous experiment were assigned to the “Event”, “Motion” and “sound columns. The pre-processing time was reduced greatly by aggregating sensor data on the Raspberry Pi. The only cleaning tasks that needed to be performed for this experiment was converting categorical data into a numerical values.

The next phase was to save the cleaned data in a separate file and being training the model.

The preprocessing tasks for the Madrid experiment consisted of creating a correlation matrix in Python to determine if the additional measurement (i.e. accelerometer, gyroscope and UV index) had any correlations with occupancy. The correlation matrix depicted that the additional measurements were not correlated with occupancy. Therefore to retain consistency between experiments, the accelerometer, gyroscope and UV index columns were removed. Even though these measurements were not used in the experiment they prove too valuable for future experiments for a mobile sensing units, for example deploying a sensor in an elevator. The next task included converting the categorical data into numerical, similarly to the previous two experiments. The timestamps associated with this experiment were in UTC time. The timeframes were converted to a local timeframe using pandas, a popular data analysis toolkit. Following this procedure the cleaned data file was saved to a .csv a used for building the prediction model.

5.3 SVM implementation

This section describes the process for training the SVM model in order to predict occupancy. When importing the cleaned .csv files from each experiment, a new "Status" column was created to reflect occupancy in a binary representation, which was directly associated to the "Event" column. If the value for the "Event" column was 0, then the room was considered to be free, therefore labeled as 0 in the "Status" column, otherwise it was labeled 1 for occupied. This assumption was generated from the fact that if

occupancy occurs, then a series of events will follow the initial event. The “Status” column was be the focal measurement of occupancy because it was a binary representation of when the room was occupied or free, which is a key component for training the SVM algorithm.

To validate the training process, a grid search and 10-fold cross validation test was performed to determine the most optimal values for tuning the hyper parameters. To perform a grid search a series of possible hyper-parameter values are selected and executed iteratively to determine the most accurate prediction. The grid search hyper-parameter tuning for the first experiment concluded that the values of C = 1.0, kernel = ‘rbf’ (radial basis function) and a gamma = auto depreciated, would yield the most accurate results, seen in table 10. Using sklearn (a machine learning Python library) the hyper-parameter were input along with the status label (occupied/free) to 80% of the dataset. Leaving the remainder of the unlabeled dataset (20 %) for predicting occupancy. The same process was performed for the other two experiments. The cleaned dataset was imported, an additional column was created labeling the status of the room and a grid search was executed to determining the most optimal hyper-parameters, the hyper parameter adjustments can be seen in Table 13.

Table 13: Hyper-parameter tuning prior to performing a grid search, for each experiment.

EXPERIMENT	KERNEL	GAMMA	C
1	Radial Basis Function (‘rbf’)	auto	0.50
2	Radial Basis Function (‘rbf’)	auto	0.75
3	Radial Basis Function (‘rbf’)	3	0.50

5.4 Discussion of Results

The results of the first experiment located in the E-11 classroom led to the conclusion that room usage was sporadic regardless of the scheduled class times. The data displayed that occupancy occurred throughout the day and at night when exams and final presentations were occurring. A limitation of occupancy classification is individual occupancy due to fewer registered events. The sensor data depict the initial instance of occupancy using the PIR motion sensor. However when remaining stationary for longer periods of time the motion sensor did not register movements. Therefore other sensors were integrated to compliment stationary occupancy. If the occupant was relatively close to theCO2 and sound sensor, increased sound activity and CO2 concentrations were registered which is reflected in the results.

When group occupancy occurs, the labeled class coincided with the occupancy and class scheduling of the room. Motion was the primary sensor triggered when the room became occupied, however patterns of sensor triggered were also recognized, for instance, motion was triggered, CO2 values increased for a brief moment as the person walked by, followed by luminosity and noise. Temperature changes were not as significant as the room was regulated using the HVAC system at all times, which is also an important finding for this experiment because rooms are heated and cooled at a specific threshold despite having anyone in the room for prolonged periods of time. CO2 and TVOC events were also among the highest events being triggered which begged two questions, if no other sensors are triggered, has someone simply walked by the room and not entered or is the air quality in the room changing 80 ppm causing an CO2 event to be triggered, if so will adjusting the threshold mitigate this issue?

Another interesting finding displayed large spikes in CO2 and TVOC changes throughout the first week of April. The values ranged from 50,000 ppm to the maximum value of 60,000 ppm. These findings presented another event occurring in the room besides occupancy for lectures or studying session. The room was being renovated and when re painted this caused the values for the CO2 and TVOC sensors to rise because of the fumes being emitted. When the painter was occupying the room other sensors were being triggered along with having high CO2 and TVOC values, however on April 4th the other sensors stopped registering events, the only event which re-occurred was CO2 and TVOC. During that time frame the room was unoccupied because of the fumes and to allow the paint to dry.

The hypothesis of the results for the first experiment were tested on the results from the second experiment. Both led to similar conclusions, the CO2 and TVOC were among the most triggered events for both experiments. The data from the sensor located in the E-54 GGE department office had a remarkably high CO2 and TVOC event count, similar to when painting had occurred in the classroom E-11. Out of the 1.6 million observation collected approximately 200,000 observations were registered as CO2 and TVOC changes. Despite the one week of have high CO2 values many other CO2 events were also registered regardless of having any other sensor being triggered.

By analyzing the results of the two experiments threshold adjustments should be made to mitigate the slight fluctuation in the room air concentration of CO2, for a more concise occupancy label. On the other hand the sensor in the E-54 hallway and E-54 department office displayed that occupancy was drastically more sporadic than the E-11 classroom. Occupancy would occur in clusters from 9 am until approximately 6 pm

throughout the weekdays and occur less frequency during the weekend, which was expected. However, there were registered occupancy events during the night and even during the weekend, which is significant to consider for the occupancy model, and how would the model be able to facilitate to not only individual occupancy but rather to facilitate to completely random individual occupancy.

The final experiment conducted in Madrid yielded similar results as the experiment performed in the E-11 classroom. On the other hand, the CO2/TVOC events occurred far less than the sensor nodes deployed in experiment 1 and 2, which is one reason why there are far fewer observations (another being that there was only one sensor node deployed rather than two). However, noise and motion were the most triggered events. Occupancy occurred is dense clusters on Tuesdays from 8:00 am until 7:30 pm and Thursdays between 7:45 am until 11:00 pm and again at 3:00 pm until 6:30 pm (Madrid Time). Occupancy also occurred on Monday Wednesday and Friday, but far fewer occupancy events were registered. The results measured from the sensor node displayed that the instances of occupancy coincided with the class timetable found on the Polytechnic University of Madrid's website. Random occupancy occurred as well throughout the data, similar to the E-11 experiment, but occurred less frequently. Unfortunately the experiment was conducted externally therefore was difficult to develop a concrete hypothesis towards why random classroom occupancy occurred less than the experiment conducted in the E-11 classroom.

Evaluating the performance of the SVM algorithm comparative to the literature previously presented in Chapter 2, the results of all three experiments proved to reach a higher accuracy than previous research work. The occupancy prediction model achieved

a 94% occupancy prediction for the first experiment and 95% for the second and 92% for the third. To train the model a grid search was performed on dataset with three threshold options for the C value, gamma and kernel. The values were selected based on the output of the grid search and the suggested value which yielded the best fit for the model. The most optimal parameter tuning for the SVM model is arbitrary and will change with every dataset, which is why it is important to perform a grid search. The training was then selected based on previous studies and recommended training sizes for building the predictive mode, therefore 80% of the data was used to train the predictive model.

The accuracy of the model was assed as the total number of true positive and true negative observations divided the total number of observations (True positive, True negative, False positives and False negatives), seen in Figure 8. As previously mention the three focal metrics for assessing the performance of the algorithm, the precision, recall and F1 score was analyzed.

Experiment I		
	Free	Occupied
Free	TP = 39002	FN = 1351
Occupied	FP = 474	TN = 8712
Experiment II		
	Free	Occupied
Free	TP = 87239	FN = 17479
Occupied	FP = 591	TN = 217958
Experiment III		
	Free	Occupied
Free	TP = 56745	FN = 3623
Occupied	FP = 3016	TN = 27135

Figure 16: Confusion Matrix for Experiment I, II, and III.

The first experiment yielded an overall accuracy of 96 %. The precision value achieved a predictive accuracy of 99% (free) and 87% (occupied). The second metric evaluated was the recall metric. This value also achieved a high accuracy of reaching 97% for the free status prediction and 95% for the occupied prediction status. Lastly the F1 score, which is a weighted reflection of the precision and recall metrics achieved an accuracy of 98% when predicting the room as being free and 91% for the room being occupied, the results in the first experiment are displayed in Table 14.

Table 14: SVM Occupancy prediction results from experiment I.

	Precision	Recall	F1 Score	Support
Free	0.99	0.97	0.98	40353
Occupied	0.87	0.95	0.91	9186
Micro Average	0.96	0.96	0.96	49539
Macro Average	0.93	0.96	0.94	49539
Weighted Average	0.97	0.96	0.96	49539

The second experiment also achieved high accuracy for predicting occupancy. Comparatively the precision achieved 99% accuracy for the free predicted status and 93% for the occupied predicted status. The recall obtained accuracies of 83% and 100% and the F1-score attained accuracies of 96% and 94% for the free and occupied status prediction, seen in Table 15. Lastly the experiment deployed in Madrid score the lower accuracy values, however still achieved an accuracy of 92%. Evaluating the metrics of the SVM predictive model, in Table 16, the precision achieved an accuracy of 93% for occupied an occupancy status of free and 89% for occupied. The next metric evaluated was Recall. The accuracy of this measurement proved to be higher than of the second

experiment, achieving an accuracy of 91% when the room was free and 93% for when the room was occupied. This may be a results of having a more heterogeneous data sample. The number of free statues were much more balanced in the third experiment rather than the second.

Table 15: SVM Occupancy prediction results from experiment II.

	Precision	Recall	F1 Score	Support
Free	0.99	0.83	0.91	104718
Occupied	0.93	1.00	0.96	218549
Micro Average	0.94	0.94	0.94	323267
Macro Average	0.96	0.92	0.93	323267
Weighted Average	0.95	0.94	0.94	323267

The third experiment even though only have 90,152 observations for training, it contained 60,363 observations labeling when the room was free and 30,151 observations when the room was occupied. Alternatively the second experiment had 323,267 training observations, whereas approximately 100,000 observations were labeled free and 200,000 observations were occupied. This was a result to a much higher number of TVOC and CO2 events in the second experiment, which did not occur in the third experiment. Lastly the F1 score achieved an accuracy of 95% for free occupancy and 90% for predicting an occupied status. These scores were among the lowest achieved of all three experiments. This may be due to only having one sensor collecting occupancy events in a larger room therefore placing a sensor node at the entrance may facilitate to a higher granularity when predicting occupancy. It is important to highlight the fact this prediction model relied on five sensors (two sensor nodes) measuring non-intrusive information and the result demonstrated in this experiment achieved higher predictive accuracies than traditional methods. One of the limitations however is threshold adjustments. CO2 and TVOC were

predominantly changing in the data and from conducting three experiments while achieving similar results for two of them, it can be concluded that slight modifications to the CO2 and TVOC sensor thresholds should be made for future deployments. The results of experiment 3 can be seen in table 16.

Table 16: SVM Occupancy prediction results from experiment III.

	Precision	Recall	F1 Score	Support
Free	0.93	0.91	0.95	60368
Occupied	0.89	0.93	0.90	30151
Micro Average	0.93	0.90	0.92	90519
Macro Average	0.91	0.92	0.93	90519
Weighted Average	0.92	0.92	0.92	90519

CHAPTER 6: CONCLUSIONS AND FUTURE RESEARCH WORK

6.1. Summary

IoT platforms are key components for understanding how people use facilities within a building. Current research work relies on collecting intrusive information for validating occupancy or as a means for monitoring energy consumption within the building using mathematical or simulated models. The purpose of this study was to deploy an interoperable non-intrusive sensing network within an IoT platform capable of encapsulating patterns of occupancy towards behavioural pattern prediction that will be useful information for BIM, emergency protocols and facility management in the future.

To develop the five-layer architecture for the proposed IoT platform, it was then important to consider the network connectivity and the means for sending data tuples from the sensor nodes to the gateway and gateway to cloud. Therefore, to address these requirements Wi-Fi was utilized to connect the devices due to its range and ease of availability.

The sensor nodes were programed and deployed in three unique environments, each having a different network communication which send data in near-real time to a gateway (Raspberry Pi 3B+), to the cloud, or a combination of both. Each sensor node included into the IoT platform was evaluated based on cost, range, and accuracy, then compared with other available sensors in North America. One consideration however when deploying the sensor nodes was to determine the coverage of Wi-Fi access points and to maximize the distance between nodes and gateway for scalability for deploying additional sensor nodes in future experiments. To guarantee the base station had constant

Wi-Fi connectivity to the node MCU, a preliminary test bed of all possible base station and NodeMCU locations was performed that ensured Wi-Fi connectivity was maintained, and consistent data transmission was retained. Once the gateway was deployed the MQTT brokers, provided by HiveMQ, served as a viable, and secure means for publishing and subscribing data tuples for exchanging data between Nodes to Gateway and Gateway to Cloud.

Next it is important to consider the type of sensor data that will be collected. Time-series data was important for delineating the trends in each room and what the normal values were. Event based data strengthened my training model when using supervised machine learning to detect outliers which occurred due to occupancy. Unfortunately the data was inherently noisy due to breaks in connection regardless of determining the Wi-Fi coverage, or because sensors were being tampered with. Therefore to retain a consistent time series and event based dataset, the data stream had to be monitored closely and additional cleaning tasks were implemented.

6.2 Research questions

1. Would a single sensor or a set of sensors be more adequate to detect occupancy meanwhile preserving privacy?

To understand how different sensors can play a distinct role in detecting occupancy in such a way that a single sensor might be favored rather than others is not a trivial task. For example, the sensor nodes located in the class room were placed directly at the back of the room, because when classes were in session, the lecturer will be

triggering the motion, sound and CO2 sensors for approximately an hour and ten minutes. Meanwhile the second node was place at the entrance collecting data for when people have entered the classroom. The data displayed that the motion sensor at the door was triggered prior to the lecture as people entered the room and was being triggered less frequently as the lecture occurred. However, the sensor node at the back of the room was increasingly triggered, due to the lecturer.

Another consideration for selecting what types of sensors should be used will depend on the human activity that is expected to happen. The sensor nodes placed in locations where there was more activity, such as hallways or entrances of classrooms sensor at the door and the other at the front of the room is that areas further way from a door will be less affected by temperature, unless there is a densely populated room.

Alternatively, luminosity will also aid in classifying initial occupancy, and when occupants leave the room. Sensors placed in office spaces will be located in the middle of the room, since it is smaller, having a centralized point of data collection will be efficient for determining when people are occupying or not occupying the office space. If the sensors are placed at the extremities of the office space then false readings may occur. It is hypothesized that placing the sensor at the center of the room, where multiple office spaces intersect, it is possible to determine when people have entered the office space and upon further tests in the future will be able to determine which office space is in use.

2. What is the expected accuracy of SVM prediction models which are incorporated into IoT platforms?

The results presented in this thesis exceeded expectations by achieving a higher accuracy than any of the presented literature, while also retaining the privacy of occupants. SVM proved to be an accurate algorithm, yielding a predictive accuracy of 96% for the first experiment and 94% for experiment 2, and 92% for experiment 3.

The IoT experiments provided useful insights for selecting and deploying IoT sensors in an indoor environment, threshold tuning of the sensors and parameter adjustments to optimize the SVM algorithm, which in turn have had an impact on the accuracy of the predictions. For example, classroom occupancy would typically have an entry and exit events that would be triggered moments before and after a lecture. Therefore, it was important to fuse the triggered events with time series from both node MCUs located at the door and at the front of the classroom when a lecture is in session. This in turn strengthened the occupancy predictive model.

For environments such as hallways or office spaces, the high accuracy occupancy prediction is not as evident because more sporadic occupancy occurs than organized lectures in class rooms. To this end, determining how sensor data should be fused which accurately represents the indoor occupancy will continue to be a research issue to be studied in the future. From an accuracy perspective, more research work is needed towards how sensor data for whole floors should be fused, or whether the most appropriate approach should avoid data fusion.

Regardless the expected impacts different indoors environments will have on the accuracy of the SVM prediction model, and the architecture used for an IoT platform, more research is needed to evaluate other prediction models in the future.

3. Would validation of the proposed SVM model have to include alternative, more intrusive sensing units?

Although the proposed IoT platform for the prediction of occupancy using an SVM model was achieved, one of the main limitations of our approach was that we were unable to validate the results using ground truth. Other means for validating the results of the SVM model could strengthen the model in terms of accurately depicting when someone has entered a room. Introducing a video camera for validating the model only for training purposes could enhance the training label. This on the other hand will invade the privacy of occupants but will be a temporary alternative for evaluating the triggered events and correlate them with the video camera to ensure that the training model accurately represents occupancy. Another alternative would be to implement RFID tags which would serve as an occupancy counter. Similarly to the video camera this would be intrusive but yet the identity of the occupant would remain anonymous. Although it would require the occupant to temporarily use the RFID tag upon entry of the classroom(s) or hallways.

6.3 Future Research Work

The on board sensors were selected to measure a variety of ambient conditions and triggered events, which proved to accurately depict movement within the classroom, office or hallway. Based on the level of accuracy of the results, having a sensor located at the door and in the middle of the room can enhance the accuracy of the study. This can be reflected in experiment 1 and 2 comparative to experiment 3, which only had the one sensor located in the middle of the room. For future research, adjustments for the CO2 thresholds will be made before scaling the sensor deployment throughout the engineering complex. Another consideration for scaling up the deployment of the IoT indoor sensor network will be to geographically determine the best locations for the sensors. This experiment relied on consolidating data from a maximum of two sensor nodes, which efficiently measured initial occupancy upon entrance and when the occupants remained in the room. From the two sensors and consolidating that information it was apparent that there are specific events that occur sequentially. Therefore how can future experiments be scaled in order to maximize coverage and to depict sequential patterns similar to this experiment. Apart from adjusting the thresholds and sensor deployment, creating a more concise occupancy status label for training the SVM model will also be a consideration for future research. The current label occasionally misclassifies occupancy statuses when individuals are occupying a room, therefor optimizing the label will yield a more accurate occupancy prediction. Furthermore, integrating regression techniques into the analytical pipeline will associate timestamps with the predicted occupancy patterns, which will be fundamental for future research.

The research presented is a novel approach for occupancy prediction using non-intrusive information and the proposed IoT indoor wireless network is highly scalable and can be deployed on multiple floor levels and buildings due to the interoperability of sensors nodes, gateway and cloud. In the future, other technologies will be integrated into the architecture such as Cisco Edge and Fog nodes. These technologies will be a primary component from streaming analytics and pave the way for real-time sensor data analytics. The applicability of this research is not only contained to a room but can be used to analyze patterns of occupancy at the floor level, building and campus. This will contribute to optimizing operation times of HVAC systems for reducing CO2 emissions, determine if people are still occupying a room if there is an emergency evacuation, and facilitate to people within the building or campus by increasing the amenities by predicting occupancy patterns.

REFERENCES

Adafruit. (2018, 12 2). *ADAFRUIT CCS811 AIR QUALITY SENSOR BREAKOUT - VOC AND ECO2*. Retrieved from Adafruit: https://www.adafruit.com/product/3566

Abade, B. (2018). A Non-Intrusive Approach for Indoor Occupancy Detection in Smart Environments. *Sensors.*

Alexandropoulos, S., Kotsiantis, S., & Vrahatis, M. (2019). Data preprocessing in predictive data mining. The Knowledge Engineering Review, 34, E1. doi:10.1017/S026988891800036X

Agarwal, Y. (2010). Occupancy-Driven Energy Management. *Embedded Sensing Systems For Energy-Efficiency In Buildings* (pp. 1-6). Zurich: BuildSys.

Anastasi, G. (2003). *Experimenting an Indoor Bluetooth-based Positioning Service.* Rhode Island: IEEE.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(Feb), 281-305.

Brennan, C. (2018). Distributed Sensor Network for Indirect Occupancy. *IEEE* (pp. 1290-1295). IEEE.

Cheng, W. Y., & Juang, C. F. (2013). A fuzzy model with online incremental SVM and margin-selective gradient descent learning for classification problems. IEEE Transactions on Fuzzy systems, 22(2), 324-337.

Depatla, S., Muralidharan, A., & Mostofi, Y. (2015). Occupancy estimation using only WiFi power measurements. IEEE Journal on Selected Areas in Communications, 33(7), 1381-1393.

Dejan. (2016, February). *DHT11 & DHT22 Sensors Temperature and Humidity Tutorial using Arduino*. Retrieved from How To Mechatronics:

https://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/

Drakos, G. (2018, August 16). *Cross Validation.* Retrieved from Towards Data Science: https://towardsdatascience.com/cross-validation-70289113a072

Djelouat, H., Amira, A., & Bensaali, F. (2018). Compressive sensing-based IoT applications: A review. *Journal of Sensor and Actuator Networks*, *7*(4), 45.

Evans, D. (2011). The internet of things: How the next evolution of the internet is changing everything. *CISCO white paper*, *1*(2011), 1-11.

Ferdoush, S. (2014). Wireless Sensor Network System Design using Raspberry Pi and. *The 9th International Conference on Future Networks and Communications (FNC-2014)* (pp. 103-109). Elsevier.

Fleury, V. N. (2010). SVM-Based Multimodal Classification of Activities. *IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE*, 274-283.

Galar, A. F. (2012). A Review on Ensembles for the Class ImbalanceProblem: Bagging-, Boosting-, andHybrid-Based Approaches. *IEEE*, 463-483.

Girard, J. (2016, 04 22). *What are C and gamma with regards to a support vector machine?* Retrieved from Quora: https://www.quora.com/What-are-C-and-gamma-with-regards-to-a-support-vector-machine

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems, 29(7), 1645-1660.

Gungor, V. C., Lu, B., & Hancke, G. P. (2010). Opportunities and challenges of wireless sensor networks in smart grid. IEEE transactions on industrial electronics, 57(10), 3557-3564.Jin. (2018). Automated mobile sensing: Towards high-granularity agile indoor. *Elsevier*, 268-276.

Joachims, T. F.-N. (1998). *Cutting-Plane Training of Structural SVMs.* New York: Cornell University.

Joshi, R. (2016). Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures. Exsilio.

Huh, J.-H. (2017). *An Indoor Location-Based Control System Using Bluetooth Beacons for IoT Systems.* Sensors.

KY-038 Microphone sound sensor module. (2018, 11 27). Retrieved from Sensor Kit Support: http://sensorkit.en.joy-it.net/index.php?title=KY-038_Microphone_sound_sensor_module

Laput, Z. H. (2017). Synthetic Sensors: Towards General-Purpose Sensing. *GIoTTo Expedition.* Pittsburgh.

Lee, W. (2016). Instance categorization by support vector machines to adjust weights in AdaBoost for imbalanced data classification. *Elsevier*, 92-103.

Lim, W. N. (2007). A Real-Time Indoor WiFi Localization System Utilizing Smart Antennas. *IEEE Transactions on Consumer Electronics*, 618-622.

Liu, C. H., Yang, B., & Liu, T. (2014). Efficient naming, addressing and profile services in Internet- of-Things sensory environments. *Ad Hoc Networks, 18*, 85-101.

Longadge, R., & Dongre, S. (2013). Class imbalance problem in data mining review. arXiv preprint arXiv:1305.1707.

Maker.IO. (2018, 11 2). *Raspberry Pi 3 Model B+*. Retrieved from Maker.IO: https://www.digikey.ca/en/maker/blogs/2018/meet-the-new-raspberry-pi-3-model-b-plus

Melamed, R. (2016, May). Indoor localization: Challenges and opportunities. In 2016 IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft) (pp. 1-2). IEEE.

Mell, P., & Grance, T. (2009). Effectively and securely using the cloud computing paradigm. NIST, Information Technology Laboratory, 2(8), 304-311.

NodeMCU. (2018, 12 2). *NodeMCU Documentation*. Retrieved from NodeMCU: https://nodemcu.readthedocs.io/en/latest/en/modules/gpio/

Ortega, J. G., Han, L., Whittacker, N., & Bowring, N. (2015, July). A machine-learning based approach to model user occupancy and activity patterns for energy saving in buildings. In 2015 science and information conference (SAI) (pp. 474-482). IEEE

Plageras, A. P. (2017). Efficient IoT-based sensor BIG Data collection–processing and analysis. *Elsevier*, 249-357.

PohLam, K. (2009) Occupancy detection through an extensive environmental sensor network in an open-plan office building. *International IBPSA Conference* (pp. 1452-1458). Glasgow: International Building Performance Simulation Association.

Rawat, D. B., Bista, B. B., & Yan, G. (2014). Introduction to Mobile and Wireless Communications Networks. In Security, Privacy, Trust, and Resource Management in Mobile and Wireless Communications (pp. 1-10). IGI Global.

Ryo, S. H. (2016). Development of an occupancy prediction model using indoor. *Elsevier*, 1-9.

Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, *50*(1), 30-39.

Schulte, D. (2018). *BETTER BUILDINGS FOR A LOW-CARBON FUTURE.* Ottawa: House of Commons.

Shahzad, K., & Oelmann, B. (2014). A comparative study of in-sensor processing vs. raw data transmission using ZigBee, BLE and Wi-Fi for data intensive monitoring applications. 2014 11th International Symposium on Wireless Communications Systems (ISWCS), 519-524.Sunrom. (2018, 11 27). *Micro PIR Motion Detection Sensor AM312*. Retrieved from SUNROM: https://www.sunrom.com/p/micro-pir-motion-detection-sensor-am312

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637-646.

Sundmaeker, H., Guillemin, P., Friess, P., & Woelfflé, S. (2010). Vision and challenges

for realising the Internet of Things. Cluster of European research projects on the internet of things, European Commision, 3(3), 34-36.

Wang, W. (2018). *Occupancy prediction through machine learning and data fusion of the environmental sensing and Wi-Fi sensing in buildings.* California: Berkeley Lab.

Wu, M., Lu, T. J., Ling, F. Y., Sun, J., & Du, H. Y. (2010, August). Research on the architecture of Internet of Things. In 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE) (Vol. 5, pp. V5-484). IEEE.

Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., ... & Jue, J. P. (2019). All one needs to know about fog computing and related edge computing paradigms: A complete survey. Journal of Systems Architecture.

CURRICULUM VITAE

Candidate's full name: Alec Lloyd Parise

Universities attended (with dates and degrees obtained): University of New Brunswick (Fredericton Campus), Graduated with a Degree in Geomatics Engineering May 20th, 2017.

Publications:

Black, K., Wachowicz, M.,Parise, A.
(2017). Using Bi-Partite Graphs to Cluster Complex
Networks. IEEE Big Data Conference. Boston: IEEE Big Data Proceedings.

Parisé, A., Doak, C., Fitzgerald, M., AL-Yazidi, T. (2017). Indoor Navigation and Real-Time Recommender System using Bluetooth Low Energy Beacons. University of New-Brunswick, Fredericton, Undergraduate Engineering Design Symposium.

Parisé, A., Manso-Callejo, M., Coa, H., Mendoca, M., Kohli, H., Indoor Occupancy Prediction using an IoT Platform. IoTSMS 2019. Granada, Spain: The 6th IEEE International Conference on Internet of Things: Systems, Management and Security (IOTSMS 2019).