The University of Texas at TYLER The Graduate School

University of Texas at Tyler
Scholar Works at UT Tyler

Electrical Engineering Theses

Electrical Engineering

Spring 4-15-2019

SMART-DETECT: AN IOT BASED MONITORING SYSTEM FOR OIL LEAK DETECTION

Youssef Mohamed Baiji
Univerisity of Texas at Tyler

Follow this and additional works at: https://scholarworks.uttyler.edu/ee_grad

Part of the Electrical and Electronics Commons

Recommended Citation
Baiji, Youssef Mohamed, "SMART-DETECT: AN IOT BASED MONITORING SYSTEM FOR OIL LEAK DETECTION" (2019). *Electrical Engineering Theses.* Paper 41.
http://hdl.handle.net/10950/1302

This Thesis is brought to you for free and open access by the Electrical Engineering at Scholar Works at UT Tyler. It has been accepted for inclusion in Electrical Engineering Theses by an authorized administrator of Scholar Works at UT Tyler. For more information, please contact tbianchi@uttyler.edu.

The University of Texas at TYLER The Graduate School

SMART-DETECT: AN IOT BASED MONITORING SYSTEM FOR OIL LEAK DETECTION

by

YOUSSEF M. BAIJI

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering
Department of Electrical Engineering

Prabha Sundaravadivel, Ph.D., Committee Chair

College of Engineering

The University of Texas at Tyler
May 2019

The University of Texas at Tyler
Tyler, Texas

This is to certify that the Master's Thesis of

YOUSSEF M. BAIJI

has been approved for the thesis/dissertation requirement on

April 2, 2019

for the Master of Science in Electrical Engineering degree

Approvals:

Thesis/Dissertation Chair: Prabha Sundaravadivel, Ph.D.

Member: Mukul Shirvaikar, Ph.D.

Member: Chung Hyun Goh, Ph.D.

Chair, Department of Electrical Engineering

FOR JK

Dean, College of Engineering

Acknowledgments

All praise is to the Al-Mighty Allah, The Merciful and Beneficent for the strength and blessing throughout the entire time until the completion of this thesis. Peace be upon our prophet Muhammad S.A.W, who has given light to humanity.

I would like to express my gratitude and appreciation to my thesis advisor, Dr. Prabha Sundaravadivel, for her constant guidance and continuous support throughout my research process, her knowledge, and profound expertise helped me finish this thesis.

I would also like to thank my thesis committee members, Dr. Mukul Shirvaikar, and Dr. Chung Hyun Goh for their kind support and willingness to assist with my thesis. I want to express my gratitude to Dr. Hassan El-Kishky, for giving me the opportunity to conduct research and for giving me the position of research assistant. I would like to thank all staff members, colleagues, and friends in the Electrical Engineering Department for their kind support. Finally, I'm extremely pleased to have received enormous support from my family and want to express my sincere appreciation and gratitude to them in particular, to my sister Reem, Bashar, and Jasem who supported me and helped me financially throughout my studies, my Mom and Dad for their constant encouragement. My appreciation also goes to my Fiancé and my friend Selorm for their support. I will always remember the kindness and cooperation from all the people mentioned above.

Table of Contents

List of Tables

Table of Figures

Abstract

SMART-DETECT:AN IOT BASED MONITORING SYSTEM FOR OIL LEAK DETECTION

Youssef M. Baiji

Thesis Chair: Prabha Sundaravadivel, Ph.D.

The University of Texas at Tyler
April 2019

In the past couple of years, the oil and gas industry is aiming to reduce it's day-to-day costs due to reasons such as reduction in oil prices, mass overproduction and so on. This has the Oil and Gas industries aiming for innovative ways to reduce costs and minimize nonproductive time. In order to accomplish this goal, oil companies need to improve and control measurements with more reliable but relatively cheaper systems. One of the methods is using Internet-of-Things (IoT) based monitoring systems which can help in remote monitoring. IoT is offering better solutions for oil and gas companies to reduce potential failures and downtime by achieving a better and faster method to acquire information efficiently. A real-time stream of data can minimize the need for human intervention in the oil field in case of a catastrophe by reducing the risk of a hazard, saving time, and increasing the environmental pollution control. IoT can be a vital transformation for the Oil and Gas industry. The aim of this thesis is to validate and prove that IoT solutions can be feasible in the oil industry specifically in the pipe leak detection solution ,by building a prototype that operates on low power communication protocol (LoRa®) and conducting experimental procedures on an actual pipe using water instead of oil due to practical difficulty of using oil for the experiment.

Chapter 1 Introduction

1.1 The Internet of Things

The Internet of Things is defined as a network of interconnected things, in whicheach connected device can communicate with each other wirelessly. IoT-based solutions provide real-time decisions, boost performance, and improves product quality. Deploying IoT sensors across the operational field has the potential to improve the efficiency in upstream, midstream, and downstream operations. Today most oil and gas companies employ personnel to react to problems in tank levels, collecting pressure and flow rates from sensors used in oil wells, on an hourly basis to respond to potential issues. In such scenarios, the speed of operation gets delayed and time to react for potential problems is increased. With the deployment of IoT sensors across the machinery, a huge amount of data generated through these sensors enable remote monitoring of the operation and improve the overall performance of the system.

The oil and gas industry supply chain can be divided into three main sectors: Upstream, Midstream, and Downstream. The Upstream sector is associated with exploratory drilling and production of potential crude oil, hydrocarbon reserves, and natural gas. The Midstream sector is associated with transportation of crude oil from production wells to refineries via pipelines, trucks, and tanks, which is then delivered to the downstream sector that is responsible for the process of refining the crude oil. At the downstream sector, the products are derived from crude oil and natural gas to be marketed and distributed. IoT based solution can take a major role in improving these sectors particularly in the Midstream sector ranging from detecting the physical presence of oil and gas pipelines, pipeline leak detection, and monitoring pressure variation in tanks, pipelines, and wells remotely.

IoT-based systems can reduce the cost of operations in several ways. The most important is the implementation sector, where the implementation cost for IoT based solutions is relatively cheaper than other common solutions. Real-time monitoring, decision-making capabilities, acquiring data wirelessly and immediately receiving this data to the cloud makes it far superior and will help decrease the probability of failure that causes non-productive time. These capabilities can result in more revenue for the Oil and Gas industry.

1.2 Role of Communication Protocols in IoT

Nevertheless, implementing an IoT-based monitoring system specially in the rural areas and deserted oil fields require the sensor nodes to be battery powered and works for extended period of times without requiring a battery change or maintenance and in the same manner require a great wireless communication medium which has long range, better obstacle penetration, and long-distance reception. Most of the common communication standards available cannot fit this task entirely. Local area network solutions like Wi-Fi can be a great solution for indoor and short-range applications. Bluetooth and Zigbee work for even shorter-range applications. However, all these solutions have a disadvantage of poor coverage and short distance communication capabilities, and also can consume power which makes them unreliable for application the needs to operate on batteries for several years. Moreover, cellular networks are ubiquitous and have a long range, but the problem of draining the battery rapidly makes it unsuitable for an application that requires to be battery powered. Low power WAN has long range, low cost, and long battery life but it is not good for high data rates. However, to create a leak detection system,, there is no need for a large stream of data, short and small messages measured in bytes containing the raw measurement separated by seconds can fit the task. Consequently, Low Power WAN like LoRaWAN® can be used to build a leak detection system and can accomplish the goal. One of the advantages of this network is that it enables sensors to be further apart especially in outdoor communication as it has a range that exceeds 5 miles. Figure 1.1 depicts the advantages and disadvantages of each category.

Figure 1.1. The advantages and disadvantages of communication protocols

Implementing the IoT solution in the mid-stream can improve the quality of operation by analyzing the acquired data from the field using LoRaWAN® enabled sensors ranging from Flow Meters, pressure, and temperature sensors. These sensors can measure raw data, and the data acquired can be sent through LoRaWAN® network to the cloud where it can be analyzed, and intelligence plus decision-making capability can be added at the front end. Besides, LoRa® protocol is a bi-directional communication standard which offers the ability to send downlink data to the nodes; this will offer versatility and mobility for the designed application. While the security is a must, LoRaWAN® offers end to end security such as encrypted payloads and additional security keys to protect the integrity and usability of the network.

1.3 Oil Leak Detection

Oil leaks can occur from pipe corrosion, expansion due to heat, high pressure, faulty connections or several other reasons. Oil Leak detection systems cannot predict a leak occurring before it happens, can however limit potential damage and reduce the risk of oil spills. The system needs to react as soon as possible when a leak occurs and should have the ability to localize the leak. The timing is very important to consider a leak detection system to be effective because the sooner the leak is detected, the less damage the oil spill can impact on the environment and less money can be spent on cleaning up the spill. The total impact of oil spills is catastrophic, and it can cost a fortune to clean up. While the common methods used to collect data at remote locations are feasible solutions, it is not cost effective and requires human intervention in case a leak occurs, where other methods involve operators to react to a certain issue and collect the data manually. In this case, the response time will be very slow and can cost companies time and money to fix the problem. Companies can decrease their application response time with a Leak Detection system that deploys IoT technology with an analytics platform that utilizes machine learning algorithms as a central output that can produce the desired result and identifies the problem in real time. Moreover, IoT based systems can promote safety which is a very essential aspect in the oil field by reducing the manual tasks involving collecting data in hazardous locations.

1.4 Smart-Detect System Overview

The proposed novel solution consists of sensing nodes with the communication protocol, LoRa®, which is short for low power long range protocol. LoRa® based sensors are a great solution for connecting distant battery-based sensors due to excellent range and low power consumption. Because it uses the 915 MHz part of the spectrum, it has great coverage and penetration capability. This cost-effective protocol is integrated along with flow-meter and temperature sensors to form the Smart-Detect sensing nodes. Figure 1.2. shows an overview of the proposed Smart-Detect system.

Figure 1.2 illustrates the working mechanism for which the data is acquired from the sensors, the basic introduction for the system should start by defining the communication protocol. LoRa® standard is the physical layer for the wireless communication utilized to create a long-range interface. For low power communication, many wireless standards use frequency shift keying (FSK) modulation to attain low power communication. However, in order to achieve low power communication with extended range, LoRa® is designed to utilize chirp spread spectrum modulation which has the peculiar of FSK modulation but vitally increases the range of communication. Chirp spread spectrum modulation has been used for a couple of decades such as in military communications and outer space transmission. Because it has a low power and long-range capability the chirp spread spectrum was embedded in the design of the LoRa® protocol. The advantages of chirp-based modulation are that it operates below the noise floor which allows it to be more robust to interference and noise. Additionally, it has ability to have simultaneous occupation on the same channel in the same time without interference by using the Adaptive Data Rate (ADR) where two different channels can operate on different data rates that allows them to not interfere with each other creating an increase in the capacity of number of LoRa® gateways that can be utilized. The LoRaWAN® network server is responsible for managing the data rate setting and the output transmission power. The system starts with event producers where sensors gather raw analog parameters from the pipe and send it to the microcontroller, and the microcontroller has a LoRa® Transceiver which can relay LoRaWAN® messages over the LoRa® radio protocol these messages are sent in the form of packets to the gateway. These packets then forward by the gateway in the concentrator level.

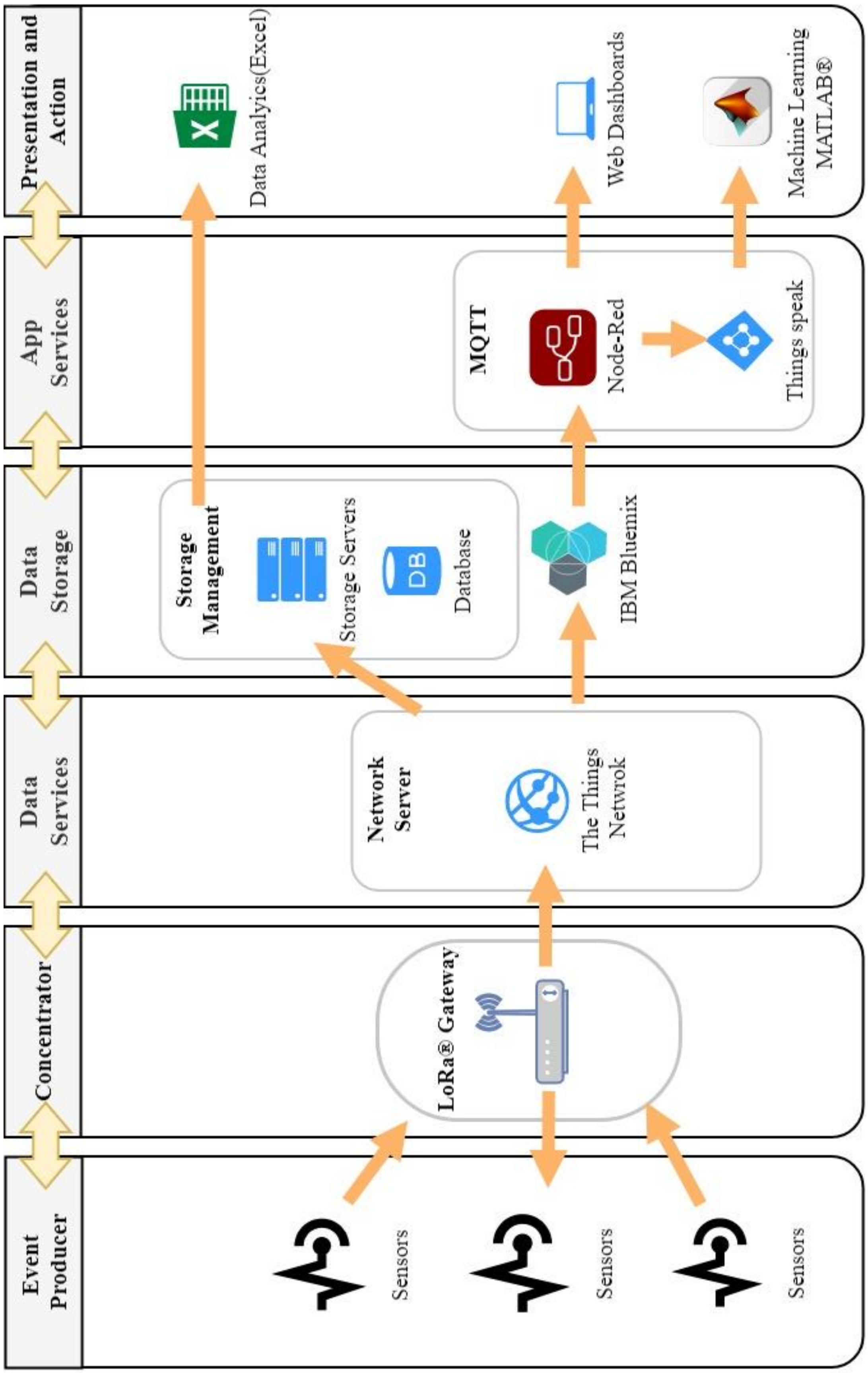

Figure 1.2. Overview of Smart-Detect system

The gateway is hardware that can be connected to any network. Some are connected by ethernet; others use WiFi or even GPRS connections to the Internet. This backhaul connection enables the gateway to forward the packets to the Network server where it can be processed. The network server (The Things Network) has a Router, Broker and Handler. The Router is responsible for scheduling transmissions and administrating the gateway's status. Each Router is connected to one or more Brokers. In the network server, the Broker is the most crucial part to couple a device to an application, forward uplink payloads to the right application and forward downlink payloads to the correct Router. The Handler uses Message Queuing Telemetry Transport (MQTT) protocol which is designed for wireless connection when the requirement is to send and receive small data packets, and the network bandwidth is limited. It has the potential to handle the data of one or more applications by connecting to a broker and acquiring the registered applications and devices information. Besides, the Handler is responsible for encryption, decryption and forwarding the payloads to the application level. From The Things network (TTN) the gathered data can be stored or processed by integrating TTN with a specific cloud solution depending on the nature of the proposed application. Nevertheless, a cost-effective solution is preferred for experimenting and designing a prototype application, at which point (TTN) was integrated with Node-RED a programming tool developed by IBM to offer the ability to connect hardware devices and online services in the form of a flow where range of nodes can be deployed to create a JavaScript functions making it a simple way to forward data to The Things Speak platform. The Things Speak is an IoT cloud developed by MathWorks in which IoT sensors can be deployed to monitor and procure a real-time analysis of the data by adding machine learning algorithms (ML) in the form of MATLAB® codes at the cloud edge to add smart decision-making capability to the application. The Machine Learning algorithms premise is to build a statistical model to add accuracy for predicting an output without explicit programming, which is essential to design a reliable Leak Detection system to obtain faster reaction time and accurate results.

So far, the LoRa® technology, built by LoRa® Alliance is relying on unlicensed spectrum to provide the communication for IoT services. The free part of the spectrum is open source and help developers innovate new ideas on lower cost. However, by the next couple of years more deployed applications can cause more interference and congestions which might encourage the LoRa® Alliance for licensing the spectrum for a more reliable network.

1.5 Organization of Thesis

This thesis is divided into five chapters. Chapter 1 is an introduction to the Internet of Things and a brief overview of its importance in today’s world. Chapter 2 gives an intensive literature survey about conventionally used methods for leak detection. Chapter 3 provides information about the communication protocol and the experimental setup of the proposed system. Chapter 4 contain the acquired data and the analysis of the results. Finally, discussion of the results also the conclusions and future work ideas are presented in Chapter 5.

Chapter 2 Literature Survey

This thesis undertakes the design and development of a novel solution of using IoT sensors and machine learning algorithm to build an IoT based leak detection system using low power communication medium.

A comprehensive literature survey on leak detection techniques, the identification of the occurrence of oil spills, the use of different communication protocols enabled sensors in the oil and gas sector, and the machine learning (ML) solutions for building a leak prediction model. All the mentioned solutions that are widely used are discussed in this chapter.

The literature survey of various ways of leak detection such as inverse transient analysis, time domain analysis, frequency domain analysis, and the negative pressure method. Also, the techniques of oil spill detection like microwave remote sensing, imaging and the use of wireless sensors to monitor the oil spill. Lastly, the introduction of various communication protocols enabled sensors implemented in the oil and gas sector and the integrated solutions of machine learning statistical models. All are discussed in the later sections of this chapter followed by the limitations of these techniques.

2.1 Leak Detection Methods

The propositions and hypothesis introduced by researchers to detect leakage instantaneously and the various approaches to tackle the problem from different aspects are presented in the upcoming section.

Hardware-based methods of leak detection have been studied in [1] by using a controller called sliding motion, and two pressure sensors mounted on both ends of the pipe. When a trajectory slides on the surface of the pipe, a response is received when a fault occurs, or the motion on the sliding surface is not stable in addition to undertaking further analysis by finding the difference between expected values and sensor readings.

Frequency domain pressure signal analysis was conducted in [2] in which by using the wavelet transform the leak reflected pressure wave could be found and this can indicate the leak location. Time domain analysis was studied in [3] by analyzing the opening and closing of a valve and the results are then transformed by using Fast Fourier transform to the frequency domain. The discrepancy in the frequency response was obtained by analyzing the amplitude of resonant pressure in case of a leak and no leak.

An experimental model was built by implementing Wavelet Discrete Transform of the resulted vibration signal using vibration sensors have been studied in [4]. A method where vibration sensors generate a series of vibration signals detected by another vibration sensors located at the location of the pipe then a processor process and determine the average power of the signal over a predetermined time. A method based on a negative pressure wave to detect the presence of natural gas in a pipe was developed in [5]. The signal generated from the negative pressure wave phenomena occur in case of a leak is collected from the pressure sensors that are installed at both ends of the pipeline, and further analysis was determined.

Leak detection by using optical fiber temperature measurements based on Raman scattering method in heating pipes was the focus of [6]. The leakage causes a rise in temperature of the soil around the pipe; this increase in temperature helps indicate the occurrence of the leakage. Finally, a Review on various methods for leak detection with pressure measurements and the advantages and disadvantages of each method were studied in [7]. Methods such as hydrostatic pressure testing, inverse transient analysis, transient steady state, transient damping, inverse resonance, pressure-flow deviation, negative pressure wave, and pressure residual vector method.

2.2 Oil Spill Detection Methods

Researchers are aiming to find a solution for post leakage occurrence to limit the effect and causes that may pollute the environment and could cost companies a fortune to clean up the oil spill. The various oil spill detection methods conducted by many researchers will be discussed below. Offshore oil spill monitoring and detection method was discussed in [8]. This paper is focused on building an offshore petroleum cyber-physical system (CPS), which is based on simulation to find the approximate location of the leak source using data from remote multi-sensing technology.

[9] proposed a new High-resolution COSMO - SkyMed Aperture Radar (SAR) images technology which is a better version of high-resolution SAR for oil spills automatic detection. This method improves the ability to detect the spills in small basins and near coast which resolves issues that encountered the previous technology. [10] has developed a wireless sensor device that can sense, process, and transmit the location and thickness information of an oil spill. The wireless sensor node was placed on oil tankers and offshore drills; it can be thrown into a spill after it happens. The research suggested two approaches, first was a light sensor array which can sense the variation in intensity and the propagation of light in a specific medium. The second was conductivity array which depends on sensing the conductivity of the medium difference; for instance, seawater has high electrical conductivity while oil has low electrical conductivity.

[11] recommended a prediction system based on a feedback control system which was founded on the theory of Dynamic Data Drive Application System (DDDAS). The system enables a combination of monitoring and simulation of oil spills. In this system, oil spill detection can be achieved by using numerical modeling and remote sensing data, where multiple simulations of different scenarios of various remote sensing approaches can help improve the accuracy of the prediction.

[12] suggested measuring the radiation emitted from the objects using Microwave radiometer. The research depicts the ability of the antenna to sense the radiant energy from the oil spill when in the position of the line of sight to the object. The discrepancy of brightness temperature between the contaminated seawater with surrounding clean water can indicate the oil spill by analyzing the quantifiable imagery of pre-defined volumes of mineral types of the oil spill on the sea surface.

[13] proposed a novel methodology in which the measurement of the extent and thickness of oil spills over the sea surface is conducted using a Special Sensor Microwave Imager (SSM/I) and Advanced Microwave Scanning Radiometer for EOS (AMSR-E) satellite. This is done in order to find some a relation between brightness temperature at various frequencies to distinguish the contaminated locations from the clean ones.

[14] researched Multitemporal optical remote sensing images and their ability to give more insight into oil spill detection. Analysis of these images gives more thorough and precise results. [15] proposed a methodology using laser fluorosensors to monitor and sense the oil spill. Laser flurosensors like Scanning Laser Environmental Airborne Fluorosensor (SLEAF) has the characteristic of distinguishing the oil from the background which can be either water, ice, or snow. The data from the sensor then can be analyzed in real-time by a real-time geographic information system (GIS) to detect the spill.

[16] employed the developed model of oil-water contrast by using imaging system called Video-Rate Infrared (IR) Multispectral imaging system making use of the difference in solar heating and thickness of oil sleek and water. The implemented system aims to achieve better accuracy at a lower price. [17] have researched real-time monitoring of oil spillage in the marine environment using an optical fiber-based sensor. This sensor can be employed remotely and can detect minor variations of oil adulteration levels within the water.

2.3 Various Communication Protocols Deployed in the Industrial and Oil & Energy Sector

In order to investigate a new communication protocol to fulfill the purpose, a literature survey must be held to examine the various communication mediums implemented by several researchers in the industrial, oil, and energy sector. Related research work is introduced in the section below. A new Wireless Geophone Network is presented by [18] to replace the conventional on-shore cable networks used in seismic exploration in which geophones are used to measure the backscattered field waves. [19] suggested the need to deploy wireless sensors underwater to monitor the production process in order to control and manage the flow of production and prevent potential failures.

[20] conducted a study on deploying Wireless Sensor Networks to remotely monitor reservoirs, equipment conditions and pipelines to detect natural gas leaks, corrosion, and H2S. An experiment was held using Zigbee wireless sensor network in industrial applications by [21] to enhance and provide better management for industrial automation facilities.

An innovative approach to managing unnecessary excessive waste of power was employed by [22] using LoRa® based Smart metering Technology to remotely monitor the user consumption and to provide the right amount of power based on the consumption to meet the demand of users. A monitoring system for Renewable energy based on LoRa® Technology was introduced by [23]. The idea is to construct a cost-effective system to monitor energy use IoT enabled sensors and IoT clouds.

[24] researched creating a LoRa® architecture that enables electric vehicles to communicate with charging stations. The charging station is powered by renewable energy source and integrated with long-range communication protocol that enables them to update the users of the current status of energy storage and occupancy of other electric vehicles. [25] developed an underground wireless sensor network to monitor the pipeline condition called Smart-Pipes. It is based on monitoring the pressure of the pipeline using Force Sensitive Resistor (FSR) technology. The researcher anticipated the need for low power wireless sensor node to detect the leak. As a result, the design was tested and developed in the field and laboratory.

A Hall Effect Flow-Meter sensor was used by [26] to control and monitor the water flow from a web server. The researchers aimed to build a pipe network and examine the way a leak affect the readings of the flow rate. This paper proposed in [27] aims to monitor and gather data from oil refineries using a more robust version of Supervisory Control and Data Acquisition systems (SCADA). SCADA systems have many wires connected actuators and sensors, also equipped with a special computer architecture that uses networked data transmission, remote terminal units, PID controllers, and programmable logic devices. The Researcher approach was to simulate the system in order to receive real-time information about oil pipelines and tanks. Furthermore, reducing the time required to respond to malfunctions and faults.

2.4 Machine Learning

The evolution of the internet of things in recent years helped boost the popularity of machine learning. Machine learning can create analytical models that enable algorithms to learn from the data acquired from IoT devices continuously. Several algorithms can solve different problems; each algorithm has different characteristics that with the right approach and the knowledge of the mathematical foundation behind each algorithm the task can be accomplished. Before the implementation of machine learning, an analyst must analyze the problem and identify the type of algorithm to use. Accordingly, supervised and unsupervised learning describes two ways that a machine learning algorithm acts on a set of data. The main difference is that in supervised learning the expected output is known. The algorithm gives a relationship between the input and the output, or simply the ability of the algorithm to learn from the training data and the process is to guide it to obtain the right output. In contrast in unsupervised learning, the output is unknown and with no available training data set. Unsupervised learning can solve complex problems only with the availability of input data, which gives the computer the ability to learn by itself. In leak detection applications the input data and the expected labeled output data are available by experimentation or by simulation. Hence the supervised learning can be more suited for this kind of applications. Algorithms that lie under the supervised learning category such as Linear Regression, Support vector machine, Logistic Regression, and Anomaly Detection, all have the potential to be successfully implemented. Logistic Regression and Support Vector Machine are classification algorithms while Linear Regression and Anomaly Detection are Regression algorithms or algorithms that rely on prediction. To build a predictive model that depends on the real-time acquisition of data and add some intelligence to the application of leakage detection, researchers explored several supervised learning algorithms to achieve the target. The different mentioned algorithms used for this sort of application are mentioned in the next section.

[28] presented a model that introduces a way to detect and locate leaks in pipelines using simulated data from flow and pressure sensors acquired from EPANET software, in which Artificial Neural Network (ANN) and Support Vector Machines (SVM) were modeled, and resulted outputs were compared to illustrate the advantages and disadvantages of each model.

[29] has conducted another Research to implement the Support Vector Machines (SVM) classification model in order to detect leakage in water distribution network using simulated data from EPANET software. [30] took a different approach by implementing the Moving Windows least square support vector machine algorithm (MWLS-SVM) model

in addition to the negative pressure wave method for leakage detection. The model constitutes of improving the training speed of the training set by applying the sum of square errors.

Furthermore, the next section consists of literature survey regarding using logistic regression to build a model that construct a binary hypothesis based on similarity to detect changes between objects. In [31] the researchers utilized logistic regression to compare the detected objects in the image and the database information. Using the maximum likelihood method, the parameters of logistic regression were estimated to obtain the similarity between the database and the synthetic aperture radar images (SAR).

2.5 The Limitations and Disadvantages of Different Conventional Methods Used for Leak Detection

All the mentioned leak detection methods that were proposed by researchers even though they have many advantages however various limitations face most of these methods, for instance, using the sliding trajectory cannot give a precise result in addition to high cost when considering pipelines that go for miles. Other methods like frequency domain analysis, time domain analysis, and negative pressure wave all these studies depend on simulated data but cannot be proven to be feasible to achieve fast reaction time in practice. Moreover, all these methods would require personnel to react to the problem in case a leak was detected. Other researchers have used vibration sensors and temperature optical fiber that sense the increase in the soil temperature to detect leaks. These methods may be affected by external factors that would generate wrong results and can set false leakage alarms. Furthermore, the main idea of exploring new solutions for this problem is to reduce the cost. However, this goal has not been fulfilled with these methods.

Nevertheless, the need for researchers to explore different solutions post leakage is a must to limit the damage and reduce pollution. Yet, all the mentioned ways for spill detection can be feasible, and spills can be detected using various imaging and fluorosensors methods. Nonetheless, this is only can limit the effects of the spill, and it is not an essential way of avoiding the catastrophe.

Next, employing wired/wireless sensors actuators can be effective to acquire data remotely and to generate data that can help companies have a better idea about the situation in the field. The extensive literature survey shows that these sensors are useful in the industrial and oil & energy sectors can help optimize and enhance these fields. Although many studies about the used communication protocol have drawbacks, for example, Zigbee requires sensors to be few meters apart because of the range limitation, and high-power consumption making applications that must be battery powered difficult to achieve with this kind of technology.

Most companies have the financial capability to use SCADA to acquire data from the field and can help determine the variation of pressure in the pipeline but due to the requirement of SCADA that range from implementing network data transmission and using a different type of controllers can be very costly for all companies. Because SCADA can provide real-time data, but it does not have the feature of decision capability. As a result, researchers are aiming to figure up a way of implementing intelligence to this system, but this will only add up the cost to the process. The limitations of implementing different algorithms for this specific application will be discussed in the next chapter

Chapter 3 System-level Design of Smart-Detect

3.1 System-level overview of Smart-Detect

The proposed novel solution is called Smart-Detect which is based on LoRa® technology. The next section will discuss the mechanism of how payloads are sent through the LoRa® Network and received at the Thingspeak cloud. Figure 3.1 illustrates the flow of data including the encryption and the decryption process embedded in LoRaWAN®.

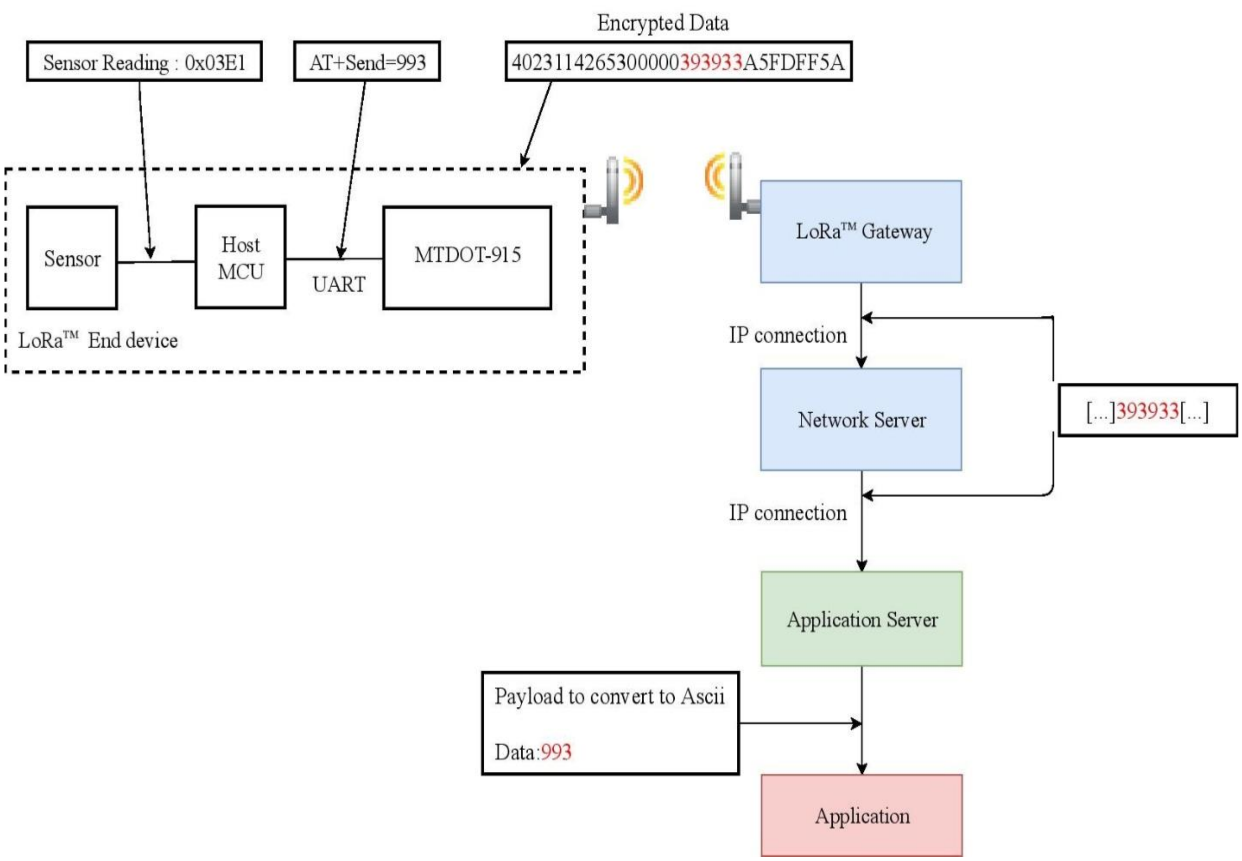

Figure 3.1. The encryption and decryption process of payloads in LoRaWAN®

In the Smart-Detect system, the sensors are soldered and connected to the used microcontroller board (mDoT) Which is an ARM Cortex M4 type. The used sensors are ½ inch Hall effect Flow-meter and a Temperature sensor BME280. All acquire payloads are binary data, and the MCU then processes this data and relays to the Universal Asynchronous Receiver/Transmitter (UART) which is serially sent to the mDoT. The mDot are equipped with a LoRa® antenna where encrypted payloads then transmitted to the used LoRa® Gateway (Multitech Conduit). The encryption Process is particular and will be activated in each end-device, for functionality there are two ways to enable end-devices to join the network securely, the first process usually handled by the Network Server called the over-the-air activation (OTAA) which is used in this system. The second called Activation by Personalization (ABP) where the keys are directly stored in the end device. The OTAA require end-devices to follow certain steps for the successful joining of the network, the successful attempt of joining the network require end nodes to be set up with particular security keys beforehand to begin the joining process. The used Network Server (The Things Network) is responsible for personalizing and providing the following keys, an AES-128 key (AppKey), the application identifier (AppEUI), and the unique end-device identifier (DevEUI). The device sends a MAC (media access control) request to the server called join request. The join request consists of AppEUI and DevEUI. If the device is permitted, and the process is not interrupted by any error the Network Server (TTN) should respond with a join-accept and provide the end-device with the following information a device address (DevAddr), an application identifier (AppEUI), a network session key (NetKey), and an application session key (AppKey).

The LoRaWAN® network is designed to provide versatility to make the user choose and manage different applications on various servers. Because it is an open source platform the user has a choice of which Network Server he wants to use, he can access the Gateway and install a new program for the chosen packet forwarder. The packet forwarder is a program that interacts with the end devices and manages the transmission of packets in the LoRa® network. In this application the packet forwarder was changed instead of using Multi-tech original packet forwarder, the TTN packet forwarder was configured in the Gateway to receive all the packets in the Things Network. Figure 3.2 illustrates the Network Server used (TTN).

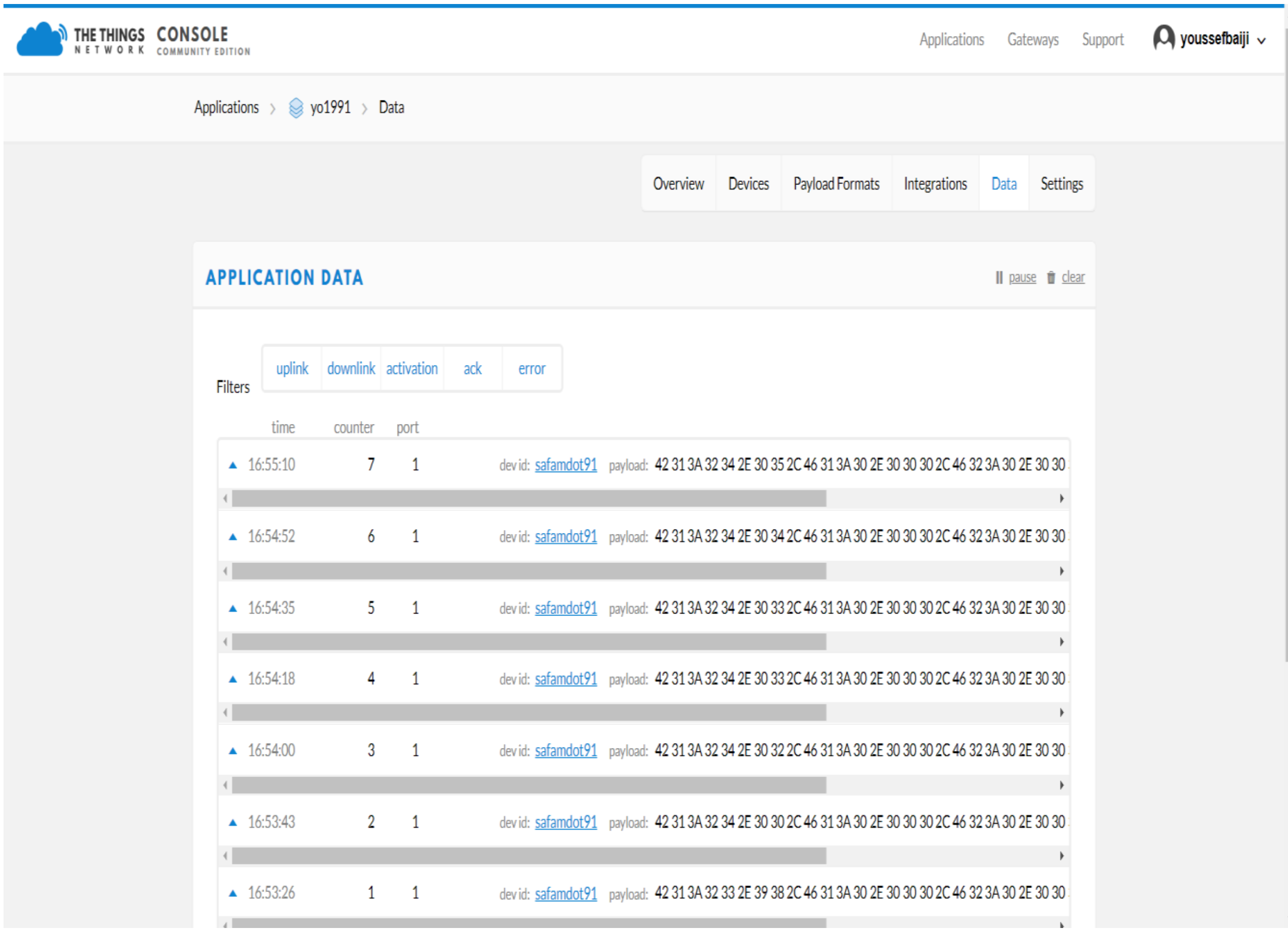

Figure 3.2. The Network Server (TTN)

After the activation, the Network server should receive all payloads from the end device. TTN shows the raw payload in base64 hex-format for convenience. This format can be decoded by using javascript code in the TTN, or it can be decoded in the application server. TTN has a great platform to monitor the coming data, but it cannot be used to act or save the data. However, it provides the users the ability to integrate their applications with different cloud platforms such as Cayenne, Amazon web server, Thingspeak, and others. Smart-detect system is implemented to build a prototype for testing purpose so the need for low-cost solution to store data is needed, so the Thingspeak platform is chosen to fulfill the purpose, but in order to integrate TTN with Thingspeak, a connection must be established by using a middleware connection server called Node-Red. The Node-Red provides an easy way to decode payload messages and to connect platforms using JavaScript palettes by establishing a desired flow to receive data at the desired end. Figure

3.3 depicts the flow in which data is preprocessed and transmitted to the Thingspeak platform.

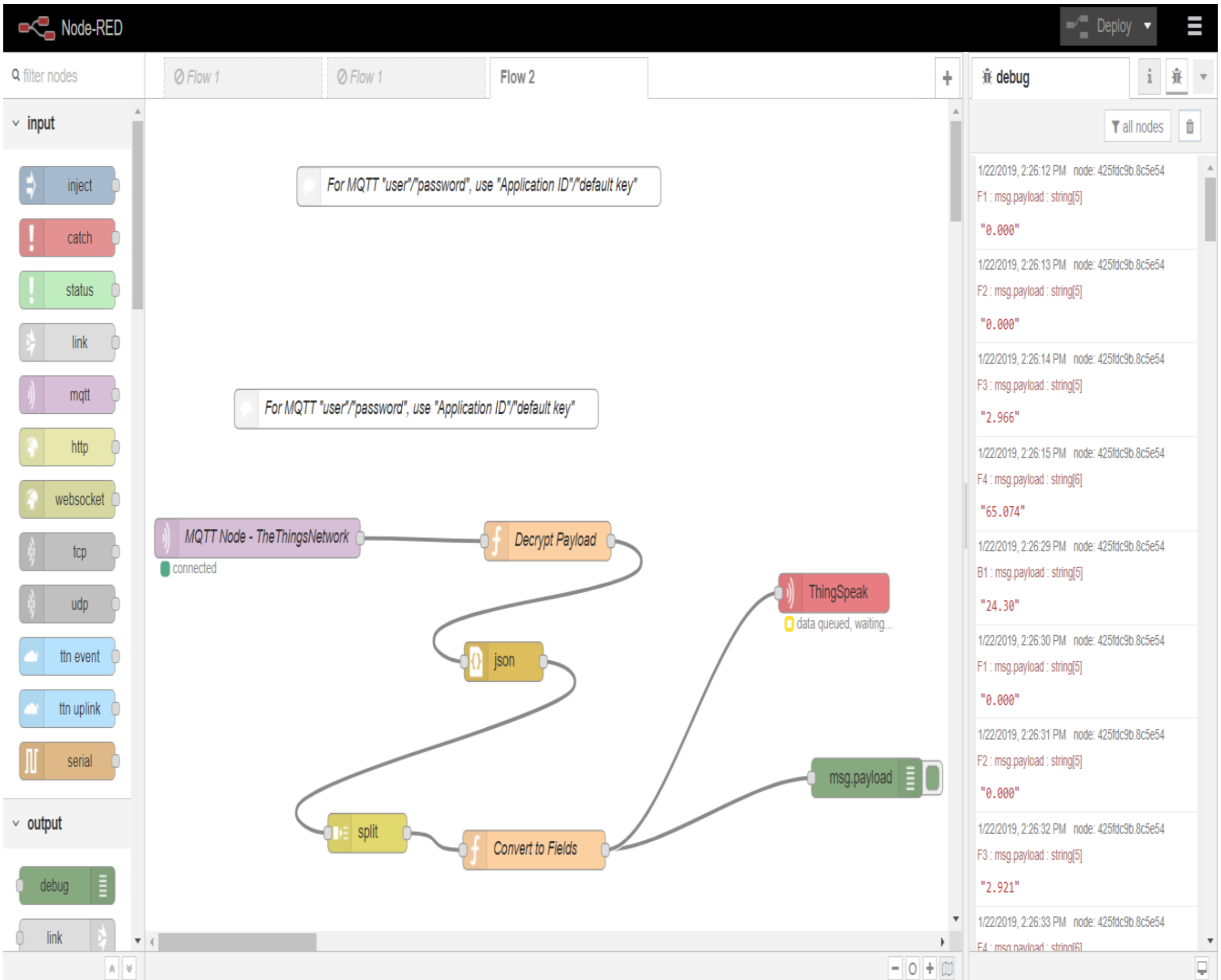

Figure 3.3. The Node-Red flow for integrating TTN with Thingspeak

In this flow, the connections are made by selecting the palette that uses MQTT broker. The MQTT broker enables the connection with the Network server by inputting the required keys in order to receive the forwarded packets from the desired end device. As mentioned, the Things Network shows the payload as Base64 hex-format, so the code has been written to generate a decryption function palette that is used to decode the payload messages to text. The Json palette converts between JSON string and JavaScript object. Usually nodes can have multiple sensors such as the temperature sensor, BME280 that is used to sense temperature, humidity, and biometric pressure, and all three readings are sent in one packet to reduce the amount of bandwidth that will be consumed ,so The split palette is used to split this payload message to separate the stream of messages with the next palette converting them to JSON fields to be read by the Thingspeak palette.

The Thingspeak is about creating a real-time channel with inputting different fields so the data in this channel will be updated constantly if the application is powered on. The Thingspeak provide developers with the ability to test applications by graphing and visualizing the upcoming data fields in charts and provide the ability to save and act on the data by writing a MATLAB code to generate the desired output in real-time. Figure 3.4 shows the Thingspeak used channels.

Figure 3.4. Thingspeak created channels

3.2 Data Acquisition using Smart-Detect "Things"

Before discussing the best-fit machine learning (ML) algorithm that provides the most accurate results. The need for defining the working mechanism of the sensors and the platform used for programming the Microcontroller is discussed in the next section.

The used ½ inch Hall-effect flow sensor measures the magnetic fields using a wheel speed sensor (RPM). It is composed of Hall Effect sensor and a permanent magnet which are placed near a rotating disk. The gap between the sensor and the teeth of the disk is very small so each time a tooth passes near the sensor by the force of the water, it changes the surrounding magnetic field which will generate an electrical square wave pulse with each revolution. These readings are processed in the microprocessor to compute the water flow data by analyzing the wheel rotation count. The processing phase of this sensor node can be accomplished by writing a C/C++ program in the ARM Mbed OS. The Mbed OS is a platform for the Internet of things which includes the required attributes to design and develop a program that is based on ARM Cortex-M Microcontrollers that includes connectivity, drivers, and a program that can be designed to achieve more outputs from the sensor. In Smart-Detect System a C/C++ has been developed to compute the flow rate of the water as follows:

Let K be the pulses per second per liter/min

* F: pulse frequency (1iter/s),
* Q: flow rate (liter/min),
* P: sensor pulses
* T: time since last measurements (s).
* C: capacity in liter/min
* Each sensor comes with different specifications, for this sensor the Capacity is 50 l/min and the K factor is 7.5 Hz per liter/min

$$K = F / Q \tag{3.1}$$

$$F = P/T \tag{3.2}$$

$$Q = (P / T) / K \text{ (l/min)} \tag{3.3}$$

The Volume can be calculated by:

$$V = Q / C \text{ (liter)} \tag{3.4}$$

The mentioned attributes such as current flow rate, current volume, total flow rate, and total volume are all considered to be features in constructing the Machine Learning algorithm. The accuracy of the readings needs to be on point. As a result, a calibration was conducted on the flow meter by changing the correction factor as follows:

Average correction rate = k-factor / corrections over time * total time (3.5)

This can be done by installing the flow meter at the end of a pipe and conducting trials where each time an observation an must be made for instance by filling a bottle of known capacity and checking how much error the volume readings of the flow meter is showing. By setting a correction factor at each trial, the total error would be reduced, and the readings can be accurate.

Another Sensor used in the Smart-Detect system which is the BME 280. It is composed of three sensors that measure the pressure, humidity and temperature. A program also has been written to acquire data from the sensor, in the Mbed OS. However, the only temperature has been chosen tan o be acquired and considered to be a feature for monitoring purpose of water temperature, and the measurement unit is in Celsius. After the debugging, Process, the program can be compiled and can be loaded in the mDot microcontroller using the Universal Developer Kit, taking in consideration configuring the LoRa® connectivity drivers and installing the right keys to have a successful pairing with the Network Server.

3.3 Data Analytics for the Smart-Detect system

A predictive model is based on analysis and the ability of a system to learn from data and to make decisions with minimal human intervention, also, to independently adapting when exposed to a new set of data. In consideration of building an accurate predictive model, the observation is needed for data types and features that can help to decide which method is the best solution for the problem.

Three different approaches have been taken in order to solve the problem of leakage detection. At first, the Support Vector Machine (SVM) algorithm, where the concept is to use to classify the leak from the non-leak data based on a hyperplane, was deployed. This will require gathering real data or using simulation data and labeling the data for instance with one being the (non-leak) and zero is being the (leak), by finding the right hypothesis a decision boundary can separate the two classes. Training this data using SVM should indicate if a leak happens when applying a set of test data. However, after experimental trials, it can be noted that SVM is better for classification if the occurring leakage is large. SVM can falsely identify the leakage if the leak is very small because SVM, in this case, will classify all data as normal (negative) in other words if the trained data is further apart SVM can easily separate them with a decision boundary.

In contrast, anomaly detection works better for identifying anomalies or events that are significantly different from the majority of the data or when training data a few anomalies exist among large regular data points this can occur in case a small leak happens. As a result, a second approach using anomaly detection was conducted, gathering data in normal condition and then introducing the smallest leak in the pipe with labeling the data in the same manner as the first method. In practice, the anomaly detection is based on computing the probability of the trained data where high probability data will be the majority and can be called non-anomalies. Furthermore, by setting a boundary value where any newly introduced data that is outside the boundary can be indicated as an anomaly and with lower probability values. This approach is useful especially by training the data with the smallest leak scenario for guarantying even if a large leak was introduced; the newly tested data will be categorized as an anomaly (leak). Although this sounds promising, but by setting trials an error can occur when the water in the pipe normally jerks back and causing a sudden flow that may introduce a flowrate that is larger than the normal. In this case, the anomaly detection model will categorize this data point as an anomaly as well, causing it to falsely identify an occurrence of leakage. The final approach that was used is Binary Logistic Regression. This method is used as classifier also, where an output value can be modeled as non-leak (one) or leak (zero). The concept behind the method is that the algorithm assumes a distribution from the trained data where it can compute a set of coefficients called (Regression beta values) by using the maximum likelihood estimation. The beta coefficient values dictate the outcome of the predictor. Subsequently, if the beta coefficient is not statistically significant, the variable does not essentially predict the outcome. More simply, the error in the probabilities predicted by the model compared to those in the data is minimized by the beta values. More about these methods will be discussed in the next section.

3.4 Data Analytics using Support Vector Machines (SVM)

The goals of SVM are separating data of two or more classes with a hyperplane. The idea behind SVM is to maximize the margin between the hyperplane and the closest data points; this means that the optimal hyperplane will have the biggest margin. SVM hyperplane can be a line that separate two separable data, the hyperplane can be represented mathematically as a vectorized linear equation:

$$\omega + x \, . \, b = 0 \tag{3.6}$$

The hyperplane then is used to predict by:

$$h(x_i) = \begin{cases} +1, if \;\; \omega + x \, . \, b \geq 0 \\ -1, if \; \omega + x \, . \, b < 0 \end{cases} \tag{3.7}$$

The point below the line will be classified as -1 and the point above the line will be classified as +1. However, how the best fit hyperplane is chosen, the answer is there are many methods to find the optimal separating hyperplane which maximizes the margin of the training data in SVM using optimization kernels like linear and Gaussian kernels.

In the approach used in this thesis, the SVM was used to classify the leak from non-leak data. A hyperplane was chosen to separate the two classes using the linear kernel, and the results were optimal in case the model was trained with marginally large leak data as shown in Figure 3.5. However, when small leak data was used to train the model, it was difficult to separate the two classes because a small leak can have similar data values as non-leak.

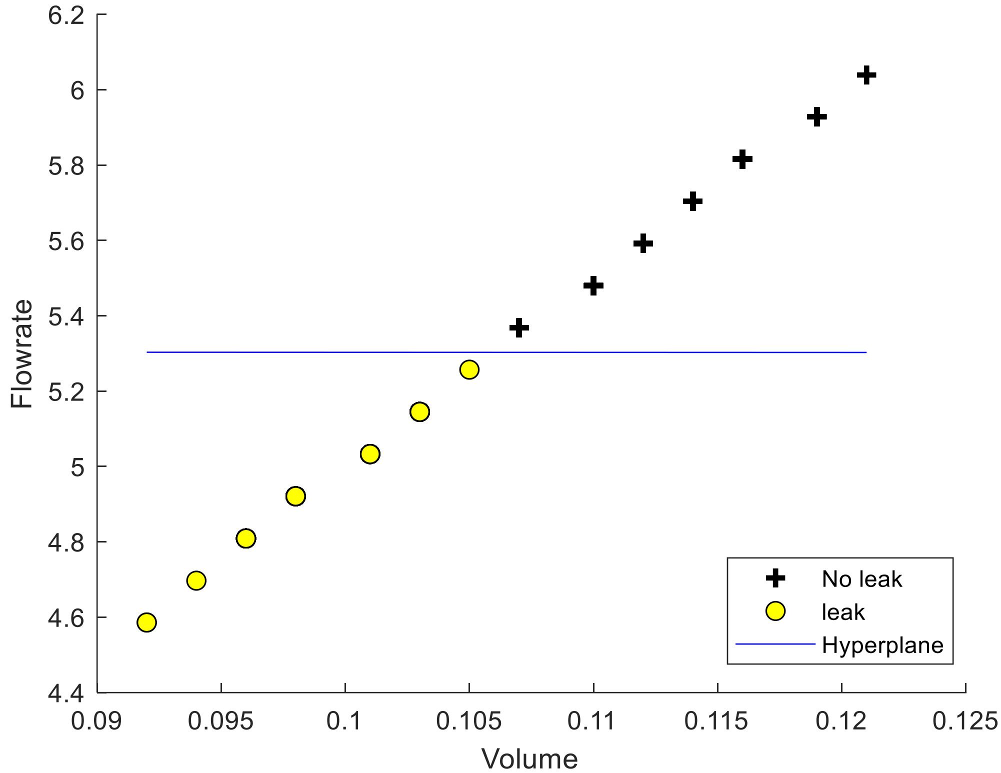

Figure 3.5. Using SVM to separate two classes

3.5 Data Analytics using Anomaly Detection

Anomaly detection is a technique that is called outliers used to identify anomalies that do not conform to expected behavior; this method seemed like a perfect fit to the problem of leak detection. The normal behavior is the water running in the normal flow with few anomalies when the flow spikes down where a small leak is occurring. There are many methods to find the outliers but, in this thesis, the normal distribution method was implemented.

The model should predict if a certain data point is anomalous as following:

$$(x_i) = \begin{cases} anomaly, if\ Px_{(test)} < \varepsilon \\ nonanomaly, if\ Px_{(test)} \geq \varepsilon \end{cases} \tag{3.8}$$

Anomaly detection model can be developed from un label data as it can indicate the probabilities of training data points using the Gaussian distribution by finding the mean and the variance of the training data, the P(x) then can be computed by computing the Gaussian probability.

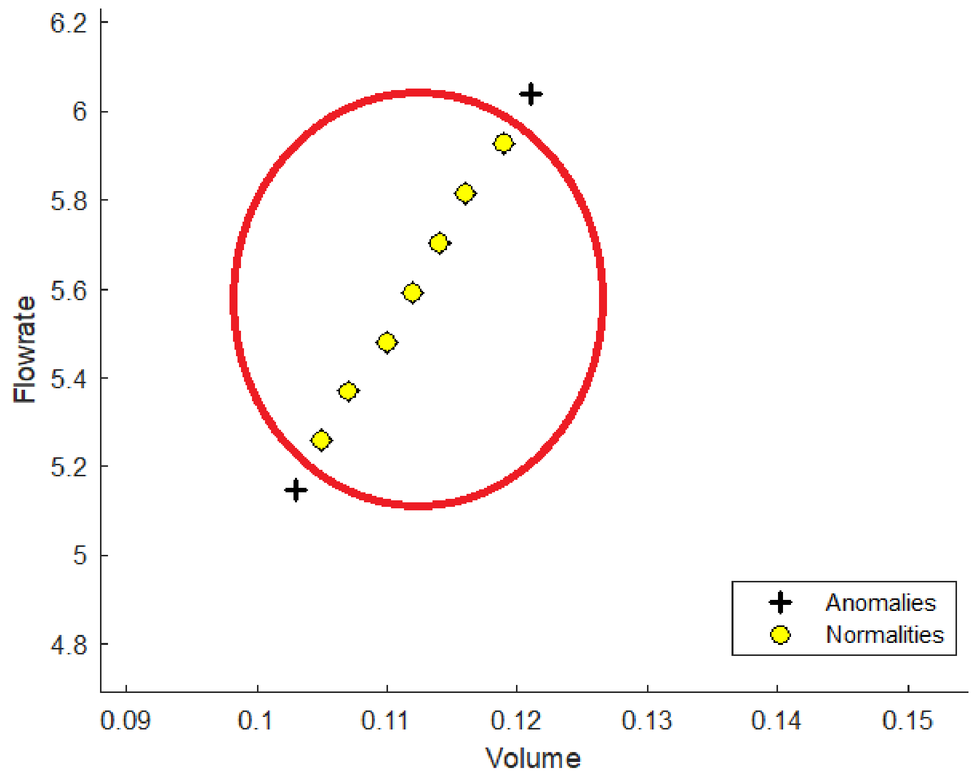

Figure 3.6. Anomaly detection

It can be explained from Figure 3.6 and by observing the computed value of P(x) that any data point newly introduced data point with higher probability is indicated as non-anomalous ($P(x) < \varepsilon$) and data points with lower probability is indicated as an anomaly ($P(x) \geq \varepsilon$). In other words anything inside the decision boundary is with higher probability, and any data point outside the decision boundary is with lower probability, where the decision boundary is constructed by setting a threshold value (ε).

In this thesis this approach was used as shown in Figure 3.6 the model successfully indicates the anomalies that are outside the red circle, the problem was that there are certain points that indicate higher pressure which occurs occasionally from sudden flow, these points shouldn't be flagged as anomalies, as a result this problem will produce errors in the prediction.

3.6 Data Analytics using the Logistic Regression Model

Logistic Regression is simply the logic transformation of linear regression. The algorithm came upon the need to solve the problem that is facing linear regression when the output response needed is binary. Because in linear regression prediction of probabilities usually is greater than one or less than one (negative), this can affect the accuracy of the prediction. Therefore, this transformation will constrain the predicted probabilities to lie between 0 and 1.

$$logit(P) = ln\left(\frac{P}{1-P}\right) \tag{3.9}$$

where:

- P is the probability of occurrence of event Y, $P(Y = 1|X)$
- $\left(\frac{\mathrm{P}}{1-\mathrm{P}}\right)$ is the odds ratio

Logistic regression computes parameters called beta coefficients (b) to predict a logit transformation of the probability of occurrence and the X's are the features that can be used to construct the algorithm.

$$logit(P) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \cdots + b_kX_k \tag{3.10}$$

Hence, the estimated probability is computed in the following equation:

$$P = \frac{1}{1+e^{-logit(P)}} \tag{3.11}$$

This function is called the sigmoid function, and it looks like the following figure.

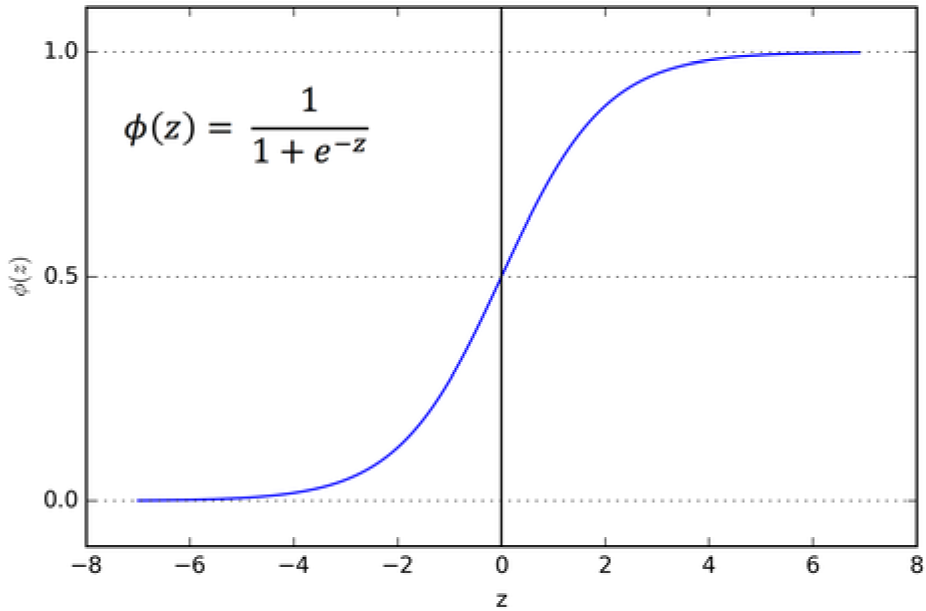

Figure 3.7. The Sigmoid function

The sigmoid function can map all real numbers into the range between 0 and 1. In logistic regression sigmoid function turns the output into a probability which has a range between 0 and 1, where the high probability is denoted as positive numbers, and lower probabilities are denoted as a negative number. Alternatively, in another case an optimal cutoff is chosen for example (P=0.5) where $P \geq 0.5$ classified as 1 (positive class) otherwise 0 (negative class).

Furthermore, instead of choosing parameters that minimize the sum of squared errors (as in linear regression), logistic regression uses maximum likelihood estimation (MLE) where parameters are chosen based on the maximum likelihood of the training examples if the beta coefficient is not statistically significant the variable does not essentially predict the outcome.

In (MLE) the log-likelihood function is computed first by:

$$LL(b) = \sum_{i=1}^{n} y^{(i)} \, log \, \sigma(b^T x^{(i)}) + (1 - y^{(i)}) \, log[1 - \sigma(b^T x^{(i)})] \qquad (3.12)$$

Where:

The likelihood of independent training values is:

$$L(b) = \prod_{i=1}^{n} P(Y = y^{(i)} | X = x^{(i)})$$

And

$P = \sigma(b^T x^{(i)})$ according to the likelihood of Bernoulli

The function in equation (3.8) must be maximized but it's simpler to convert it to a negative log likelihood to turn it into a cost function that can be minimized as illustrated in equation (3.9):

$$J(b) = -\sum_{i=1}^{n} y^{(i)} \, log \, \sigma(b^T \, x^{(i)}) + (1 - y^{(i)}) \, log[1 - \sigma(b^T \, x^{(i)})] \qquad (3.13)$$

The gradient descent is used to minimize the cost function and can be accomplished by taking the partial derivative of the cost function and updating the beta values until the slope of the gradient converges to zero.

$$\frac{\partial J(b)}{\partial b_j} = \sum_{i=1}^{n} [\, y^{(i)} - \sigma(b^T \, x^{(i)}) \,] \, x_j^{(i)} \qquad (3.14)$$

To arrive at the local maximum, small steps must be taken in the direction of the gradient by constantly updating the beta values on every iteration as following:

$$b_j^{new} = \, b_j^{old} + \alpha \, . \sum_{i=1}^{n} [\, y^{(i)} - \sigma(b^T \, x^{(i)}) \,] \, x_j^{(i)} \qquad (3.15)$$

Where α is the step size or the learning rate.

Chapter 4 Implementation and Validation of Smart-Detect System

4.1 Validation of LoRaWAN® for the Smart-Detect System

The extended range that LoRaWAN® presumably have, and the high expectation of communication coverage must be tested beforehand to validate the use of the network for a specific IoT application.

In order to test whether LoRaWAN® network is the best solution for this IoT application prototype or proof-of-concept, a site survey was conducted using a device that helps consumers test their application concept before implementation. This device called the mDot box which is illustrated in the figure 4.1.

Figure 4.1. The mDot box

With a site survey, data were gathered at various power levels and data rates to check the reliability of the proposed technology. The survey was conducted in the lab wherein each room as shown in Figure 4.2 a survey data was gathered to check how far the communication coverage can deliver a good index without degrading. The house simulation in the figure was created to illustrate the distance between each simulated location of a sensor node and the LoRaWAN® gateway router.

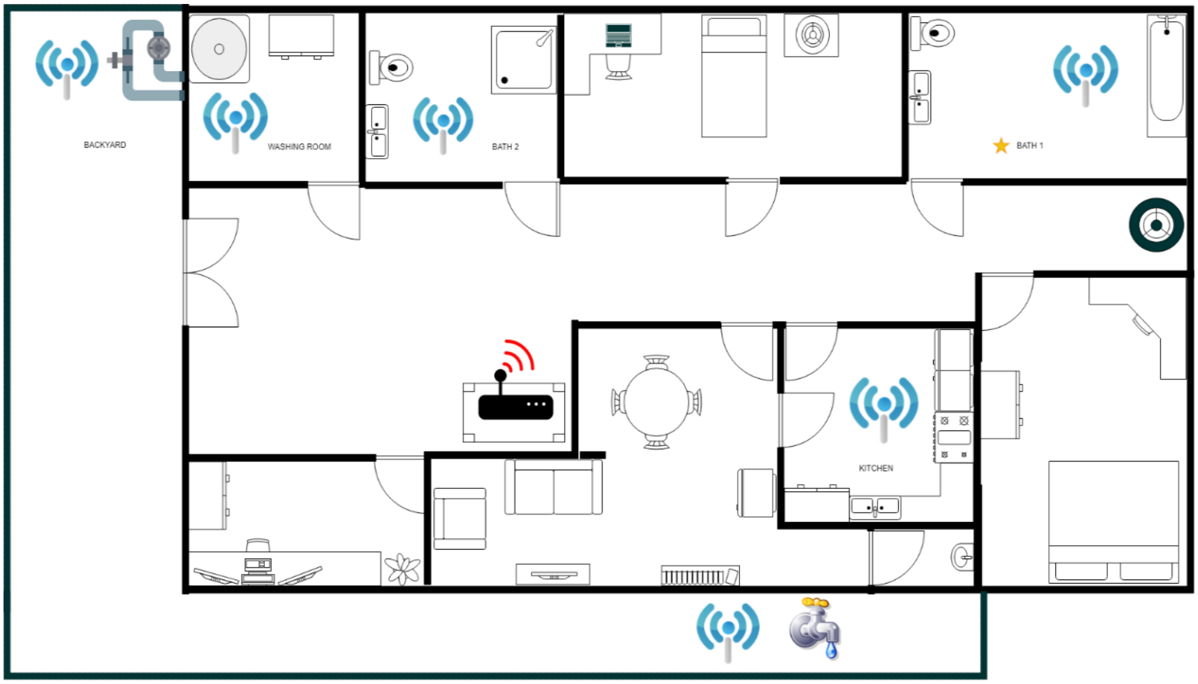

Figure 4.2. The simulation graph of the locations of each sensor node

Data were gathered in the same approximate distance from the router to the sensors nodes as shown in Figure 4.2. All surveys were done for each room that has LoRaWAN® End Device. Data gathered from one such node (Bath 1) was analyzed using MATLAB®. Table 1 shows data gathered from the node in Bath 1 in the figure. The MATLAB® Code that was utilized in this system is included in Appendix A.

The purpose of the survey is to test the reliability of LoRaWAN® network. Figure 4.3 illustrates the results of the survey conducted using different transmission power and data rates to compute the Signal to noise ratio (SNR) which is the difference in decibels between the received signal and the background noise measured in decibels milliwatts (dBm). Whereas Figure 4.4 illustrates the Received signal strength and the margin.

Table 4.1. Survey data gathered from the node in Bath 1

Number of Surveys	Margin	RSSI	SNR in dbm	Data Rate	Power
1	25	-74	7.5	0	2
2	26	-71	7.2	0	8
3	24	-65	7.5	0	14
4	26	-65	7.7	0	20
5	21	-69	7	1	2
6	24	-71	7.7	1	8
7	23	-69	8	1	14
8	24	-75	7.5	1	20
9	21	-66	6.5	2	2
10	20	-65	7.7	2	8
11	21	-71	7	2	14
12	20	-70	8	2	20
13	14	-67	6.7	3	2
14	16	-63	6.5	3	8
15	17	-71	6.5	3	14
16	17	-69	5.7	3	20
17	0	-61	6.5	4	2
18	0	-61	6.7	4	8
19	0	-63	6.7	4	14
20	0	-62	7.2	4	20

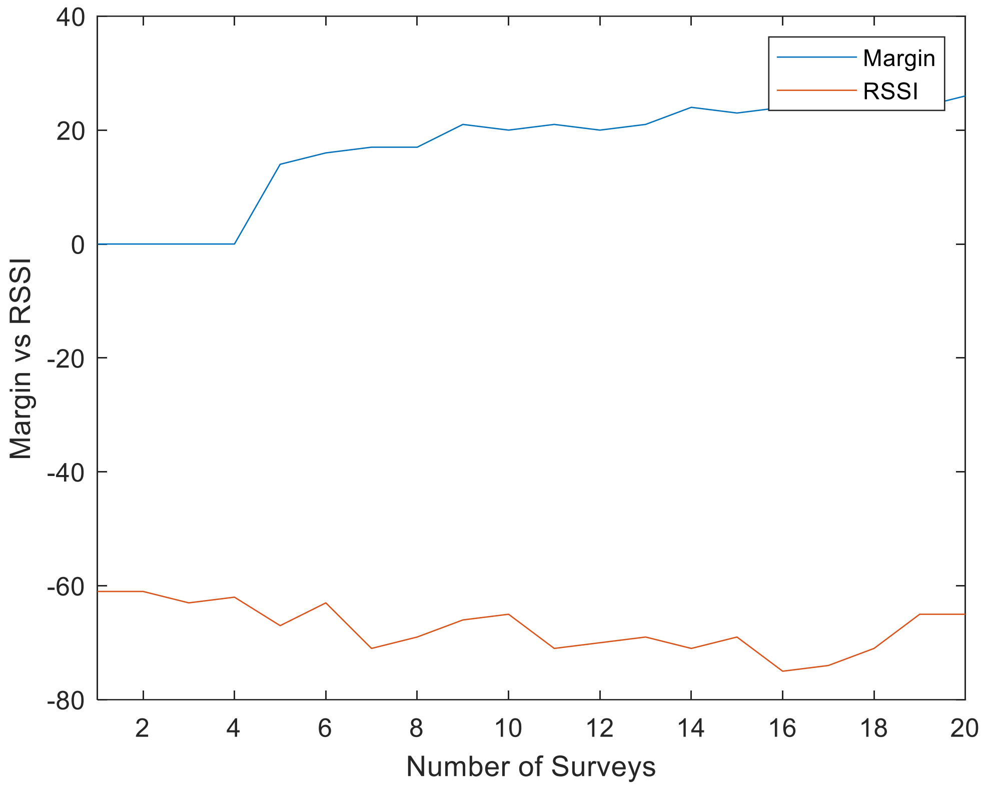

Figure 4.3. Margin vs. Received signal strength index

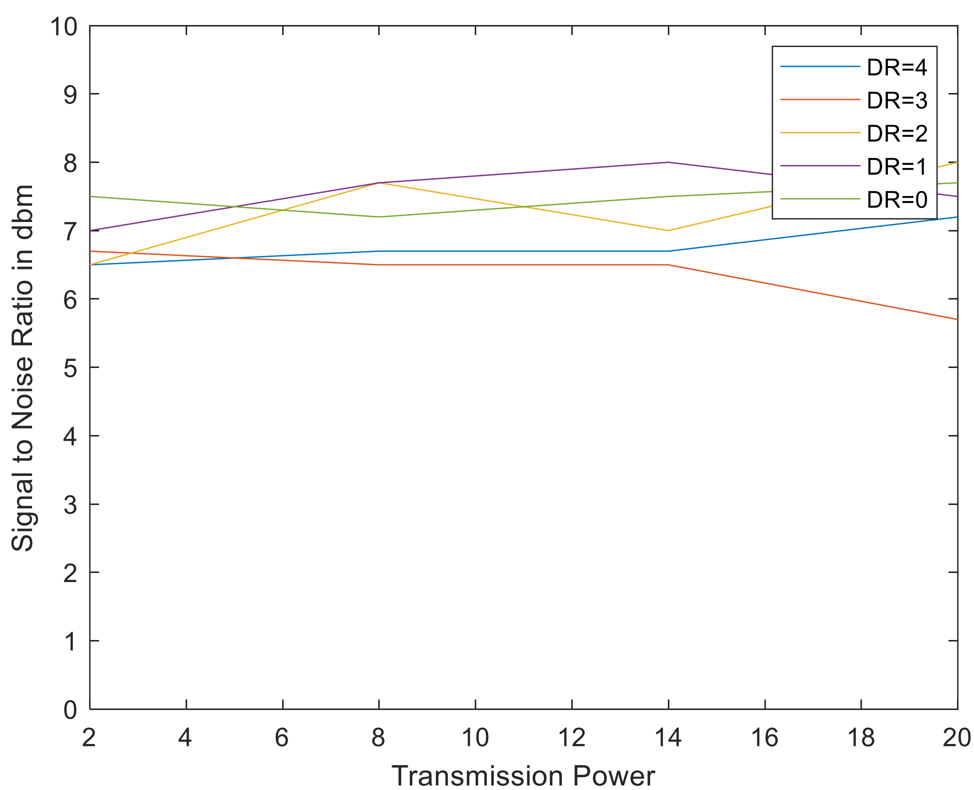

Figure 4.4. Transmission Power vs. SNR

Observing the Received Signal Strength Indicator (RSSI) in Table 1 and comparing it with the standard acceptable signal strength showing below. It shows that RSSI from Table 1 lay in the range of very good signal strength. This indication can give a good insight into the validation of the network to be implemented in the IoT based oil detection system.

Table 4.2. RSSI standard range

-30 dBm	**Amazing**
-67 dBm	Very Good
-70 dBm	Okay
-80 dBm	Not Good
-90 dBm	Unusable

4.2 System Level Simulation (EPANET)

EPANET is a public open source software used for modeling water distribution networks. The goal behind it is to perform a simulation of water behavior and hydraulics within pipe networks. Before building a prototype of the system in the lab. A simulation of the project is required to test and learn if the design will generate the expected output and to validate how accurate the practical flow meter readings would be, by comparing it to the flow rate generated by the simulation model. Standard compatibility between simulation and experiment is always preferred. In this project, a small network was built in EPANET software to build a virtual prototype that potentially can mimic the desired physical experiment. The prototype consists of two ½" Galvanized steel 5 feet pipe each connected in series by a steel fitting. The first End is connected to the sink; the sink is supplied by the water from the water storage tank where the water reservoir fills it. The other end is simulated to fill an empty reservoir for the sake of the experiment.

Before start creating the model, a few coefficients and options must be set. First, the pipe roughness coefficient was changed to use the Hazen-William coefficient (C) which has to be taken in the account as it can affect the readings. The Hazen-William coefficient for Galvanized steel is 120. Also, the Flow rate unit was changed to Liter/Minute (LPM). All shown Below.

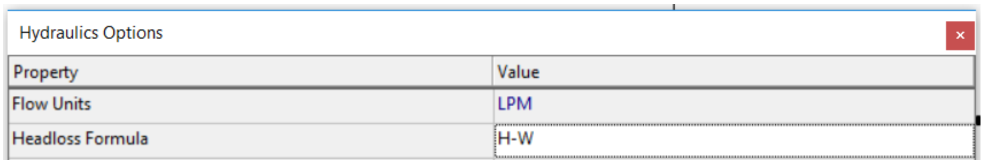

Figure 4.5. EPANET options for different coefficient selections

Furthermore, it is essential to change the system units beforehand to assure the accuracy of the model as shown below.

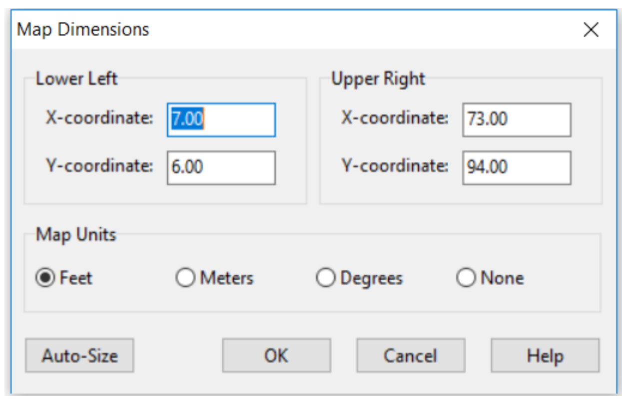

Figure 4.6. EPANET options for different unit selections

In the normal condition where there is no leak, the simulated model is depicted as follows.

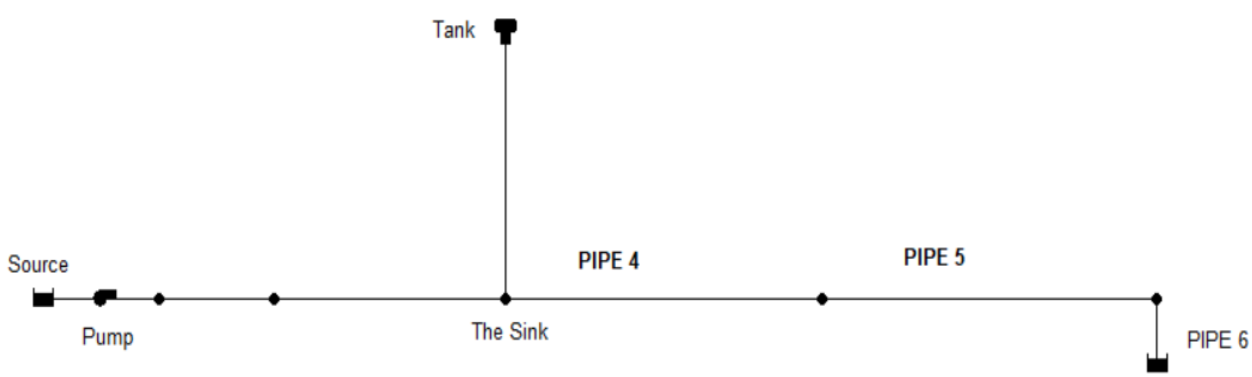

Figure 4.7. EPANET Non-leak Model

The model was created in essence of virtually simulating the amount of water flow the lab sink receives. Thus, only after the illustrated point (The sink) in the previous figure is viable for this experiment. The analysis options must be taken into consideration for Pipe 4 and Pipe 5 such as length of the pipe, pipe roughness, and pipe diameter, as follows:

Pipe 4

Property	Value
*Pipe ID	4
*Start Node	12
*End Node	13
Description	
Tag	
*Length	5
*Diameter	0.5
*Roughness	120
Loss Coeff.	0
Initial Status	Open

Pipe 5

Property	Value
*Pipe ID	5
*Start Node	13
*End Node	1
Description	
Tag	
*Length	5
*Diameter	0.5
*Roughness	120
Loss Coeff.	0
Initial Status	Open

Figure 4.8. The simulation results of the flowrate in pipe 4 and 5

The table of results from the created model illustrated below shows that the flowrate in Pipe 4 and Pipe 5 in normal condition is 6.08 l/min, with length of pipes are 5 feet and diameter is 0.5".

Table 4.3. EPANET normal flow results

Link ID	Length Ft	Diameter In	Roughness	Flow LPM	Status
Pipe 1	40	2	120	87.65	Open
Pipe 2	5	0.5	120	-12.35	Open
Pipe 4	5	0.5	120	6.08	Open
Pipe 3	40	2	120	-121.57	Open
Pipe 5	5	0.5	120	6.08	Open
Pipe 6	2	0.5	120	6.08	Open
Pump 9	#N/A	#N/A	#N/A	87.65	Open

Leakage can be created in EPANET either by adding extra demand of water at a specific node to simulate the leak or by finding the corresponding emitter coefficient in the network and change it to get the desired magnitude of leakage. In the previous model, a leakage was produced after the node between Pipe 4 and Pipe 5 causing the flowrate in Pipe 5 to be reduced. The following table of results illustrates the flowrate for each pipe. The created model shows that the flow rate in Pipe 4 is 6.08 l/min, while Pipe 5 has a flow rate of 5.08 l/min caused by the leak in the node between Pipe 4 and 5.

Table 4.4. EPANET leakage results

Link ID	Length Ft	Diameter In	Roughness	Flow LPM	Status
Pipe 1	40	2	120	88.34	Open
Pipe 2	5	0.5	120	-11.66	Open
Pipe 4	5	0.5	120	6.08	Open
Pipe 3	40	2	120	-110.27	Open
Pipe 5	5	0.5	120	5.08	Open
Pipe 6	2	0.5	120	5.08	Open
Pump 9	#N/A	#N/A	#N/A	88.34	Open

4.3 Data Acquisition and Processing for the proposed Smart-Detect system

In order to test the proposed design, a hardware prototype was built that consist of two ½" Galvanized pipes 5 feet each connected using a ½" steel fitting, both are connected to a water fountain by a hose. Moreover, a valve was connected to the middle fitting between the two pipes, in order to simulate a leak during the experiment. The end of the pipe is connected to a flowmeter, and a temperature sensor both soldered into a microcontroller (mDot) with LoRa® enabled antenna. The following figures show the setup of the prototype.

Figure 4.9. Prototype setup for the proposed IoT-based Smart-Detect framework (1).

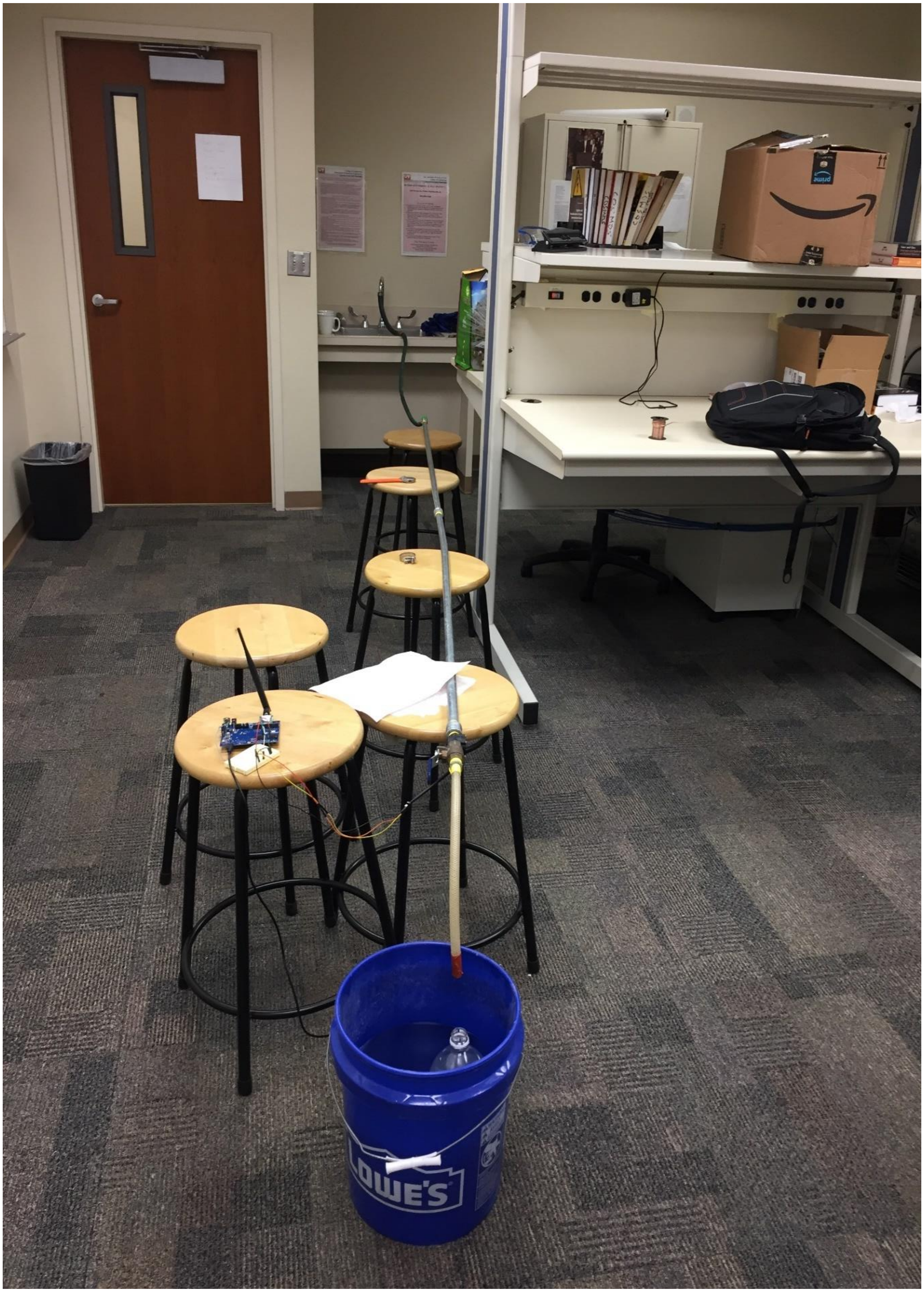

Figure 4.10. Prototype setup for the proposed IoT-based Smart-Detect framework (2).

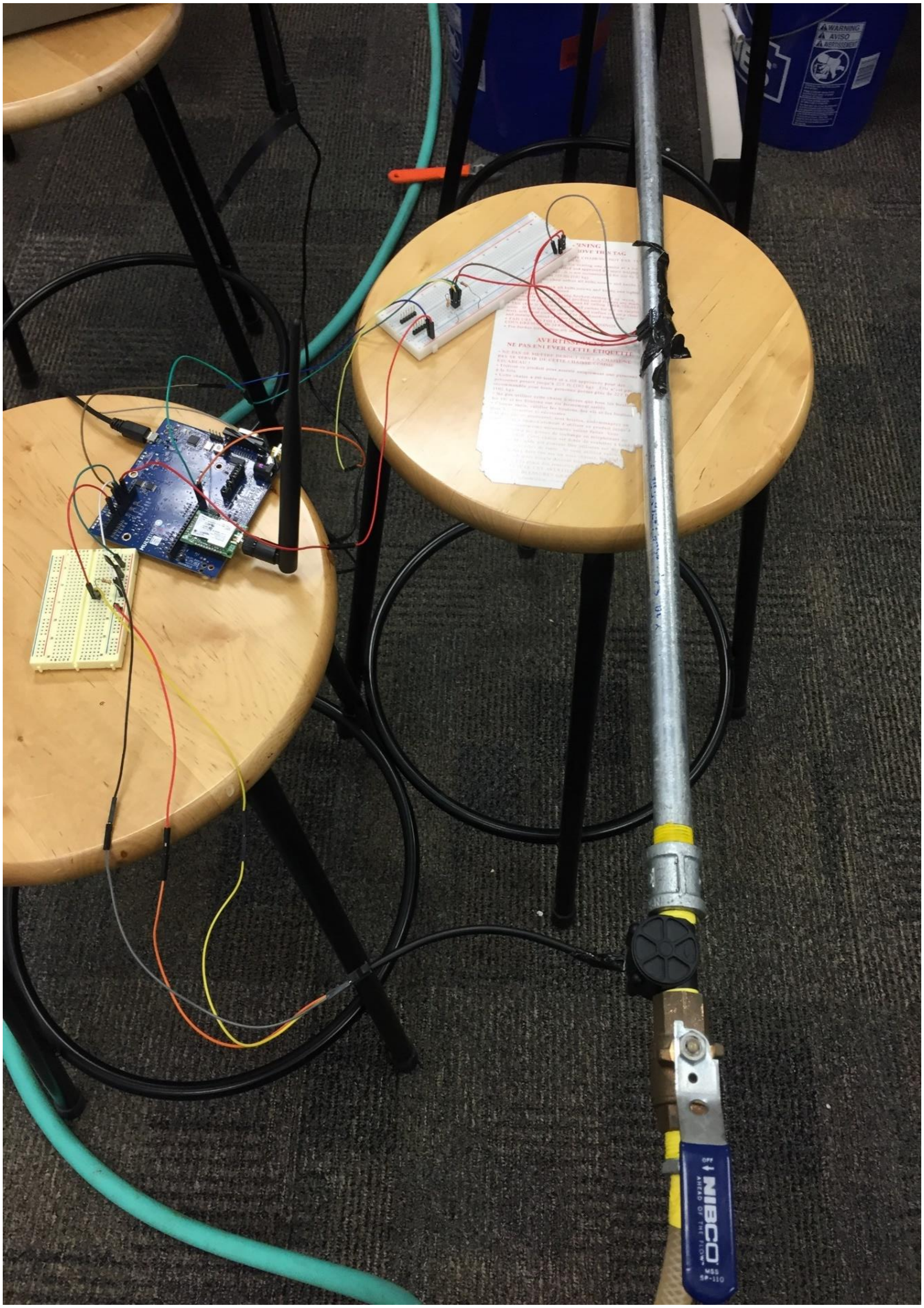

Figure 4.11. Prototype setup for the proposed IoT-based Smart-Detect framework (3)

In this experiment, a set of data was gathered in the normal condition (no leak) like illustrated below, where water is running at full pressure through the pipes, and the data acquisition is taken in this fashion for a couple of hours.

Figure 4.12. Water running at full pressure (middle valve is closed)

As mentioned in Chapter 3 there is a process in which the data is acquired when using LoRaWAN® network. The next section discusses the procedure of how the data was acquired from the prototype.

Figure 4.13. The LoRaWAN® Gateway and the microcontroller

The previous figure depicts the used microcontroller which is called mDot (MTDOT-915-X1P-SMA-1) together with the Multiconnect® Conduit® gateway. The mDot is mounted on a Multiconnect mDot Developer Kit (MDK), this is needed for programming the mDot, where Mbed OS is utilized to write and debug the program to acquire data and send it through the LoRaWAN® network, next the Conduit® gateway forward the data to the network server (TTN). Before uploading the program to the microcontroller, the connection should be established with the sensors (Flowmeter and BME280), the pins should be connected to the right sockets and the use of pull-up resistors (10Kohms) between the input voltage and the pin required to trigger a specific signal is essential. The next figure illustrates the established connections.

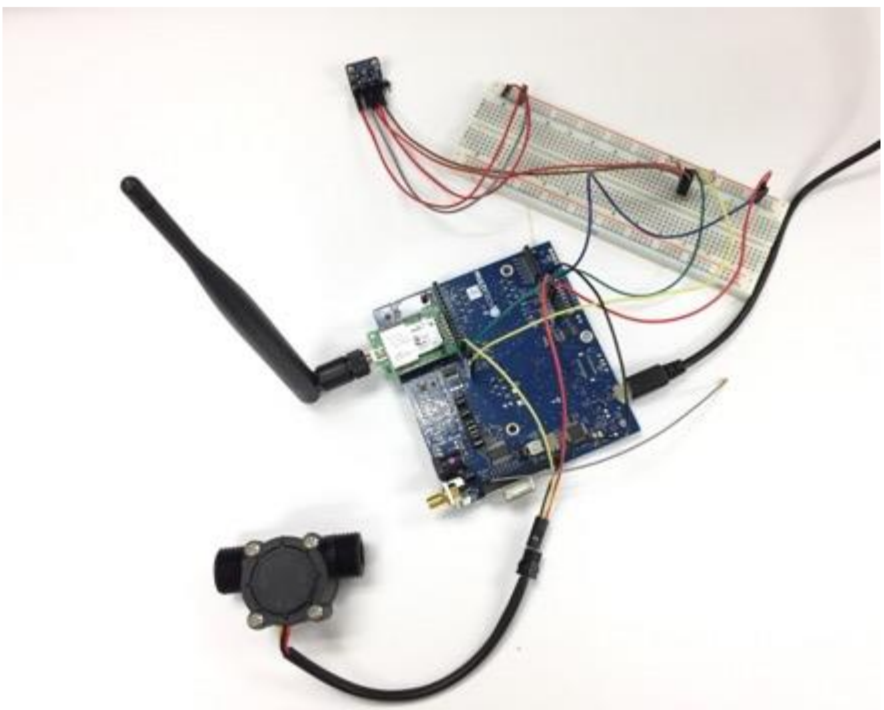

Figure 4.14. Sensor node with attached flowmeter and temperature sensor

The Mbed OS is used here to include the libraries of the two sensors plus including the LoRa® connection driver, where keys are inputted in the LoRa® Library of the Mbed OS. These keys are provided from the network server (TTN) upon creating an application on the platform using OTAA as mentioned in section 3.1 in the previous chapter. The following figures show the Mbed OS and TTN platforms. Also, part of the program that was utilized in this system, is included in Appendix B.

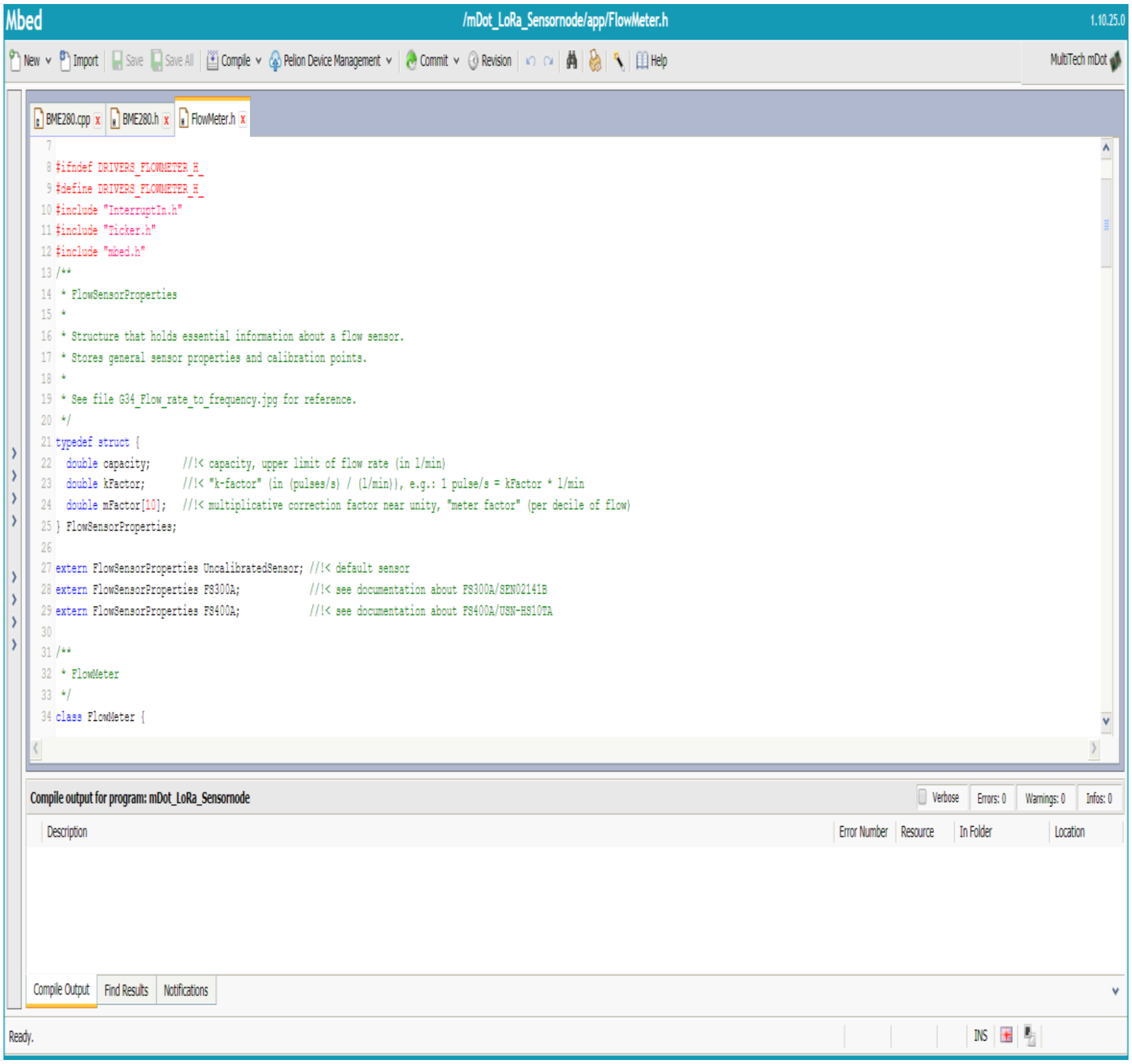

Figure 4.15. The Mbed OS Platform

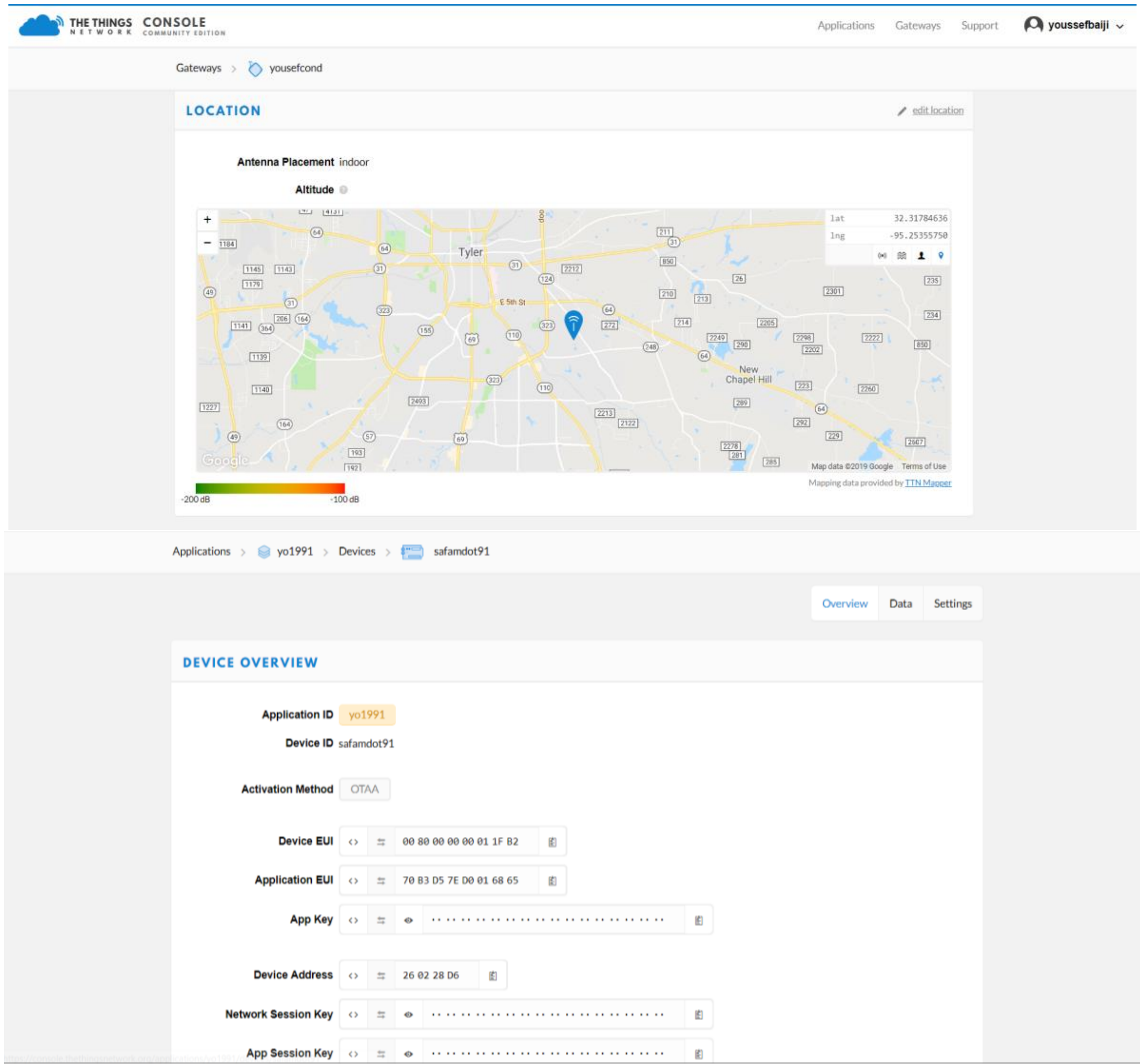

Figure 4.16. The Things Network platform

The TTN platform shows the specific location of the gateway on the map in addition to the keys required to connect to this specific device. After a successful configuration, the network server was receiving encoded payloads from the microcontroller as illustrated in the figure below. The raw data from the sensors are received in base64 hex-format almost 17 seconds apart. This raw data will be converted in Node-Red and will be forwarded to the Thingspeak platform as mentioned in chapter 3.

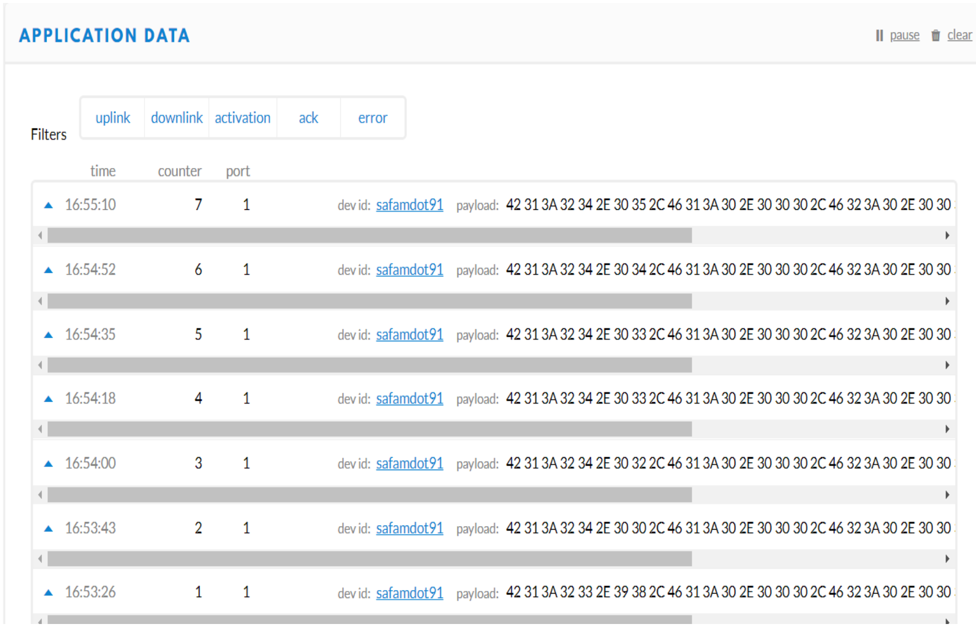

Figure 4.17. The received encoded payloads in TTN

This created flow help preprocess the data to be received in real time without errors to the Thingspeak platform, also provides a simple way to check the data by using a debugger as shown below.

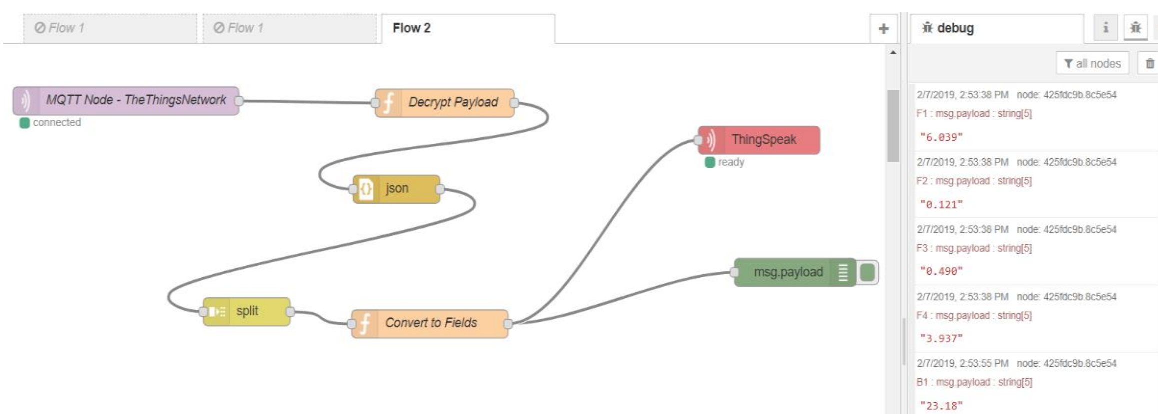

Figure 4.18. Node-Red flow for forwarding data to Thingspeak

In Thingspeak a channel was created to gather the data for the non-leak condition, in which data was received and plotted in real time as shown below.

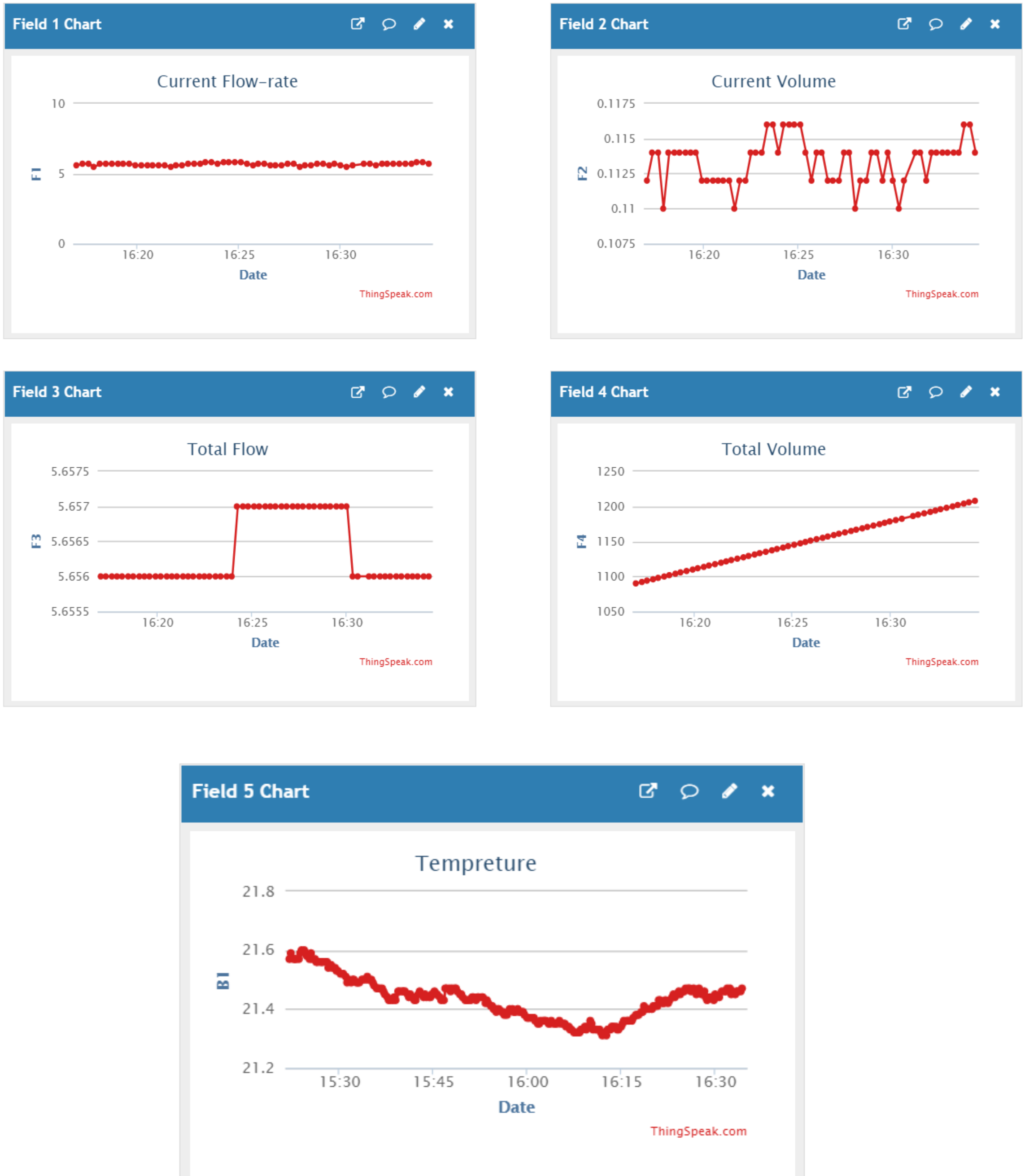

Figure 4.19. The received data from the sensor node in real time

Totally around 600 data points were gathered when the water was running at full pressure for a couple of hours. The data were examined and crosschecked with the simulated data from EPANET in section 4.1 to make sure that the sensor is not producing wrong readings. The following table illustrates part of the gathered data.

Table 4.5. Non-Leak Data generated from the Smart-Detect system

Time Stamp	Current Flow-Rate	Current Volume	Total Flow	Total Volume	Temperature
'13-Dec-2018 13:37:03'	5.816	0.116	5.925	4.266	23.18
'13-Dec-2018 13:37:20'	5.928	0.119	5.921	6.039	23.22
'13-Dec-2018 13:37:37'	5.928	0.119	5.921	7.816	23.25
'13-Dec-2018 13:37:55'	5.928	0.119	5.921	9.592	23.31
'13-Dec-2018 13:38:12'	5.928	0.119	5.922	11.37	23.36
'13-Dec-2018 13:38:30'	5.928	0.119	5.923	14.926	23.42
'13-Dec-2018 13:38:47'	6.039	0.121	5.924	16.826	23.47
'13-Dec-2018 13:39:04'	5.816	0.116	5.915	18.928	23.51
'13-Dec-2018 13:39:21'	5.816	0.116	5.91	20.921	23.58
'13-Dec-2018 13:39:39'	5.928	0.119	5.907	22.919	23.58
'13-Dec-2018 13:39:56'	5.592	0.112	5.898	25.006	23.57
'13-Dec-2018 13:40:14'	5.592	0.112	5.879	26.927	23.56
'13-Dec-2018 13:40:31'	5.704	0.114	5.865	28.972	23.52
'13-Dec-2018 13:40:48'	5.816	0.116	5.859	30.938	23.49
'13-Dec-2018 13:41:06'	5.816	0.116	5.857	32.917	23.44
'13-Dec-2018 13:41:23'	5.816	0.116	5.855	35.016	23.38
'13-Dec-2018 13:41:40'	5.816	0.116	5.854	37	23.34
'13-Dec-2018 13:41:58'	5.816	0.116	5.854	38.986	23.29
'13-Dec-2018 13:42:15'	5.928	0.119	5.854	41.095	23.23
'13-Dec-2018 13:42:32'	5.928	0.119	5.854	43.086	23.19
'13-Dec-2018 13:42:50'	5.816	0.116	5.848	45.032	23.16
'13-Dec-2018 13:43:07'	5.48	0.11	5.831	46.996	23.11
'13-Dec-2018 13:43:24'	5.704	0.114	5.823	48.911	23.09
'13-Dec-2018 13:43:42'	5.816	0.116	5.82	50.87	23.06
'13-Dec-2018 13:43:59'	5.816	0.116	5.819	52.957	23.03
'13-Dec-2018 13:44:16'	5.704	0.114	5.818	54.926	23.03
'13-Dec-2018 13:44:34'	5.816	0.116	5.818	57.015	23.03
'13-Dec-2018 13:44:51'	5.816	0.116	5.817	58.986	23.02
'13-Dec-2018 13:45:09'	5.816	0.116	5.816	60.952	23
'13-Dec-2018 13:45:26'	5.704	0.114	5.815	63.034	22.98
'13-Dec-2018 13:45:43'	5.592	0.112	5.81	64.953	22.97
'13-Dec-2018 13:46:01'	5.704	0.114	5.804	66.861	22.95
'13-Dec-2018 13:46:18'	5.704	0.114	5.801	68.913	22.94
'13-Dec-2018 13:46:35'	5.816	0.116	5.799	70.868	22.93
'13-Dec-2018 13:46:53'	5.704	0.114	5.798	72.823	22.93
'13-Dec-2018 13:47:10'	5.592	0.112	5.789	74.798	22.93
'13-Dec-2018 13:47:28'	5.704	0.114	5.786	76.726	22.93
'13-Dec-2018 13:47:45'	5.704	0.114	5.784	78.663	22.93
'13-Dec-2018 13:48:02'	5.816	0.116	5.782	80.716	22.92

For convenience, another channel was created in Thingspeak for leakage data, in which the smallest leak was created as shown below, and another 600 data points were gathered where a portion of them are shown in Table 4.2.

Figure 4.20. The leak created by marginally opening the middle valve

These set of data were combined in MATLAB® to be used to train the Logistic Regression Model. In supervised learning, the algorithm teaches itself to learn from the labeled data. Thus, in this case, the training data should be labeled as one in case of a non-leak and zero in case of a leak. So, after obtaining a labeled dataset, the logistic regression model can be applied to train the data in order to predict the label for the new upcoming unlabeled data. As mentioned in Chapter 3 beta coefficient have to be evaluated in logistic regression, but before training the model feature selection is a crucial step for the effective algorithm. an analyst must decide how many features and how integral the feature can contribute to the output. For this experiment, two features have been selected from the data like Current Flowrate and Current Volume. Whereas Temperature data are used only for monitoring the water temperature and could not be used as a feature. Due to the constant variant nature stemming from the change in the temperature of the pipes cause by weather variation. As a result, the logistic regression equation will be represented as following:

$$logit(P) = b_0 + b_1 * \ Current\ Flowrate + b_2 * \ Current\ Volume \tag{4.1}$$

The program that was utilized in this system, is included in Appendix C.

Table 4.6. Leak Data generated from the Smart-Detect system

Time Stamp	Current Flow-Rate	Current Volume	Total Flow	Total Volume	Temperature
'14-Dec-2018 15:38:59'	5.704	0.114	5.805	2.438	30.02
'14-Dec-2018 15:39:16'	5.704	0.114	5.791	4.169	29.75
'14-Dec-2018 15:39:34'	5.704	0.114	5.768	5.883	29.14
'14-Dec-2018 15:39:51'	5.704	0.114	5.748	7.587	29.05
'14-Dec-2018 15:40:08'	5.816	0.116	5.748	9.312	29.14
'14-Dec-2018 15:40:26'	5.704	0.114	5.746	11.032	29.06
'14-Dec-2018 15:40:43'	5.704	0.114	5.752	14.495	29.23
'14-Dec-2018 15:41:01'	5.592	0.112	5.75	16.329	28.93
'14-Dec-2018 15:41:18'	5.592	0.112	5.731	18.338	28.76
'14-Dec-2018 15:41:35'	5.704	0.114	5.723	20.259	28.59
'14-Dec-2018 15:41:52'	5.704	0.114	5.722	22.201	28.36
'14-Dec-2018 15:42:10'	5.704	0.114	5.722	24.263	28.95
'14-Dec-2018 15:42:27'	5.704	0.114	5.72	26.198	28.36
'14-Dec-2018 15:42:45'	5.704	0.114	5.716	28.238	27.59
'14-Dec-2018 15:43:02'	5.704	0.114	5.715	30.175	27.59
'14-Dec-2018 15:43:19'	5.592	0.112	5.714	32.114	27.63
'14-Dec-2018 15:43:37'	5.704	0.114	5.714	34.168	27.6
'14-Dec-2018 15:43:54'	5.704	0.114	5.712	36.103	27.57
'14-Dec-2018 15:44:11'	5.704	0.114	5.711	38.033	27.53
'14-Dec-2018 15:44:29'	5.704	0.114	5.709	40.08	27.17
'14-Dec-2018 15:44:46'	5.704	0.114	5.708	42.01	26.93
'14-Dec-2018 15:45:03'	5.704	0.114	5.706	43.938	26.45
'14-Dec-2018 15:45:21'	5.704	0.114	5.705	45.985	26.46
'14-Dec-2018 15:45:38'	5.704	0.114	5.706	47.927	26.71
'14-Dec-2018 15:45:56'	5.704	0.114	5.707	49.877	26.85
'14-Dec-2018 15:46:13'	5.816	0.116	5.709	51.948	26.98
'14-Dec-2018 15:46:30'	5.704	0.114	5.711	53.908	27.09
'14-Dec-2018 15:46:47'	5.816	0.116	5.713	55.988	27.18
'14-Dec-2018 15:47:05'	5.816	0.116	5.715	57.954	27.24
'14-Dec-2018 15:47:22'	5.704	0.114	5.718	59.923	27.28
'14-Dec-2018 15:47:40'	5.48	0.11	5.717	61.972	27.34
'14-Dec-2018 15:47:57'	5.704	0.114	5.713	63.871	27.45
'14-Dec-2018 15:48:14'	5.704	0.114	5.711	65.792	27.6
'14-Dec-2018 15:48:32'	5.704	0.114	5.711	67.848	27.72
'14-Dec-2018 15:48:49'	5.592	0.112	5.71	69.778	28.55
'14-Dec-2018 15:49:06'	5.592	0.112	5.708	71.686	27.83
'14-Dec-2018 15:49:24'	5.592	0.112	5.705	73.711	27.27
'14-Dec-2018 15:49:41'	5.704	0.114	5.704	75.637	27.46
'14-Dec-2018 15:49:59'	5.592	0.112	5.703	77.56	27.24

4.4 Experimental Results

By using the machine learning toolbox in MATLAB, the model was built, and data was read directly from the created Thingspeak Channels. Running the code section regarding the training process, the results showing below illustrates the computed Beta coefficients from the training process.

mdl = Generalized linear regression model: logit(y) ~ 1 + x1 + x2
Distribution = Binomial

Estimated Coefficients:

	Estimate	*SE*	*tStat*	*pValue*
(Intercept)	*-11.496*	*4.2463*	*-2.7073*	*0.0067832*
x1	*-771.94*	*388.06*	*-1.9892*	*0.046675*
x2	*17.46*	*7.2602*	*2.4049*	*0.016179*

1200 observations, 1197 error degrees of freedom
Dispersion: 1
Chi^2-statistic vs. constant model: 37.6, p-value = 6.96e-09

Results show that:

$$b_0 = -11.496$$
$$b_1 = -771.94$$
$$b_2 = 17.46$$

Regression coefficients are the average change of the response variable in association with the predictor variable. In the above results, it can be observed that the negative coefficient implies that the odds when the class = 0 are lesser than the odds in case that variable = 1. This is sensible since the introduced leak is very small ,hence most values of flow rate will be approximately close except for a few spikes associated with the drop in the flowrate which indicates the occurrence of the leak. The combined training data that was used in this model was visualized using Tableau by plotting x1 vs x2 to illustrate the proximity of the values between leak and no leak. Figure 4. 3 depicts the scatter plot of the data.

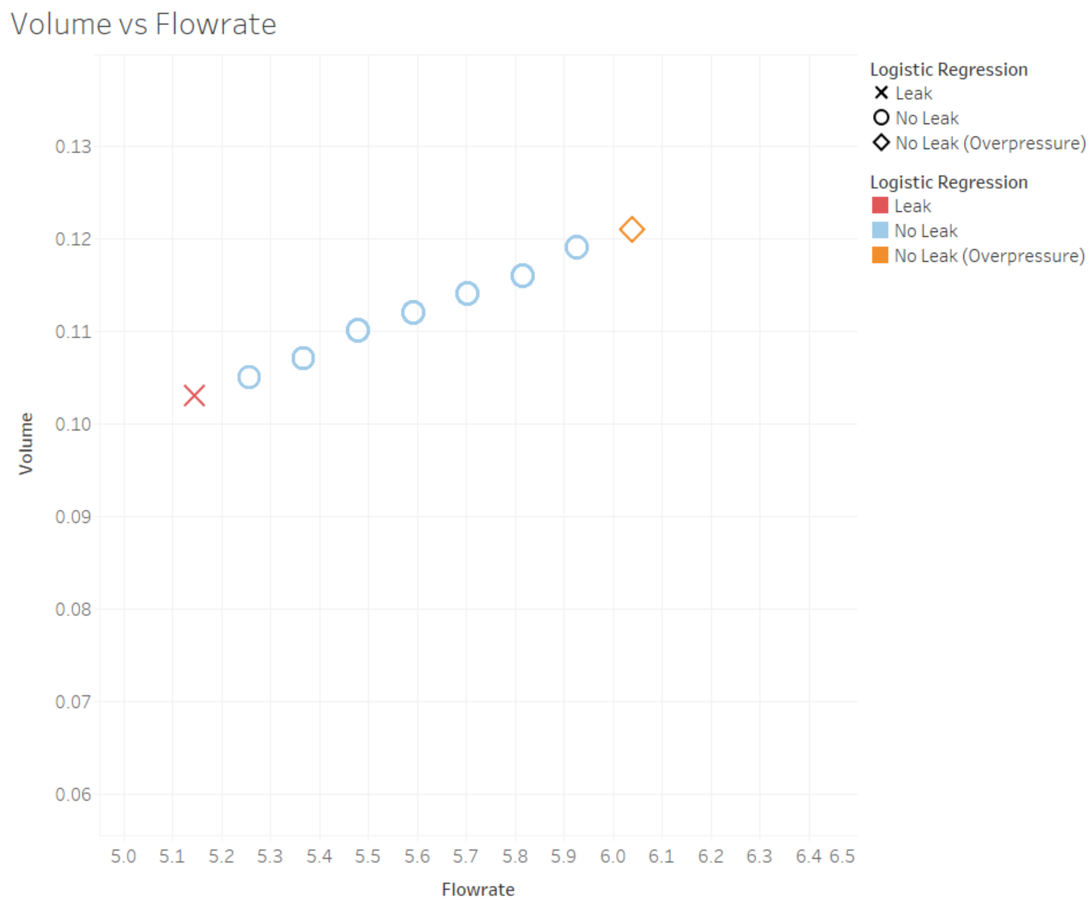

Figure 4.21. Visualization of data in Tableau (Flowrate vs Volume)

Furthermore, the p-value for the overall model should be less than the common value 0.05 this would indicate that at least one of the independent variables contributes to the prediction of the outcome. A low p-value indicates that there is a meaningful addition to the model. In contrast a large p-value indicate that changed in the response are not associated with the predictor variable. In the results it can be noted that the predictor variables of x1 and x2 which are (current flowrate and current volume) are significant because they both have p-values less than 0.05. However, the p-value of the intercept is greater than the common value of 0.05 which indicate that is not statistically significant. Whereas the overall p-value is less than 0.05 (6.96e-09) which indicates good results. The next result is known as the goodness of fit or Chi-Squared statistic which is a measure of how well the independent variables affect the outcome. In this case the result was 37.6.

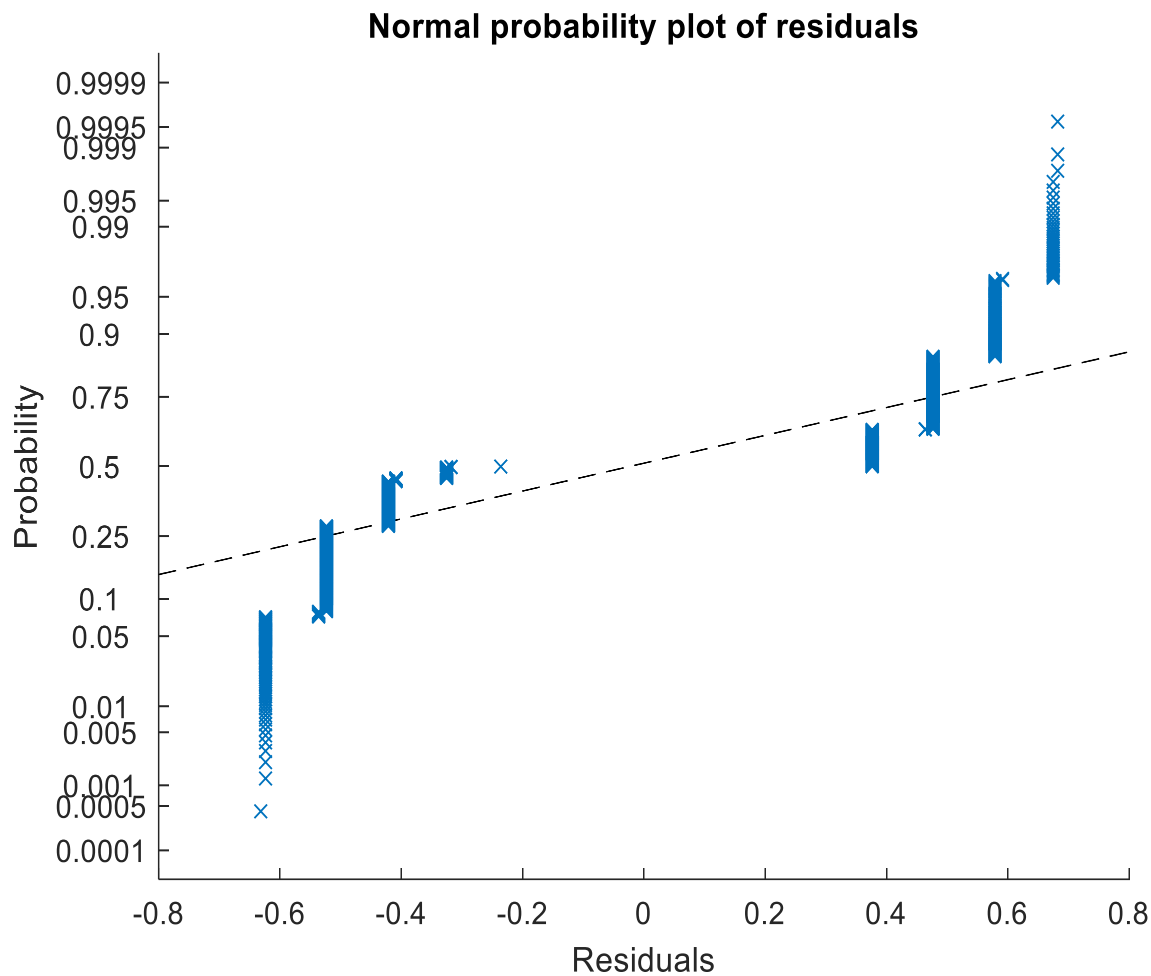

Figure 4.22. The decision boundary of the created Logistic Regression model and the Residuals

The figure illustrates the hyperplane which separates the two classes, when the probability is above 0.5 according to the decision boundary for instance $P \geq 0.5$ it classifies the class as 1 (non-leak) and when $P < 0.5$ it classifies the class as 0 (leak). Also, it shows the residuals which is the difference between the observed label (y) and the label (ypred) predicted by the model.

After Training the model is time to test whether the model can predict or produce the right outcome. The constructed MATLAB® code should be copied to the Thingspeak to test it in real time. As mentioned before a third channel was created in Thingspeak for testing. The upcoming figures shows the results in real time of the system predicting the outcome in three different cases.

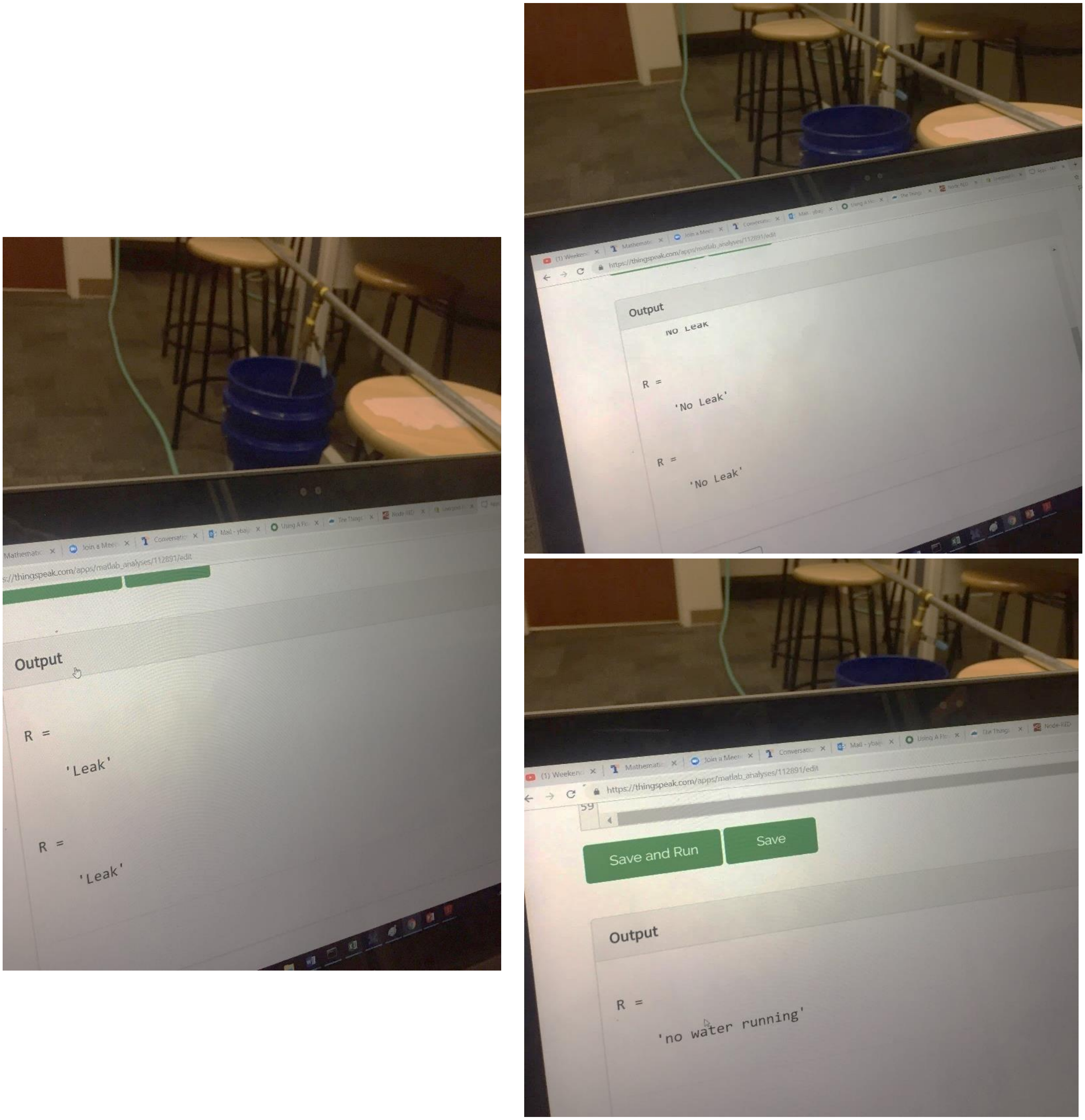

Figure 4.23. The predicted real time results in three cases (no water, no leak, leak)

Chapter 5 Discussion and Conclusion

5.1 Discussion

The best way to predict the effectiveness of the proposed design is to conduct sufficient amount of trials and observe the system reliability, by creating different scenarios in which different sizes of leaks are introduces and checking whether the system produce the right outcome each time. By calculating the misclassification error which is how many times the system predicts the wrong output in comparison to predicting the right output, this procedure can indicate the overall efficiency of the system.

First Trial:

Table 5.1. First prediction trial for testing the Smart-Detect system

Entry_ID	Current Flow-Rate	Current Volume	Projected Prediction	Middle Valve	Actual Prediction
1	0	0	No water running	Closed	'no water running'
2	0	0	No water running	Closed	'no water running'
3	0	0	No water running	Closed	'no water running'
4	0	0	No water running	Closed	'no water running'
5	0	0	No water running	Closed	'no water running'
6	0	0	No water running	Closed	'no water running'
7	0	0	No water running	Closed	'no water running'
8	6.039	0.121	No leak	Closed	'No leak'
9	6.039	0.121	No leak	Closed	'No leak'
10	6.039	0.121	No leak	Closed	'No leak'
11	6.039	0.121	No leak	Closed	'No leak'
12	5.928	0.119	No leak	Closed	'No leak'
13	5.928	0.119	No leak	Closed	'No leak'
14	5.928	0.119	No leak	Closed	'No leak'
15	5.928	0.119	No leak	Closed	'No leak'
16	5.928	0.119	No leak	Closed	'No leak'
17	5.928	0.119	No leak	Closed	'No leak'

18	5.592	0.112	Leak	Open 5%	'Leak'
19	5.592	0.112	Leak	Open 5%	'Leak'
20	4.697	0.094	Leak	Open 10%	'Leak'
21	4.362	0.087	Leak	Open 10%	'Leak'
22	4.474	0.089	Leak	Open 15%	'Leak'
23	3.914	0.078	Leak	Open 25%	'Leak'
24	3.914	0.078	Leak	Open 25%	'Leak'
25	3.914	0.078	Leak	Open 25%	'Leak'

Second Trial:

Table 5.2. Second prediction trial for testing the Smart-Detect system

Entry_ID	Current Flow-Rate	Current Volume	Projected Prediction	Middle Valve	Actual Prediction
1	0	0	No water running	Closed	'no water running'
2	0	0	No water running	Closed	'no water running'
3	0	0	No water running	Closed	'no water running'
4	0	0	No water running	Closed	'no water running'
5	0	0	No water running	Closed	'no water running'
6	0	0	No water running	Closed	'no water running'
7	5.928	0.119	No leak	Closed	'No leak'
8	6.039	0.121	No leak	Closed	'No leak'
9	5.928	0.119	No leak	Closed	'No leak'
10	5.928	0.119	No leak	Closed	'No leak'
11	5.928	0.119	No leak	Closed	'No leak'
12	5.928	0.119	No leak	Closed	'No leak'
13	5.816	0.116	No leak	Closed	'No leak'
14	5.816	0.116	No leak	Closed	'No leak'
15	5.816	0.116	No leak	Closed	'No leak'
16	5.816	0.116	No leak	Closed	'No leak'
17	5.592	0.112	Leak	Open 5%	'Leak'
18	5.592	0.112	Leak	Open 5%	'Leak'
19	5.816	0.116	Leak	Open 10%	'No leak'
20	5.145	0.103	Leak	Open 10%	'Leak'
21	5.033	0.101	Leak	Open 15%	'Leak'
22	4.586	0.092	Leak	Open 20%	'Leak'
23	4.586	0.092	Leak	Open 20%	'Leak'
24	4.474	0.089	Leak	Open 25%	'Leak'
25	4.474	0.089	Leak	Open 25%	'Leak'

Two trials have been conducted, at first the water wasn't running and in this case the algorithm correctly indicate that there is no water running in the pipe, and after the water is opened to the full pressure the algorithm correctly predict that there is no leak, finally after a leak is introduced the valve is opened marginally and in each case the algorithm correctly predict a leak in most trials. From the above trials it can be noted that the system has an overall efficiency of 98% since there is only one misclassification.

5.2 Conclusions

Most commonly used techniques for leak detection depend on simulation which can produce great results. However, the variation of the medium pressure and the influence of the outer environment can severely affect these sorts of applications. These nonlinearities can't be produced in a computer simulation, due to these factors a practical approach is needed for better control and precision. Proving that this technology can be used to practically implement leakage detection system with a good efficiency can open a new window towards research and innovation in this direction, because it is not just about getting the right outcome this system overcomes several drawbacks that faced the researchers before, drawbacks like communication range and coverage, cost, reaction speed, real rime monitoring, and the battery life. The later drawback was mostly the core problem because this application has to be battery powered, and a low power communication protocol that keeps the battery from draining up in a short period of time is needed. LoRaWAN® solve most of these problems as shown in this conducted experiment.

In this section a further explanation of how an IoT based system like the used system in this research can overcome the mentioned drawbacks. Firstly, in section 4.1 a wide explanation and an experiment was introduced to show the coverage capability and the performance of added range of LoRaWAN® technology, an outdoor range that exceeds 5 miles and up to 10 miles. While in deep in-building penetration it has a range of 1 to 3 miles.

The cost of implementing this system is relatively cheap as compared to other utilized systems in the oil field sector. The utilization of LoRaWAN® that operates in the 915 MHZ part of the spectrum merely cost nothing at least from regulator point of view, there is no license fees at all although the technology itself requires a certain cost, but that cost is not very great when spread across many devices.

Moreover, as observed from the results the reaction time can be very fast mostly 20 seconds and can be reduced if a direct integration with a well-known cloud was implemented, clouds like IBM Cloud, Amazon Web services, or Microsoft Azure and other. Usage of Virtual Server Instances of these clouds can add more cost to the system. However, the system results can be considered real time as the outcome was predicted while the system is acquiring data in real time. The real time detection and fast reaction time are essential in constructing a feasible leak detection system.

The battery life is important for leak detection system. Battery life depends on many variables, including data rate, transmit power and duty cycle. A low power communication protocol like LoRaWAN® help retain the battery life and reduce the consumption of power due to low data rate. Furthermore, developers introduced further enhancements in the LoRa® enabled microcontroller firmware which is called auto sleep feature, a power optimization scheme that allows the microcontroller to automatically stop and go to sleep mode after an uplink transmission ends and in between two receive windows, this additionally increase the battery life which makes it operate for several years.

Finally, the versatility and ease of use of this technology offers an extra capability that might save cost and effort of constructing a machine learning model for leak localization. Because each sensor node can be registered in the network server, the location of each sensor node can be known. For more precision a GPS can easily be integrated with the sensor node where the location of a leak can be indicated.

5.3 Future work

Many different adaptations, tests, and experiments can further improve the system for instance further improvement in the test setup can be made by introducing different leaks in the pipe in various locations and with different sizes. In addition, adding more sensors like pressure sensor and trying different pipe diameters with different materials, can increase the number of features used in gathering the data used to train the model, additionally training the model with different water pressure. all mentioned approaches can produce a more powerful algorithm for leak detection. Moreover, an extension of the proposed system is to build a mobile application or a web service that can alarm the user in case a leak occurs, which is an important aspect in building an IoT based system. Another approach can be suggested is using statistical approaches to generate more data by using Monte-Carlo simulation.

References

[1] W. Rahiman and Z. Ding, "Leak detection in pipeline via sliding motion," in *2017 17th International Conference on Control, Automation and Systems (ICCAS)*, 2017.

[2] M. Ferrante and B. Brunone, "Pipe system diagnosis and leak detection by unsteady-state tests. 1. Harmonic analysis," *Advances in Water resources,* vol. 26, pp. 95-105, 2003.

[3] W. Mpesha, M. H. Chaudhry and S. L. Gassman, "Leak detection in pipes by frequency response method using a step excitation," *Journal of Hydraulic Research,* vol. 40, pp. 55-62, 2002.

[4] M. Bentoumi, D. Chikouche, A. Mezache and H. Bakhti, "Wavelet DT method for water leak-detection using a vibration sensor: an experimental analysis," *IET Signal Processing,* vol. 11, pp. 396-405, 2017.

[5] L. Dong, S. Chai and B. Zhang, "Leak detection and localization of gas pipeline system based on wavelet analysis," in *2011 2nd International Conference on Intelligent Control and Information Processing*, 2011.

[6] J. Du, L. Wang, C. Cai, C. Yin and G. Zhao, "Study on distributed optical fiber heating pipeline network leak detection system," in *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2017.

[7] A. Abdulshaheed, F. Mustapha and A. Ghavamian, "A pressure-based method for monitoring leaks in a pipe distribution system: A Review," *Renewable and Sustainable Energy Reviews,* vol. 69, pp. 902-911, 2017.

[8] X. Chen, D. Zhang, Y. Wang, L. Wang, A. Zomaya and S. Hu, "Offshore oil spill monitoring and detection: Improving risk management for offshore petroleum cyber-physical systems: (Invited paper)," in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD),* 2017.

[9] P. Trivero, W. Biamino and F. Nirchio, "High resolution COSMO - SkyMed SAR images for oil spills automatic detection," in *2007 IEEE International Geoscience and Remote Sensing Symposium*, 2007.

[10] H. Denkilkian, A. Koulakezian, R. Ohannessian, M. S. Chalfoun, M. K. W. Joujou, A. Chehab and I. H. Elhajj, "Wireless Sensor for Continuous Real-Time Oil Spill

Thickness and Location Measurement," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, pp. 4001-4011, 12 2009.

[11] Y. Li, L. Wang, L. Chen, Y. Ma, X. Zhu and B. Chu, "Application of DDDAS in marine oil spill management: A new framework combining multiple source remote sensing monitoring and simulation as a symbiotic feedback control system," in *2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS*, 2013.

[12] Z. Qingling and L. Ying, "Monitoring Marine Oil-spill Using Microwave Remote Sensing Technology," in *2007 8th International Conference on Electronic Measurement and Instruments*, 2007.

[13] O. P. N. Calla, H. K. Dadhich and S. Singhal, "Oil spill detection using multi frequency microwave sensor onboard satellite; SSM/I and AMSR-E," in *2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS*, 2013.

[14] S. Liu, M. Chi, Y. Zou and A. Samat, "A multitemporal change detection solution to oil spill monitoring," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016.

[15] M. N. Jha and Y. Gao, "Oil spill contingency planning using Laser Fluorosensors and web-based GIS," in *OCEANS 2008*, 2008.

[16] Y. Li, W. Shih, Z. Han and W. Yin, "Oil spill sensor using multispectral infrared imaging via ℓ1minimization," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[17] E. O'Connell, S. O'Keeffe, T. Newe and E. Lewis, "Low cost hydrocarbon spillage sensor for the marine environment with interfacing to a mote platform," in *SENSORS, 2011 IEEE*, 2011.

[18] S. Savazzi and U. Spagnolin, "Synchronous ultra-wide band wireless sensors networks for oil and gas exploration," in *2009 IEEE Symposium on Computers and Communications*, 2009.

[19] M. Dalbro, E. Eikeland, A. J. Veld, S. Gjessing, T. S. Lande, H. K. Riis and O. Søråsen, "Wireless Sensor Networks for Off-shore Oil and Gas Installations," in *2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008)*, 2008.

[20] M. Akhondi, A. Talevski, S. Carlsen and S. Petersen, "Applications of Wireless Sensor Networks in the Oil, Gas and Resources Industries," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, 2010.

[21] L. Zheng, "ZigBee Wireless Sensor Network in Industrial Applications," in *2006 SICE-ICASE International Joint Conference*, 2006.

[22] Y. Cheng, H. Saputra, L. M. Goh and Y. Wu, "Secure smart metering based on LoRa technology," in *2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, 2018.

[23] C. Choi, J. Jeong, I. Lee and W. Park, "LoRa based renewable energy monitoring system with open IoT platform," in *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, 2018.

[24] A. Ouya, B. M. D. Aragon, C. Bouette, G. Habault, N. Montavont and G. Z. Papadopoulos, "An efficient electric vehicle charging architecture based on LoRa communication," in *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2017.

[25] A. M. Sadeghioon, N. Metje, D. N. Chapman and C. J. Anthony, "SmartPipes: Smart Wireless Sensor Networks for Leak Detection in Water Pipelines," *Journal of Sensor and Actuator Networks,* vol. 3, pp. 64-78, 2014.

[26] R. Rahmat, I. Satria, B. Siregar and R. Budiarto, "Water Pipeline Monitoring and Leak Detection using Flow Liquid Meter Sensor," *Journal of Physics Conference Series,* vol. 190, p. 012036, 11 2016.

[27] M. Al-Fadhli and A. Zaher, "A Smart SCADA System for Oil Refineries," in *2018 International Conference on Computing Sciences and Engineering (ICCSE)*, 2018.

[28] M. T. Nasir, M. Mysorewala, L. Cheded, B. Siddiqui and M. Sabih, "Measurement error sensitivity analysis for detecting and locating leak in pipeline using ANN and SVM," in *2014 IEEE 11th International Multi-Conference on Systems, Signals Devices (SSD14)*, 2014.

[29] S. Porwal, S. A. Akbar and S. C. Jain, "Leakage detection and prediction of location in a smart water grid using SVM classification," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 2017.

[30] Q. Li, X. Du, H. Zhang, M. Li and W. Ba, "Liquid pipeline leakage detection based on moving windows LS-SVM algorithm," in *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2018.

[31] M. Chabert, J. Tourneret, V. Poulain and J. Inglada, "Logistic regression for detecting changes between databases and remote sensing images," in *2010 IEEE International Geoscience and Remote Sensing Symposium*, 2010.

Appendix A: The analysis of survey data in MATLAB®

```
A=dlmread('bathdata.txt');
power4=A(1:4,11);
power3=A(5:8,11);
power2=A(9:12,11);
power1=A(13:16,11);
power0=A(17:20,11);
datarate=A(:,10);
SNR0down=A(17:20,9);
SNR1down=A(13:16,9);
SNR2down=A(9:12,9);
SNR3down=A(5:8,9);
SNR4down=A(1:4,9);
RSSIdown=-1*A(:,8);
margin=A(:,7);
Gateway=A(:,6);
figure (1)
plot(power4,SNR4down)
hold on
plot(power3,SNR3down)
hold on
plot(power2,SNR2down)
hold on
plot(power1,SNR1down)
hold on
plot(power0,SNR0down)
xlabel('Transmission Power')
ylabel('Signal to Noise Ratio in dbm')
xlim([2 20])
ylim([0 10])
legend('DR=4','DR=3' ,'DR=2' ,'DR=1','DR=0')
figure (2)
plot(margin)
hold on
plot(RSSIdown)
xlim([1 20])
xlabel('Number of Surveys')
ylabel('Margin vs RSSI')
legend('Margin','RSSI')
```

Appendix B: The code used for acquiring data from the flowmeter

```
#include "FlowMeter.h"
#include "RawSerial.h"
int main() {
        InterruptIn* pulseInput = new InterruptIn(XBEE_DIO0);
        FlowMeter* flowMeter = new FlowMeter(pulseInput);
        RawSerial* debugSerial = new RawSerial(USBTX,USBRX);
        while (true) {
        debugSerial->printf("Flowrate: %.3f\n",flowMeter->getCurrentFlowrate());
 }
 return 0;
 }
```

Appendix C: Logistic Regression model in MATLAB®

```
% Read non-leak data from Thingspeak channel

data=thingSpeakRead(641423,'ReadKey','ACXT1MEWNON7B8EQ','Fields',[1:5],'Num
Points',600,'OutputFormat','table');
X1=data(:,3); X2= data(:, 2);
X3=data(:, 6);

X=[X1 X2];

% Read leak data from Thingspeak channel

data2=thingSpeakRead(654254,'ReadKey','8Z05YRHBBV3C71T7','Fields',[1:5],'NumPo
ints',600,'OutputFormat','table');
z1=data2(:,3); z2= data2(:, 2);
z3=data2(:, 6);

z=[z1 z2];
X=table2array(X);
Z=table2array(z);
y2 = ones(600, 1);
y1=zeros(600,1);
y=vertcat(y1,y2);
G=vertcat(X,Z);

% Training the model
mdl = fitglm(G,y,'Distribution','binomial');

% Channel created for real time Testing
[L,timestamps,chInfo]=thingSpeakRead(677317,'ReadKey','75BOG66LMMEUMTA7','F
ields',[1:5],'NumPoints',25,'OutputFormat','table');
L1=L(:,3);L2= L(:, 2);
L3=L(:, 6);

L=[L1 L2];
l=table2array(L);

ypred = predict(mdl,l);

figure(1);
 pos = find(y==1); neg = find(y == 0);
% Plot
plot(G(pos, 1), G(pos, 2), 'k+','LineWidth', 2, ...
'MarkerSize', 7);
plot(G(neg, 1), G(neg, 2), 'ko', 'MarkerFaceColor', 'y', ...
```

```
'MarkerSize', 7);
 T = zeros(size(ypred));  % Make another array to fill up...

% classify using the computed Probability
 if l==0
    R=sprintf('no water running')
 else
for ii = 1:length(ypred)
    if any(ypred(ii)>= .5)
       R=sprintf('no leak')

    else
       R=sprintf('leak')

    end
end
 end

figure(2);
plotResiduals(mdl,'probability');
```