LEAK DETECTION, LOCALIZATION AND SIZE PREDICTION IN WATER PIPELINE SYSTEMS

by

Md Toufikul Islam, B.S.

A thesis submitted to the Graduate Council of
Texas State University in partial fulfillment
of the requirements for the degree of
Master of Science
with a Major in Engineering
December 2018

Committee Members:

Semih Aslan, Chair

Harold Stern

Bahram Asiabanpour

FAIR USE AND AUTHOR’S PERMISSION STATEMENT

Fair Use

This work is protected by the Copyright Laws of the United States (Public Law 94-553, section 107). Consistent with fair use as defined in the Copyright Laws, brief quotations from this material are allowed with proper acknowledgment. Use of this material for financial gain without the author’s express written permission is not allowed.

Duplication Permission

As the copyright holder of this work I, Md Toufikul Islam, authorize duplication of this work, in whole or in part, for educational or scholarly purposes only.

DEDICATION

To my beloved father, mother, brother, sister, and wife-
for their endless support and encouragement

ACKNOWLEDGEMENTS

I would first like to thank my committee chair and thesis advisor, Prof. Aslan for his support, patience, motivation, enthusiasm, and immense knowledge during my M.Sc. research. He always guided me into the right the direction whenever he thought I needed it. Also, I am grateful to him for offering me the research assistantship opportunity in his group.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Harold Stern and Prof. Bahram Asiabanpour for their support and encouragement. I am also grateful to them for their precious comments on this thesis.

I would also like to thank Dr. Vishu Viswanathan for his valuable suggestions regarding my study, career, and research. Whenever I ran into problems and difficulties or had a question about my research, he was always there to help.

I would also like to recognize my friend Abdullah Al Bashit for his valuable suggestions throughout my research. Also, I am grateful to Bangladeshi Student Association (BSA) and Muslim Student Association (MSA) for making my stay at Texas State University so enjoyable.

Finally, I must express my very heartfelt appreciation to my beloved wife - Mehrina Nazmi for her unfailing support and continuous encouragement throughout my years of study and writing this thesis.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF ABBREVIATIONS

Abbreviation	**Description**
AWWA	American Water Works Association
EM	Electromagnetic
EPA	Environmental Protection Agency
GPR	Ground Penetration Radar
MI	Magnetic Induction
MOU	Memorandum of Understanding
PIG	Pipeline Inspection Gauge
POC	Proofs of Concept
PSI	Pounds Per Square Inch
PVC	Polyvinyl Chloride
USGS	United States Geological Survey
WCNs	Wireless Communication Networks
WUCNs	Wireless Underground Communication Networks

ABSTRACT

Wireless Sensor Networks (WSNs) consist of wireless devices that are either installed above the ground or buried under dense soil or placed in any underground spaces. WSNs have an immense future to impact on diverse applications including leak detection in water, oil and gas pipelines. Any leak in the pipe can trigger significant financial losses and possible environmental damages. This thesis presents a novel method for detecting and locating a leak in a pipe and estimating its size using pressure sensors that can detect the slightest change of pressure. A laboratory-based test bench system has been designed and developed to collect real-world datasets from sensors using a wireless sensor network. Afterward, all datasets were preprocessed, and datasets containing leak information were separated. Next, exponential curve fitting with the least square method was used to pinpoint leak location. However, leak size cannot be predicted using this method. Support Vector Machine (SVM) and Multi-layer Perceptron (MLP) neural network algorithms were then used to predict leak sizes. In our experiments, the MLP neural network showed higher accuracy over SVM in predicting leak sizes.

1. INTRODUCTION

1.1 Background

Pipeline transport systems are one of the most commonly used means for transporting water, oil, and gas. Performance of the pipeline systems is often subjected to corrosion, cracking, theft, accidental damage, and manufacturing flaws. Most of the pipes, especially underground pipelines are operated at high pressure, and in challenging environmental conditions. This can create many different malfunctions in pipelines. Hence, the assessment of the pipes is critical to prevent damage and losses.

Texas State University and NEC Corporation of America proclaimed the signing of a Memorandum of Understanding (MOU) on November 4, 2014 that establishes a partnership between NEC and Texas State University to run collaborative research and development to improve existing social infrastructure operation and management, such as water conservation and resource management [1]. Water leak detection project is one of the ongoing funded projects of the NEC at Texas State University. NEC has developed innovative technology that offers municipal water utility companies an economical way to supervise water resources. It is going to help them to meet the water necessities of the communities in which they distribute. This technology uses high accuracy sensors that accumulate data on leaks in a community's water distribution system. NEC collects and analyzes the sensor data and then offers a solution to the water utility companies. They are going to conduct Proofs of Concept (POC) of the water leak system in a few U.S. cities. As a part of this goal, in this thesis, an experimental setup has been developed, water pressure sensors have been deployed, and data from these sensors have been collected and then analyzed to obtain high accuracy in predicting the leak location and size.

1.2 The State of the World's Water Distribution System

Freshwater is crucial for good health. However, over the past several decades, about a billion people in developing countries have not had safe water. A person consumes approximately 7.5 liters of water per day for daily activities (not including shower or bath) [2]. Furthermore, at least 50 liters of water per person per day is required to guarantee all personal cleanliness, food hygiene, domestic sanitation, and laundry requirements [3]. So, keeping the water distribution systems in good shape is essential for our survival.

The demands of agriculture often overshadow the domestic water uses. As announced by the United States Geological Survey (USGS), water used for irrigation represents nearly 65 percent of the world's freshwater withdrawals excluding thermoelectric power [4]. The use of agricultural water makes it feasible to produce fruits & vegetables and increase livestock. When water is used efficiently and safely in agriculture, production increases by a margin. A leak in the irrigation pipeline system can trigger a loss of water; therefore, the production can decrease. Identifying leakage in advance can save money, minimize damage and protect property values.

Water distribution system framework is considered as a vital asset of a water utility. The system is defined by the American Water Works Association (AWWA) as "covering all water utility elements for the distribution of finished or potable water through pumps or gravity storage feed through distribution networks to customers or other users, including supply equalizing storage [5]". These systems should also be able to supply water for other uses as well, such as irrigation systems and fire suppression [6]. As the population grows and communities expand worldwide, new pipes are installed each year.

Water pipe distribution systems can be located above the ground or in the dense soil. Aboveground pipeline systems can be either suspended in support structures or rested directly on the ground surface. The system installation must be justified by any one of several factors, such as economic considerations of a temporary piping system, simplicity of inspection and maintenance, local conditions, or nature of the applications [7]. An illustration of an above the ground pipeline distribution networks is shown in Figure 1. Aboveground pipelines are made of steel, Polyvinyl Chloride (PVC), fiberglass, or copper carrier pipe, depending on the applications. A classic example of an aboveground pipeline system is the pipelines suspended on a bridge or highway, used to transport water and oil.

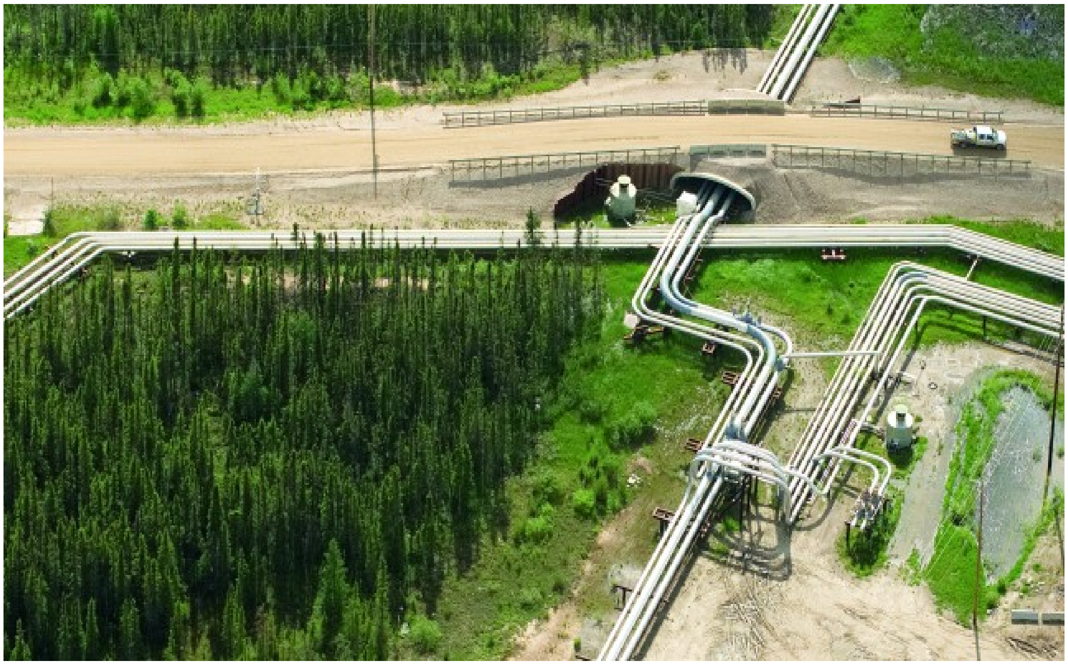

Figure 1. Above the ground pipeline system [8]

The underground pipeline system is another form of pipeline distribution network. An illustration of this system is shown in Figure 2. PVC pipes are often preferred for underground pipeline systems because of the ease of installation. Moreover, it has a higher resistance to internal friction compared to other types of pipe of a given diameter. Also, installing an underground pipeline system can be used with the same efficiency but more economically, as compared to above the ground PVC pipelines [9]. Although an underground pipeline system has several advantages, it often becomes difficult to monitor and repair.

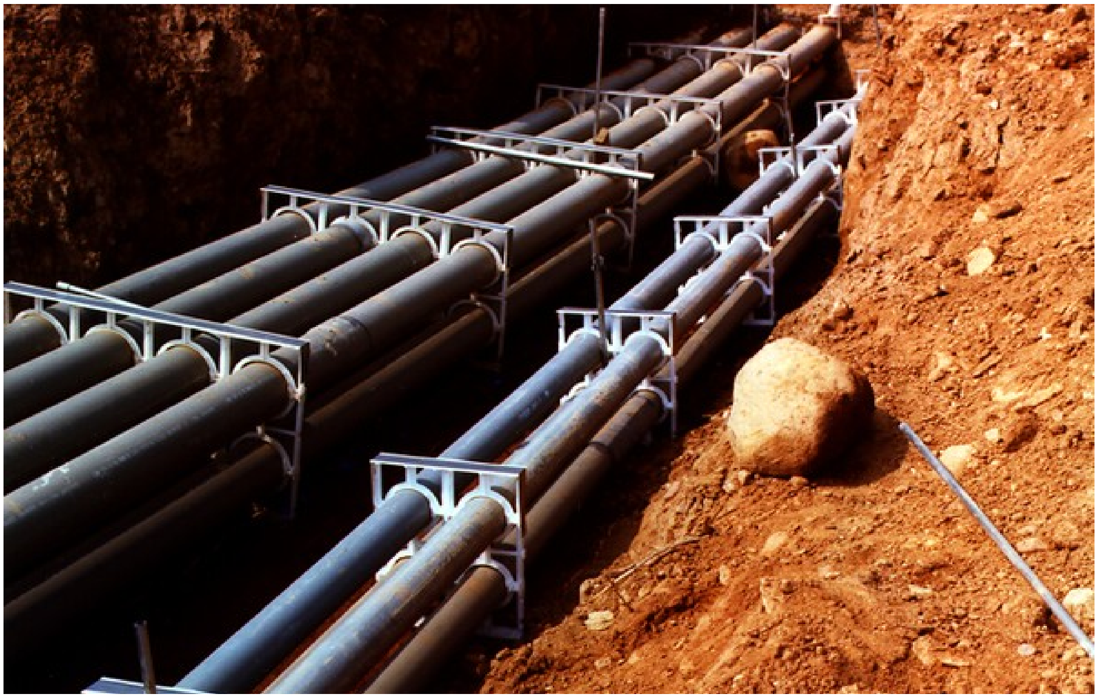

Figure 2. Underground pipeline system [10]

Pipeline systems, especially buried infrastructure makes it very hard to take control of immediate problems due to the enormous costs and efforts involved in digging the surface [11]. Leaks over long periods of time generate many challenges, including the loss of water, the chemicals used to treat leaks, or possible contamination of the drinking water inside the pipe [12]-[13]. Any leak in the pipeline can lead to significant financial losses and probable environmental hazards. In September 2009, the burst of 62-inch water main washed away cars and flooded several homes in Studio City, Los Angeles [14]. In 2014, more than 4.76 billion gallons of drinkable water seeped from Austin pipelines, which is enough to fill Lady Bird Lake twice [15]. Ten percent of homes in the USA have leaks that waste 90 gallons or more per day. Leaks in the household pipeline can waste more than 1 trillion gallons annually nationwide. That's identical to the annual household water use of more than 11 million homes [16]. According to an assessment of the public water system by the US Environmental Protection Agency (EPA) in 2018, $472.6 billion is needed to refine the infrastructure for thousands of miles of pipe as well as thousands of plants, storage tanks, and other vital assets in the next 20 years [17].

Continuous remote monitoring and assessment of the pipelines are necessary to prevent losses. A better system for pipeline monitoring can easily lead to a reduction in leakage, which could be very vital especially in a few situations where people undergo water shortages. Leak detection is a very significant and demanding job, therefore, the prime focus of this thesis.

1.3 Leak Detection Methods

With the new generation of electronics, sensors have become smaller, less expensive, and more sophisticated. As a result, it is possible to produce more data which allows more accurate assessment of a system, warning of environmental threats, and sensing of problems. These improvements lead to installing the advanced pressure sensor in the pipeline to detect leaks. Researchers have used various methods to identify and locate holes in the pipeline system. Acoustic measurements, vision-based systems, fiber optic monitoring, Ground Penetration Radar (GPR) based systems and multimodal systems are the most common methods [18].

A notable amount of research has already been done on acoustic or vibration measurements for pipeline monitoring [19], [20], [21], [22], [23]. Most of these methods are regarding the detection of acoustic emissions from the pipe. Leakage in pipes creates vibrations which are transmitted along the pipe walls. These waves can be detected by using acoustic sensors or accelerometers installed on the pipe wall for analysis [19]. The leak location can be identified using different cross-correlation methods. Although there are several advantages of using this technique, some disadvantages make it unsuitable to use, especially when it comes to the underground wireless network system. It requires a high sampling rate to measure the acoustic signal. So, due to the high consumption of power in the nodes, the lifetime reduces significantly. Moreover, the system requires a sophisticated algorithm to process a large data set that also increases power consumption in the nodes.

Vision-based systems utilize a Pipeline Inspection Gauge (PIG) with a form of image processing or laser scanning to find rifts and faults in pipelines [20]. These systems demand access to the inside of the pipe to control. Besides, they take measurements at long time intervals because it is costly to install the PIGS inside the tube. Moreover, they need high handling power or a highly skilled operator to examine the outcome [24].

Fiber optic technology has a different level of success regarding leak detection [25], [26]. A leak can be detected using this method by installing optical cables over vast distances. However, the system is complicated to install. Also, in many cases, fiber cable needs to be introduced during pipe installation. Moreover, if a pipe section needs to be repaired or replaced, the fiber optic system could be out of service in that area [27].

GPR is a method that uses high-frequency electromagnetic waves to obtain information. GPR answers to changes in electrical properties, which are a function of soil or rock material and moisture content. Then a few laboratory experiments are conducted to determine the leak [28]. However, the accuracy of these systems highly relies on the soil type and condition and they are more suitable in dry soil conditions. Furthermore, these systems are not useful for the continuous monitoring of large pipe networks [18].

Sensors play a pivotal role in detecting a leak in the pipeline system. Liu and Kleiner [29] study the sensor technologies used in monitoring pipe structural failure. The multimodal wireless sensor network is becoming more popular as it uses low power sensors [18], [30], [31]. Even sparse deployment allows these systems to have an acceptable level of redundancy within the system. Moreover, the flexibility of deployment makes spot monitoring possible where pipe monitoring is only needed for a limited segment of the pipe [18].

1.4 Scope and Emphasis

The problems associated with leak detection are to find the location of the leak & and to accurately predict its size with different water pressure levels and different leak sizes. Sometimes it becomes hard to identify leak location with less water pressure and mainly when the leak size is tiny. This paper solves the problem of identifying leaks with less pressure and small leak size by designing a simple sensor network and analysis method.

The system analyzes a large number of data sets where each set contains pressure values of 6 sensors to identify specific data sets that leak. Each data set is 2-5 minutes of a text file, and the wireless sensor network is used to transfer the data sets to a computer. Next, this thesis reflects a novel methodology for uncovering informative patterns from data which includes the preprocessing, feature extracting and designing a classifier model to identify the leak location and its size with relatively high accuracy.

1.5 Outline of Thesis

The coordination of this thesis is as follows: Chapter 2 presents a review of some previous works related to signal propagation techniques for wireless sensor network and data acquisition systems. Next, chapter 3 presents a description of the proposed system which contains a detailed description of sensors, wireless system, and data analysis. Experimental analysis and results are shown in chapters 4 and 5 respectively. Finally, chapter 6 presents our conclusions and suggestions for future work.

2. LITERATURE SURVEY

This chapter will introduce some of the previous works related to our research. This thesis is mainly about leak location identification and leak size prediction in pipeline distribution systems. Before leak detection, it is essential to understand the radio propagation technique. Moreover, it is also important to know what sensor can be used in leak detection and how to transmit, receive, and process sensor data. This chapter has been divided into four parts. In the first part, the signal propagation technique is introduced, followed by the sensor selection in the second part. Data acquisition techniques and data classification techniques are studied in the third and fourth parts respectively.

2.1 Signal Propagation Technique for Wireless Sensor Network

Both Wireless Underground Communication Networks (WUCNs) and Wireless Communication Networks (WCNs) consist of wireless devices that operate above and below the ground surface respectively. The main difference between WUCNs and WCNs is the communication medium. Buried devices have the difficulty of transmitting the sensed information back to the surface because typical data transmission systems do not work well due to the challenge of propagating a RF signal through the soil. The general formula for the received signal, P_r, at the receiver end can be expressed as (1):

$$P_r = P_t + G_r + G_t - L_o \tag{1}$$

Here, P_t is the transmitted power, G_r and G_t are the gains of the receiver and transmitter antennae, and L_o is the path loss in free space. However, in the case of underground signal propagation, the signal experiences additional path loss when it travels through an underground medium such as soil. The received signal, P_r, at the receiver end for the underground communication can be expressed as (2):

$$P_r = P_t + G_r + G_t - L_o - L_u \tag{2}$$

Here, L_u is the additional path loss caused by the propagation in the underground medium.

So far, there has been a great deal of research on signal propagation in soil medium. As demonstrated in [32], the authors emphasized the signal propagation for WUSNs. They provided a channel model for both electromagnetic (EM) waves and magnetic induction (MI). Then they compared both techniques regarding path loss while using different frequencies for signal propagation. The comparison between the EM wave, MI, and MI waveguide is shown in Figure 3 for the 900MHz signal.

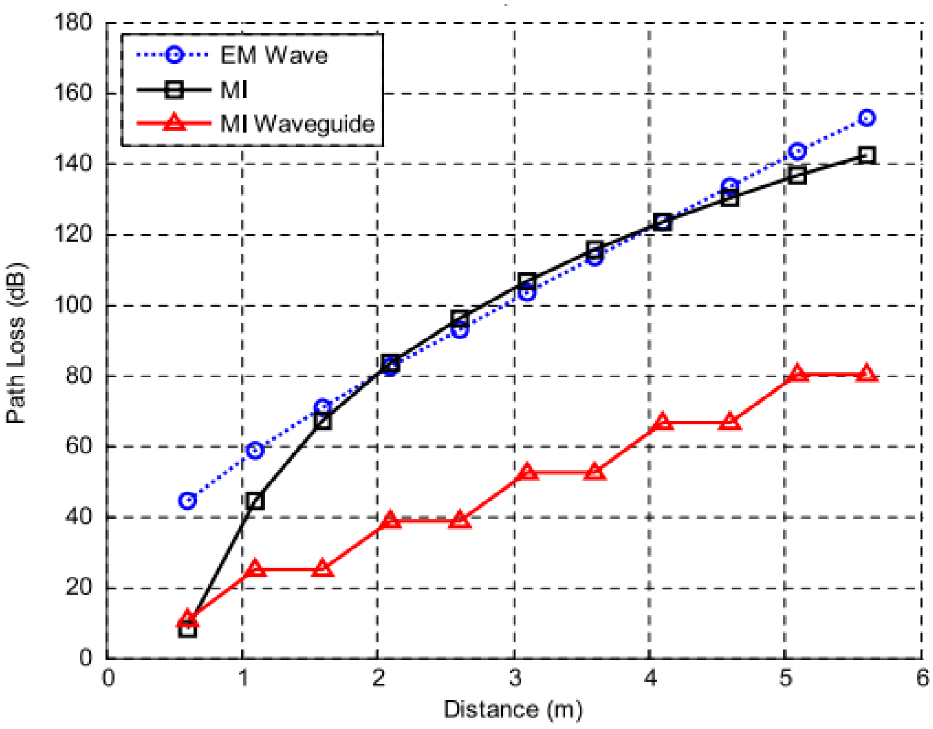

Figure 3. 900MHz signal in the soil [32]

In Figure 3, the horizontal axis on the graph shows the distance a signal can travel in the underground and the vertical axis represents path loss. It shows, when the signal moves more distance in the underground, it experiences more path loss. Also, the path loss of EM waves and MI is almost the same for the operating frequency of 900MHz. These curves are fundamental analysis considered for the thesis. In this thesis, the EM signal was transmitted and received wirelessly using 900MHz operating frequency.

Jiang, Georgakopoulos, and Jonah [33] focused on computing the transmission loss and propagation loss of RF waves penetrating soil at various frequencies and depths. Figure 4 depicts the total loss during the penetration of the signal from air through ground or vice versa. This is a very useful analysis considered in this thesis during system design. It helped us to choose the right standard module for wireless communication.

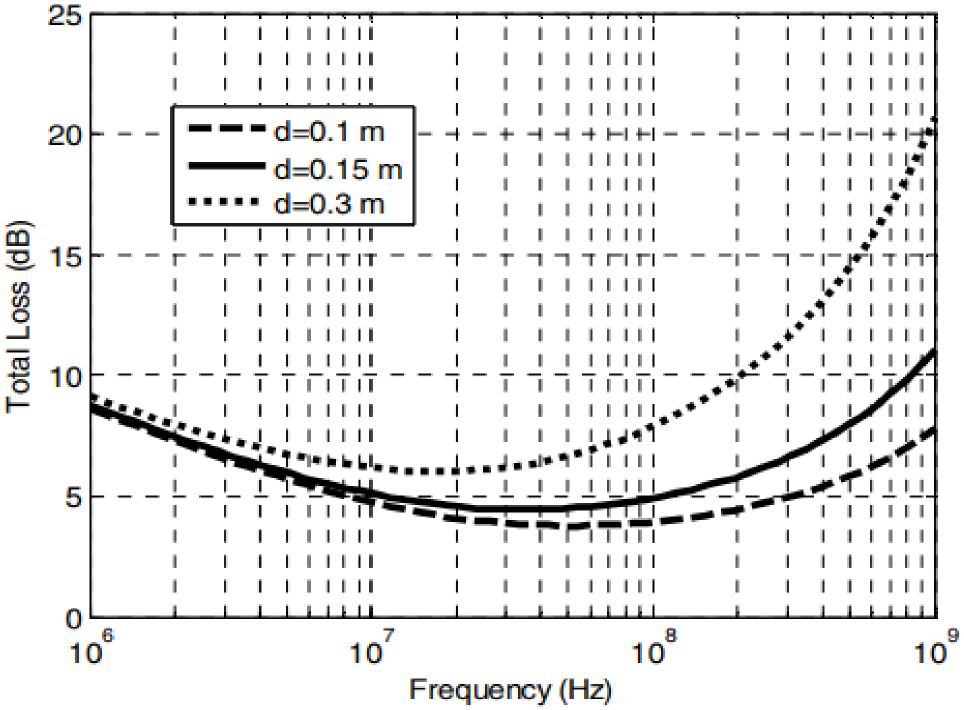

Figure 4. Total loss with different depth d and frequency [33]

As seen in [34], the authors presented an estimation of extending underground communication radius using the underground radio propagation model. Using this estimation, they proposed a concept of low-frequency wireless signal networks for the subsurface monitoring applications. They buried sensor nodes at different depths and calculated received signal strength. Figure 5 shows received signal strength at various sensor locations.

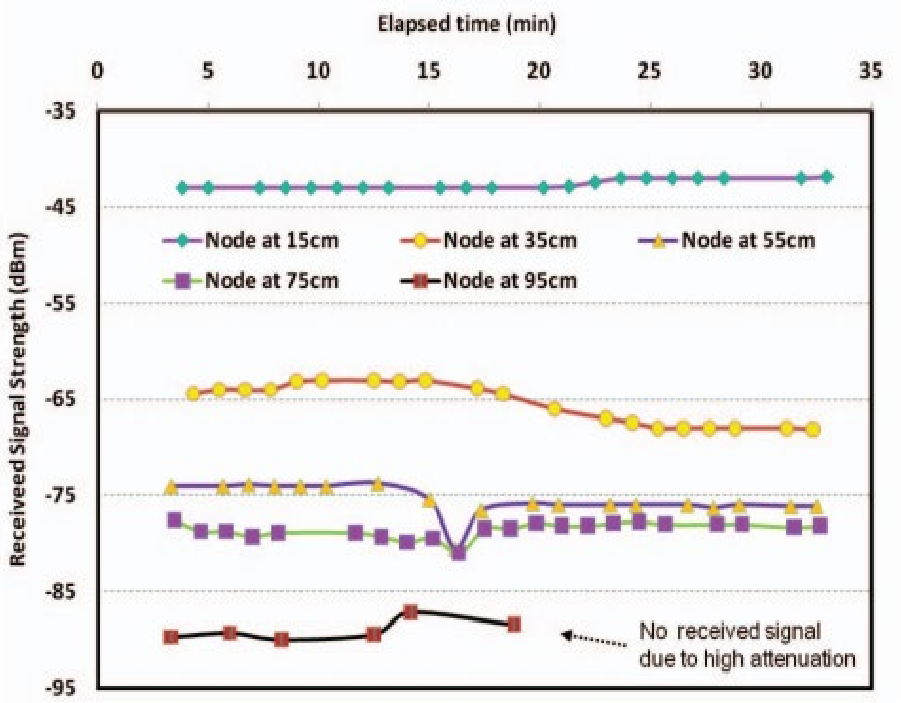

Figure 5. Received signal strength with different depth [34]

As shown in Figure 5, received signal strength is reduced significantly when the distance between earth and buried sensors increased. At 95cm, the receiver lost the signal due to high attenuation in the underground. This analysis gave us a general idea about sensor positioning in the underground.

2.2 Sensor Selection for Leak Detection

Over a decade, the concept of a wireless sensor network has been investigated for leak detection. In most of the studies, researches considered the multimodal system to detect a leak. To our best knowledge, one of the most relevant works has been done by Sadeghioon et al. [18]. They used the combination of Force Sensitive Resistors (FSR) and temperature sensors to detect a leak. They collected pressure data generated from the FSR sensor and temperature data from the temperature sensor for three days. As the name implies, the FSR sensor is very responsive to the force applied to it. When water goes through the pipe, it creates pressure that is detected by the sensor. However, the main drawback of the FSR sensor is that it exhibits low accuracy. Also, the authors concluded about a leak by observing the pressure profile (Figure 6) of every FSR sensors.

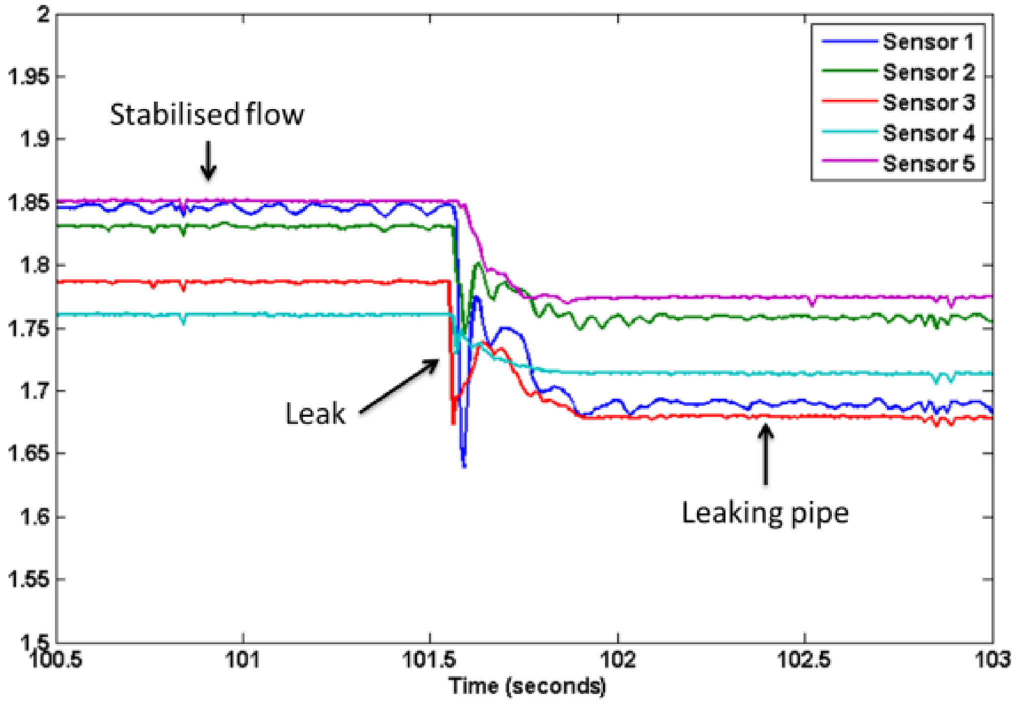

Figure 6. Pressure profile of five FSR sensors due to the occurrence of a leak [18]

Authors [18] claimed leak location could be identified easily by just looking at the downstream pressure profile of sensors due to the leak. However, in many practical conditions it becomes hard to determine the leak location by this method. In this thesis, a similar laboratory-based test bench has been constructed as in [18, (see Fig. 8)]. Furthermore, data were analyzed and taken a few steps further to make the system more efficient.

Lakshmi and Gomathi [30] used five flow sensors in their experiments to determine the leak. The performance of the flow sensor is impacted by dirt and fluid which is a significant disadvantage. As seen in [31], the authors followed the same procedure as [30] but used humidity, temperature, pressure, and gas sensors to determine leak location. The leak detection system becomes more efficient using more types of sensors. However, more types of sensors prompt more power consumption at the sensor node. In this thesis, we used only pressure sensors that reduced power consumption at a sensor node while maintaining acceptable accuracy to detect the leak, its location, and size.

2.3 Data Acquisition System Design

Wireless tools are usually used in locations that are hard to access due to severe conditions such as high temperature, pressure, humidity, pH, etc. Operators can continuously monitor processes using remote sensors. Moreover, several different locations can be observed from one station using a wireless sensor network. One of the biggest challenges, when using WSN, is to accumulate data from different sensor nodes [35].

So far, plenty of research has been done on the data acquisition systems using a wireless sensor network. As seen in [36], the authors have developed their own Zigbee (XBee) module to transfer data wirelessly. They claimed that the wireless system in the ZigBee network has very high reliability and battery lifetime of 10 years. They used two Zigbee routers to send data and one Zigbee coordinator to receive it. They calculated the delay time for each of the routers with 1 & 3 hop count as shown in Figure 7. The delay time is caused by the processing time needed in the ZigBee stack. This analysis was very crucial because it gave a vital piece of information about data transfer, hence, considered in this thesis.

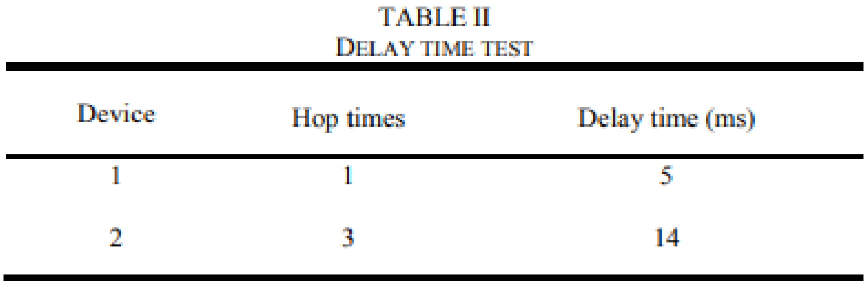

TABLE II
DELAY TIME TEST

Device	Hop times	Delay time (ms)
1	1	5
2	3	14

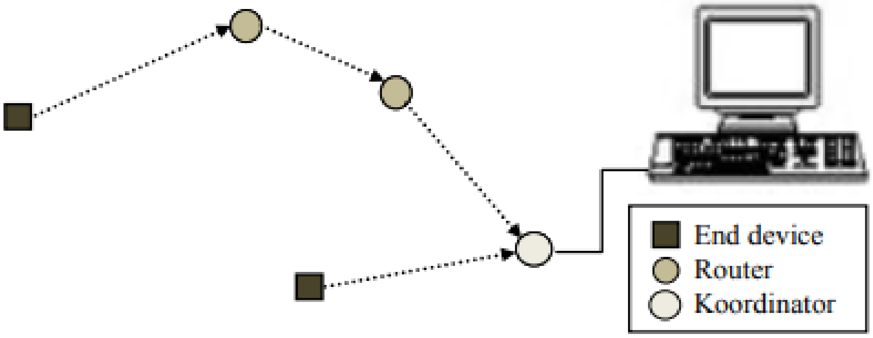

Figure 7. Delay time test for ZigBee router [36]

The connectivity between wireless sensor network standards is also studied by Lakshmi and Gomathi [30]. They explained Zigbee topologies such as star, cluster, and mesh. In this thesis, all the topologies have been studied and star topology has been selected as the right fit for this thesis.

In the home environment or a small irrigation system, installation of various sensors that are wirelessly connected has been studied over the past decade. For this type of wireless sensor networks, [37] showed, the ZigBee standard can be a good choice. As seen in [38], the authors explained the importance of using Zigbee for data communication. They also compared the data rate between the two operating frequency bands of Zigbee. Dang and Cheng [39] designed a wireless sensor network in the monitoring system based on Zigbee. They explained how Zigbee network topologies work in a communication system. These are all critical analysis considered for this thesis.

2.4 Data Classification Technique Selection

The water pressure profile of sensors can vary at different times of the day or night depending on pressure level and leak sizes that make the whole leak detection method complicated. Computational pattern recognition techniques such as Support Vector Machine (SVM) and Artificial Neural Network (ANN) are especially useful for this type of problem.

To our best knowledge, one of the most relevant works has been done by [40]. They used Support Vector Machine (SVM) to pinpoint leak location and determine leak size. The maximum accuracy obtained by them was 74.4% and 57.25% while using 10 and 40 sensor nodes respectively.

As seen in [41], they extended the work of [40] but used differential pressure sensors instead of gauge pressure sensors to get higher accuracy. They used ANN alongside

SVM and then compared these two methodologies. They showed ANN is better in the presence of minimal noise where SVM is less sensitive and more stable to noise increment. This analysis helped us to understand which technique performs better in what condition.

The work in [42] used the SVM classification technique to analyze pressure and flow data at different locations on the distributed network. They claimed the maximum efficiency in detecting a leak is about 90% using ten nodes only.

In all the references mentioned in this section, sensor data were obtained by simulating the water distribution network using the EPANET tool. EPANET is used to model water distribution systems. The tool is used to simulate the water quality behavior within pressurized pipe networks, which consist of pipe types, junctions, pumps, etc. [43]. However, in our experiment, we have developed our test bench system to predict leak and its size under real-world conditions for a given network. Although a vast number of datasets were not possible to obtain from our test bench system, collected data were enough to show acceptable accuracy to determine leak location and size.

3. PROPOSED SYSTEM

In this thesis, three mathematical tools were utilized to accomplish water leak location identification and leak size prediction. The first one is exponential curve fitting using least square regression, a novel approach to pinpoint the leak location. Typically, data from two sensors separated by a correct distance is enough to pinpoint leak location using this method. The second tool is the Support Vector Machine (SVM) that has been widely used by researchers to classify data in different categories. In this thesis, SVM is used to predict leak sizes. The third tool is another classifier called Multi-layer Perceptron (MLP) and is used to predict leak sizes as well. MLP is a remarkable tool to extract output from noisy and complicated data.

This chapter has been divided into four parts. In the first part, the test-bench system is introduced, followed by the pressure sensor characteristics used in this thesis. In the second part, the data acquisition system is explained, followed by the features of Arduino and ZigBee. Next, data processing methodology is described in the third part supported by the theory of exponential curve fitting and feature scaling. The fourth part describes the functions of SVM and MLP classifier.

3.1 Pipeline Model

A test bench system is a system used to justify the soundness of a model or design. A laboratory-based test bench system has been designed and developed to solve the leak detection problem and make the model effective and efficient. The system was comprised of PVC pipes, three leaks, six pressure sensors, one water tank, and one water pump. PVC pipes with three different diameters: 0.75-inch, 1-inch, and 1.5-inch were used for the experiment. The test bench system is illustrated in Figure 8.

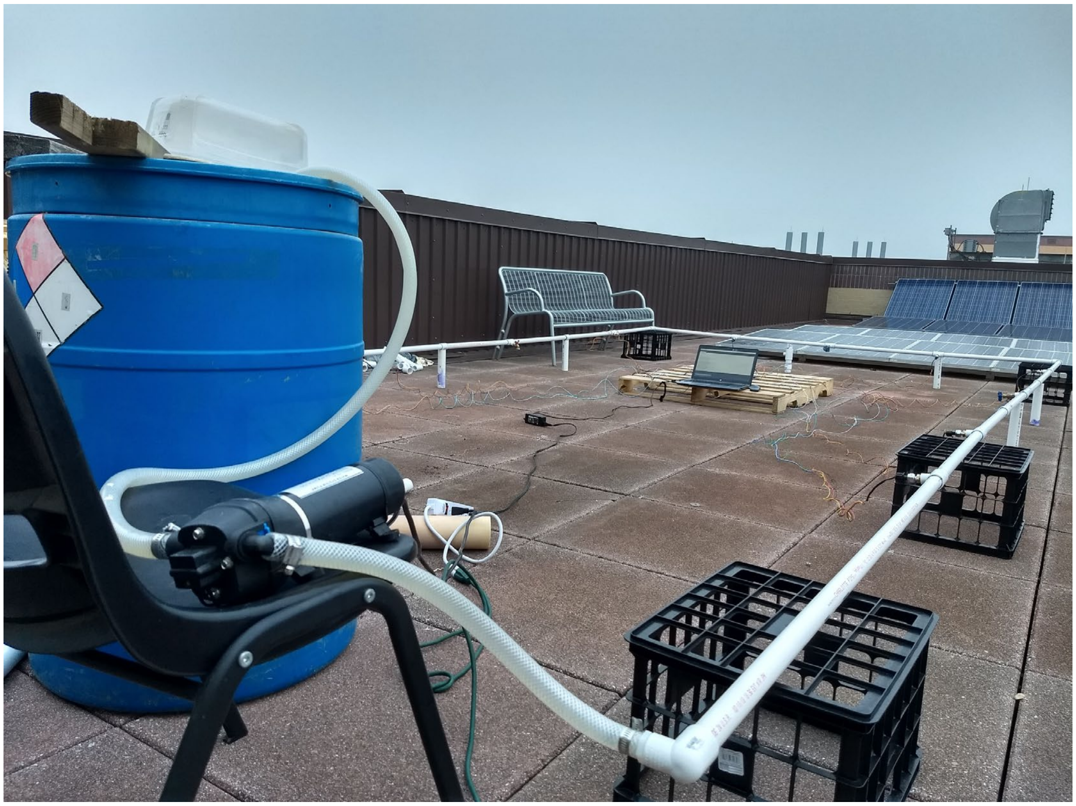

Figure 8. Laboratory-based test bench system.

The pump stand was placed at the inlet end of the pipeline system. The pump stand was tall enough to provide the pressure required at the outlet. Also, pump stands size was more significant than the diameters of the pipeline, to dissipate a high-velocity stream and release entrapped air before water enters the pipeline [10]. PVC pipes were used because they are lightweight, flexible, longer in length, versatile, flame resistant, and inexpensive. A conventional water tank was used to release and collect water.

3.1.1 Pressure Sensor

A pressure sensor is an instrument that senses the pressure of the gas, water, or oil and transforms it into an electric signal where the amount depends on the applied force. Pressure sensors control and monitor thousands of everyday applications. The pressure sensor can be different types such as an absolute pressure sensor, gauge pressure sensor, or differential pressure sensor depending on whether the pressure is relative to vacuum, atmosphere, or other measuring elements. The absolute pressure sensor detects the pressure corresponding to a perfect vacuum. On the other hand, the gauge pressure sensor determines values relative to atmospheric pressure. One side of this sensor is usually exposed to ambient conditions. A differential pressure sensor measures the difference between two pressures, one connected to each side of the sensor. It calculates pressure drops across air filters, fluid levels or flow rates. The differential pressure sensor can be challenging to use due to the existence of two different pressures or more on a single mechanical structure [44].

In this thesis, gauge pressure sensors, specializing in the water and oil leak detection technology, were used as shown in Figure 9. This analog Pressure Sensor is a 5V sensor that can measure pressures up to 200 pounds per square inch (PSI). These sensors are small, lightweight, waterproof and thus easy to handle. These sensors have built-in carbon steel alloy and can easily be installed and require no special handling. Most importantly, these sensors are inexpensive and that makes the whole leak detection system inexpensive yet efficient. These are analog pressure sensors are identified as model SKU237545 in the market.

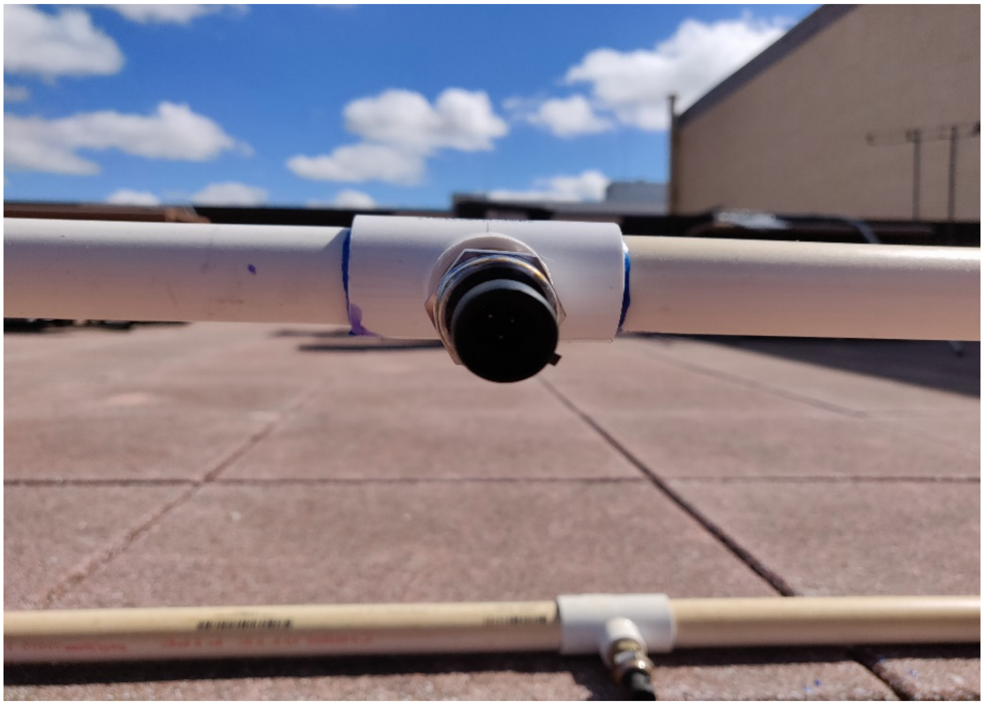

Figure 9. Gauge pressure sensor

There are some disadvantages of an analog pressure sensor. The efficiency of the pressure sensor decreases with time; thus, accuracy reduces. Although the accuracy of an analog pressure sensor is not as precise as a digital sensor, it provides excellent value for the cost. Table 1 presents the specification of the pressure sensor. It indicates the pressure sensor will work at a very high temperature and pressure, therefore, the sensor is suitable for installing both in the underground and aboveground. Also, measuring and temperature range error of these sensors are very less which is very important for leak detection and localization.

Table 1. Specification of the pressure sensor

Working Voltage	DC 5.0V
Output Voltage	DC 0.5-4.5 V
Sensor material	Carbon steel alloy
Working Current	≤10 mA
Working Pressure Range	0-1.2 MPa
The Biggest Pressure	2.4 MPa
Destroy Pressure	3.0 MPa
Working TEMP. Range	0-85℃
Storage Temperature Range	0-100℃
Measuring Error	±1.5 %FSO
Temperature Range Error:	±3.5 %FSO
Response Time	≤2.0 ms
Life Cycle	500,000 pcs

3.1.2 Leak Simulation

Hose bibs are threaded faucets that can be found on the outside of a home where the hose is fitted. The shutoff valve attached to them allows easy access to water. Hose bibs are commonly connected to PVC, copper, or galvanized piping. In this paper, leak simulation was created by using a hose bibb as illustrated in Figure 10. The advantages of using a hose bibb are: (i) The size of the leak can be controlled manually using shutoff valve, and (ii) The pressure drop generated by the leak is sharp enough to be detected by the pressure sensors. The hose bibb is easy to use and helps simulate various predefined leak sizes. The diameter of the shutoff valve represents different leak sizes. The maximum diameter of the shutoff valve used in the system is 0.5-inch. As diameter can be varied, the system has been tested with five different leak sizes: 0.5-inch, 0.4-inch, 0.3-inch, 0.2-inch, and 0.1-inch.

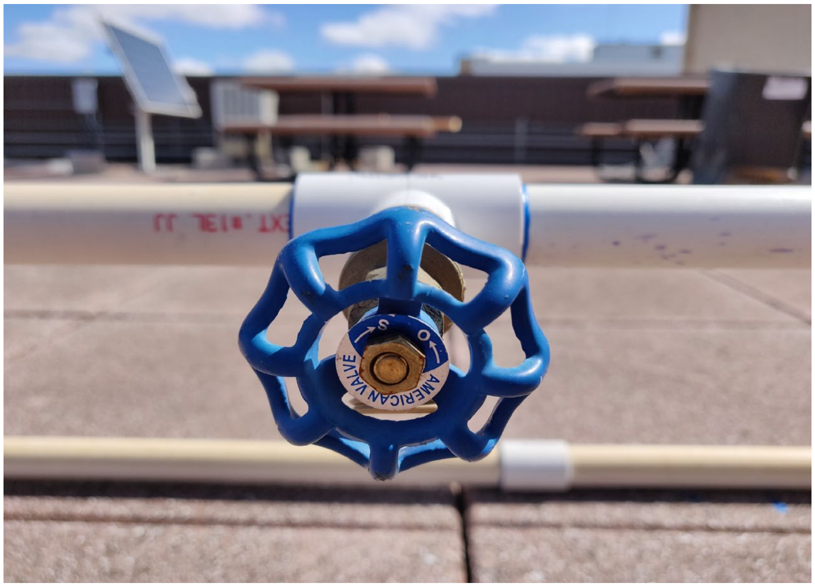

Figure 10. Hose Bibs

The hose bibb has a total number of 13 patches. In another word, if we rotate the handle 13 revolutions in the clockwise direction, the valve will open with the maximum diameter of 0.5-inch. The total number of revolutions are divided into five categories to determine leak diameter. 13, 10, 8, 5, and 2 revolutions to represent 0.5-inch, 0.4-inch, 0.3-inch, 0.2-inch, and 0.1-inch leak diameter respectively.

3.2 Data Acquisition

Data acquisition is the method of transforming data from one state to another state that is acceptable to the computing device for advance processing. The system consists of sensors, hardware, and a computing device with programmable software as shown in Figure 11. Hardware collects data such as pressure, temperature, or sound from sensors and converts the data into suitable formats that can be processed by a computer.

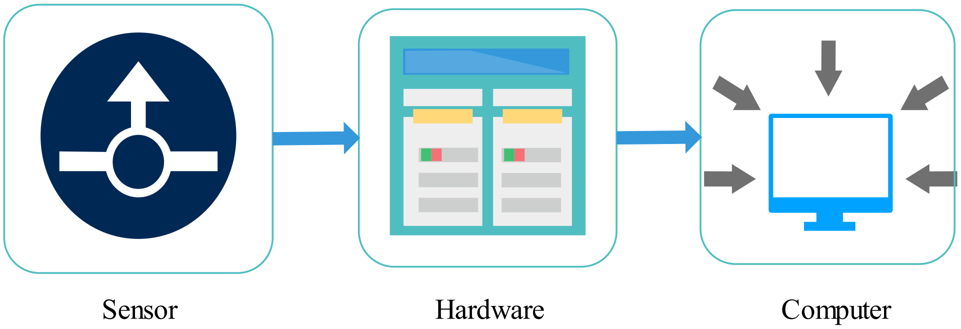

Figure 11. Data acquisition system

In this paper, Arduino and ZigBee (XBee) were used to convert raw pressure data from the sensor and to transmit it wirelessly for further processing. This whole scenario is illustrated in Figure 12. Pressure sensors were connected to Arduino by wires. The wireless communication was established between two ZigBee modules. The Zigbee receiver was connected to PC by cable. The details about Arduino and ZigBee are discussed in the next two sections.

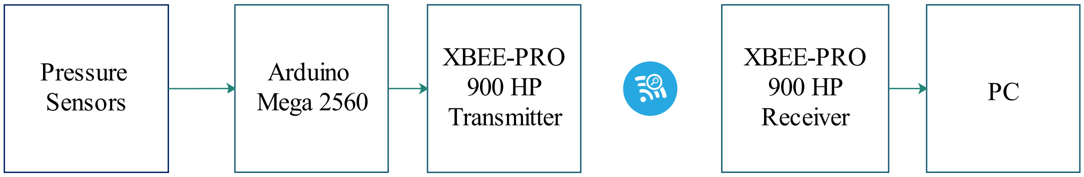

Figure 12. Data acquisition using Arduino and Zigbee

3.2.1 Arduino

Arduino is an open-source platform utilized for constructing electronics research projects. Arduino comprises both a physical programmable circuit board which is often referred to as a microcontroller and a piece of software that is called the Integrated Development Environment (IDE) which runs on a computer and is used to write and upload computer code to the board. Arduino is inexpensive, easy to use, and straightforward to program. Over the years Arduino has been the intelligence of thousands of projects starting from everyday objects to complex scientific instruments. Arduino builds several different boards such as Uno, Due, Mega, and Leonardo with distinct capabilities [45]. In this thesis, Arduino Mega-2560, shown in Figure 13, is used to collect and convert data into a usable format. Arduino Mega-2560 has 16 analog input pins. Therefore, 16 analog sensors can be connected all together. It reduces the number of sensor nodes in the system, thus, reduces the complexity of the system.

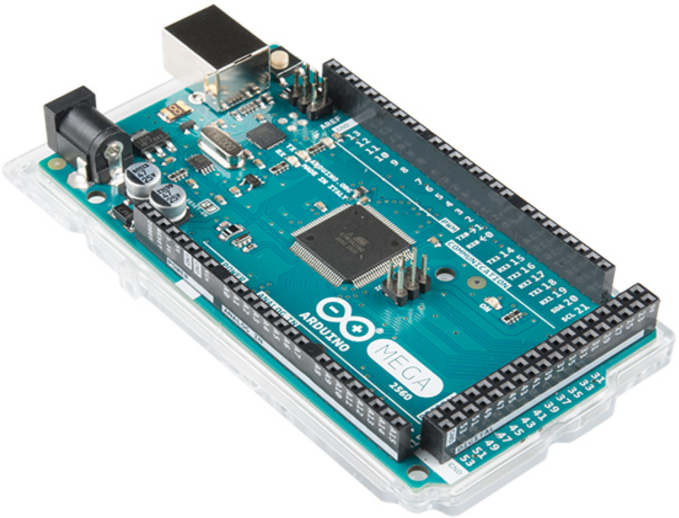

Figure 13. Arduino Mega-2560 [46]

The full specification of the Arduino Mega-2560 board is given in Table 2.

Table 2. Specification of Arduino Mega-2560

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

3.2.2 ZigBee (XBee)

ZigBee is based on the 802.11 standards as outlined by the IEEE (the Institute of Electrical and Electronics Engineers). It is used to create small personal area networks. It is low cost and requires less power. Zigbee is simpler and less expensive than Bluetooth and Wi-Fi. Zigbee can transmit up to a distance of 10-100 meters with RF data rate of 250 kbps. ZigBee can be configured in either AT command mode or API mode. Zigbee devices are configured using XCTU software. To establish a wireless connection between two

Zigbee devices, they need to be operated in AT command mode. One ZigBee is set as coordinator, and other is set as a router. The router is connected to the receiver side. On the other hand, the coordinator is attached at the transmitter side [47]. ZigBee has three network topologies: Cluster, Star, and Mesh [48] are shown in Figure 14.

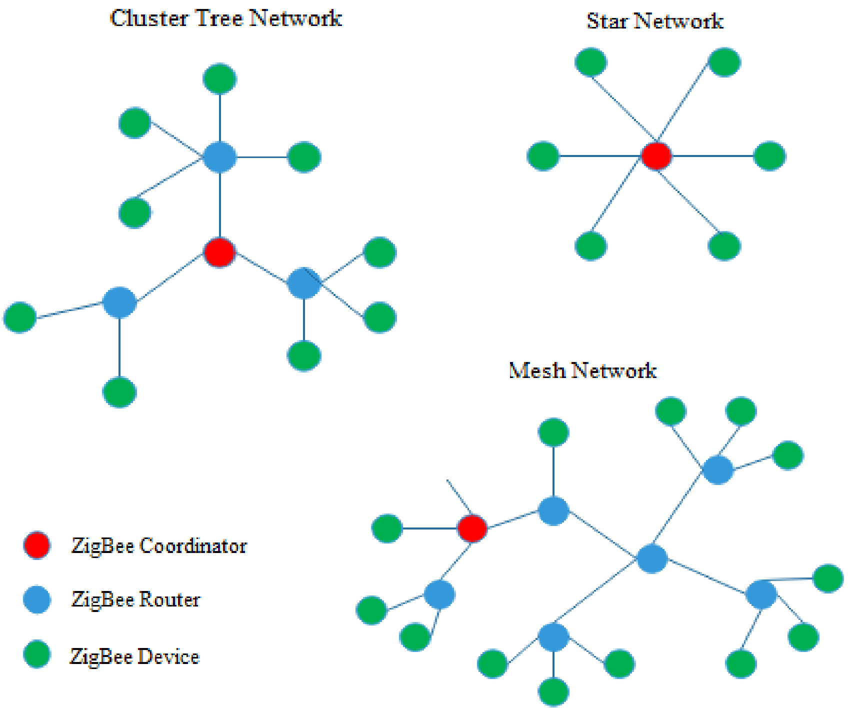

Figure 14. ZigBee Cluster, Star and Mesh Network

In this thesis, star network topology has been developed for this system where two routers were wirelessly connected with a single coordinator to cover more pipeline area. Three sensors were attached to one ZigBee router, and another three sensors were attached to another ZigBee router. One ZigBee coordinator was used to collect data from two ZigBee routers. The arrangement was made for the demonstration on how researchers can monitor a large pipeline using a wireless sensor network.

ZigBee devices communicate with each other over the air, sending and receiving wireless messages. The devices only transfer those wireless messages; they cannot manage the received or sent data. However, they can communicate with intelligent devices via the serial interface. ZigBee devices transmit data coming from the serial input over the air, and they send anything received wirelessly to the serial output. Whether for communication purposes or for merely configuring the device, a combination of both processes makes ZigBee communication possible. In this way, intelligent devices such as microcontrollers or PCs can control what the ZigBee device sends and manage incoming wireless messages [49]. For wireless communication, the antenna plays a vital role. Several options can be selected for ZigBee communication such as a chip, wire, U. FL, trace, and RPSMA antenna. The chip antenna is the universal option for the ZigBee module, but it has the worst performance of all the possibilities regarding coverage and sensitivity. Wire antenna & U. FL antenna is very tiny but have more extended transmission range advantage over chip antenna. RPSMA is big and suitable for communication where the object is in a box and the antenna is outside the box [50].

Most of the ZigBee modules operate at 2.4GHz, but there are a few that run at 900MHz. 900MHz can cover more area with a high gain antenna because the lower the frequency, the higher penetration the signal has. In this thesis, XBEE-PRO 900HP module, as shown in Figure 15, was used. XBEE-PRO 900HP supports point-to-multipoint configuration. Also, it has the capability of supporting line-of-sight range up to 28 miles with the high gain antenna. In this thesis, the wire antenna integrated with Zigbee is used for wireless communication because it’s very tiny and can penetrate earth surface.

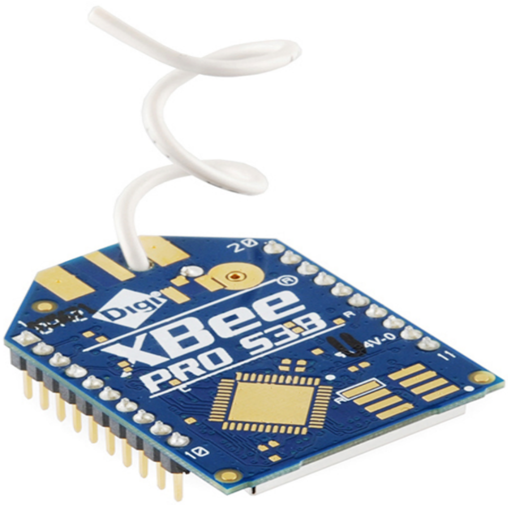

Figure 15. ZigBee module with a wire antenna [50]

The specification of XBEE-PRO 900HP is given in Table 3. As XBEE-PRO 900HP provides the best transmission range for communication and better receiver sensitivity, thus, it is considered for this thesis.

Table 3. Specification of ZigBee-Pro 900HP

Standard	ZigBee-Pro 900HP
RF data rate	10 Kbps to 200kbps
Indoor range	10Kbps reaches up to 600m 200Kbps reaches up to 305m
Outdoor range	10Kbps reaches up to 15.5km 200Kbps reaches up to 6.5km
Antenna options	Wire, U. FL, or RPSMSA
Receiver sensitivity	-101dBm at 200Kbps -110dBm at 10Kbps

3.3 Data Processing

Data processing can be defined as a process of converting data into usable and desired form. This process is achieved using a predefined order of operations either manually or automatically. In most of the cases, data processing is done by using computers and thus done automatically. The difficulty of this process depends on the scale at which data collection is completed and the complication of the results which need to be acquired. The processing time depends on the operations which need to be done on the collected data and on the characteristic of the output file required to be obtained [51]. In this paper, the exponential curve fit model was used to process the data collected from pressure sensors.

3.3.1 Exponential Curve Fitting

The trend in the dataset can be captured using curve fitting by allocating a single function across the entire range. A process of approximating the pattern of the result is known as curve fitting. Exponential curve fitting is used when the rate of change of a quantity is proportional to the initial amount of the quantity. In this thesis, raw data has been collected from sensors as shown in Figure 16. It is clear from the picture, that the pressure value generated by the sensors, drop immediately after the development of the leak. Due to a decreasing trend of pressure with respect of time, an exponential curve fitting has been used to fit the pressure vs. time data. This curve can be expressed as (3):

$$f(t) = ae^{bt} \tag{3}$$

Here, b is the decay constant where $b < 0$, and t denotes time.

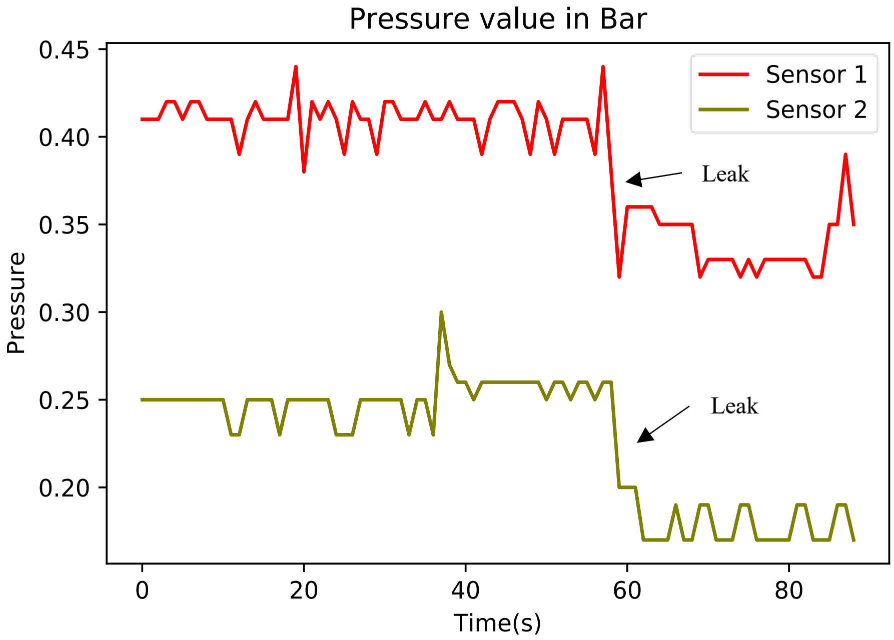

Figure 16. Raw data from pressure sensors

There are two advantages of using this method: (i) Discrete data points are represented as a continuous function and (ii) The magnitude of the decay constant ‘b’ in f(t) gives the measurement on the rate of pressure decrease in the sensor. The constants ‘a’ and ‘b’ in f(t) can be estimated and applied to the data set of each pressure sensor using a least squares approach.

The hypothesis is that the pressure drop should be most significant in the sensor that is closest to the leak. This pressure drop should reflect in a larger magnitude of the decay constant ‘b’ in the curve fit analysis. By comparing the decay constant ‘b’ for all six sensors, the leak location can be predicted.

3.3.1.1 Least Square Regression

In the case of simple fitting of data to a straight line, data points need to be plotted on a graph, then a straight edge is placed on the figure and moved it until an optimum straight line is formed. The best line does not necessarily need to pass through any of the data points, but it is close to all of them. The general form of a straight line can express by Equation 4.

$$f(x) = ax + b \tag{4}$$

When a line drawn on a graph contains lots of data points (x_i,y_i), it creates an estimated value for every data point. So, for every point of the dataset, an actual and estimated value can be found. The difference between actual and estimated value is known as error and is shown in Figure 17.

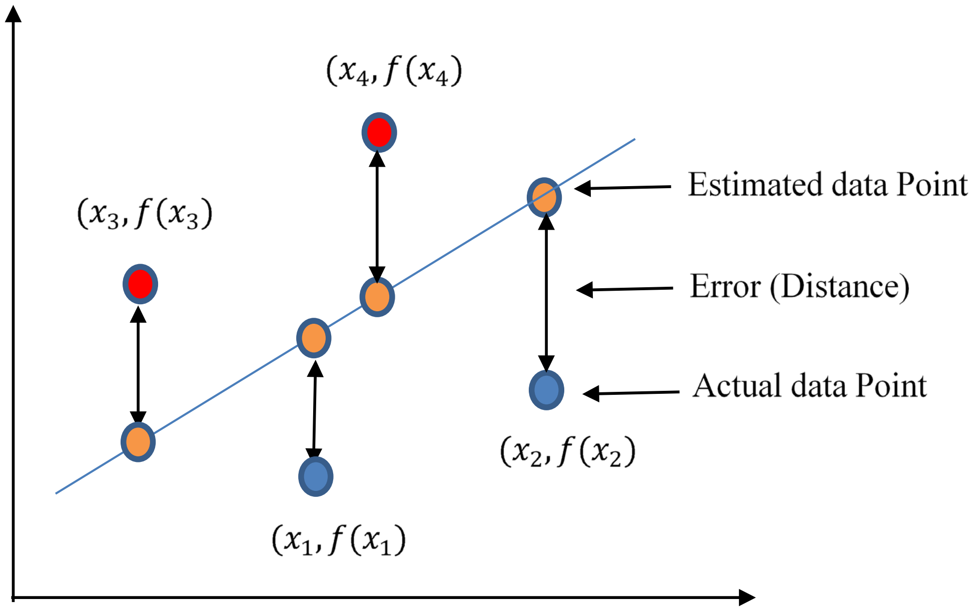

Figure 17. Least square regression

The error can be calculated by measuring the distance between actual datapoint and estimated datapoint. The value of error can be positive or negative depending on the position of the actual data point to estimated datapoint. If the actual data point is below the predicted data point, the value of the error is negative. On the other hand, if the real data point is above the predicted data point, the value error is positive. Least square method sums the square of the individual error at every data point and finds the best fit to minimize error. The total error from the four data points in Figure 17 can be calculated by (5).

$$err = ((y_1 - f(x_1))^2 + ((y_2 - f(x_2))^2 + ((y_3 - f(x_3))^2 + ((y_4 - f(x_4))^2 \quad (5)$$

The overall error can be calculated by (6):

$$err = \sum_{i=1}^{n} ((y_i - f(x_i))^2 \quad (6)$$

Here, n is the total number of datasets. For the linear fit, (6) can be written as (7):

$$err = \sum_{i=1}^{n} ((y_i - f(ax_i + b))^2 \quad (7)$$

In this thesis, an exponential fit has used to analyze datasets. For exponential curve fit, (6) can be expressed as (8):

$$err = \sum_{i=1}^{n} ((y_i - f(ae^{bt}))^2 \quad (8)$$

Using the least squares regression method, the coefficients a and b can be calculated. These coefficients define the exponential curve which most closely passes through the given data points (x_i,y_i) for $i = 1,2,3,4, \dots \dots \dots n$.

3.3.2 Feature Scaling

Although leak location can be determined by using exponential curve fitting, leak size cannot be predicted by this method. Machine learning algorithms need to be applied to overcome the problem. However, before that, data is required to be scaled or normalized. Feature scaling commonly known as normalization is the process of reducing and even eliminating data redundancy and increasing the coherence of individual types. In this thesis, the raw data acquired from our test bench system varies widely. As the data varies widely, objective functions will not perform properly without normalization or scaling when it comes to machine learning. There are several types of feature sizing methods: min-max normalization, z-score normalization, and decimal scaling, etc. In this thesis, all these normalization techniques were applied and then compared to see which technique yields a better result.

3.3.2.1 Min-max normalization

In the min-max normalization approach, the data is scaled to a range typically between 0 to 1. The reason behind having this limited range is to obtain smaller standard deviations, which can subdue the effect of outliers [52]. The min-max scaling can be written as follows:

$$Z = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{9}$$

Where Z is the min-max normalized data, X represents the original data, X_{min} is the minimum value of X, and X_{max} indicates the maximum value of X.

3.3.2.2 Z-score normalization

Z-score normalization is the most commonly used normalization techniques used for feature scaling. It converts all the data to a standard scale with an average of zero and standard deviation of one [53]. The z score normalizing formula can be written as (10):

$$Z = \frac{x - \mu}{s} \tag{10}$$

Here, Z is the Z-score normalized data, x is the original data, μ denotes mean and s represents the standard deviation.

3.3.2.3 Decimal scaling

In this method, the decimal point in the data is moved using multiplication or division by a power of 10. The decimal scaling can be quantified as follows:

$$Z = X_i * 10^j \tag{11}$$

$$Z = \frac{X_i}{10^j} \tag{12}$$

Here, Z is the normalized data, x_i represents the original data, and j indicates the power of 10 where $j > 0$.

3.4 Data Classification Algorithm

Data classification can be defined as a process of organizing unstructured data into groups for its most methodological use. The purpose of data classification is to decide about the new category of data based on the knowledge received from the data analysis method. Classification is generally a supervised learning approach when it comes to machine learning or statistics. In this method, the machine learns from the input dataset and then utilizes this knowledge to categorize new observation. Examples of some classification problems are voice recognition, handwriting recognition, biometric identification, any document classification, etc. There are several types of classification algorithms in Machine Learning such as Linear Classifiers, Support Vector Machines (SVM), Decision Trees, Boosted Trees, Random Forest, Artificial Neural Networks (ANN), and Nearest Neighbor [54].

In this paper, leak and its location can be detected precisely using the exponential curve fitting method. However, leak size cannot be predicted using this method. Data classification can be a perfect solution to this problem. For this process, Support Vector Machine (SVM) & Multi-layer Perceptron (MLP) classifiers commonly known as Artificial Neural Network (ANN) have been applied that use a supervised learning algorithm to determine leak size. The primary goal is to develop a classifying model that can provide high efficiency in detecting leak sizes.

3.4.1 Multi-layer Perceptron (MLP) classifiers

Multi-layer Perceptron (MLP) classifier is commonly referred to as Artificial Neural Network (ANN). As the 'neural' part of their name suggests, they are brain-inspired methodologies which are intended to reproduce the way humans learn. They are outstanding tools for finding patterns which are way too complicated for a human to extract and instruct the machine to recognize [55]. Because of the faster computation speed and more efficient algorithms, ANN has been widely used in many fields, not only in engineering, science, and mathematics, but in banking, entertainment, and even literature [56].

3.4.1.1 Neuron Structure

The simplest neural network consists of one neuron which is called perceptron. The structure of a neuron is illustrated in Figure 18. A single neuron usually consists of five parts such as input (p), weight (w), bias (b), the transfer function (f), and output (a^1). The total number of features in the input data set represents the number of nodes in the input layer. Each element of input dataset is multiplied by a weight. Next, the results are summed together with a bias and passed through a transfer function. From Figure 18, input vector p is multiplied by the weight matrix w and then forwarded to the adder. Bias b is another element that is passed to the adder. The output of the adder is called net output is and then fed to the transfer function f that results in an output.

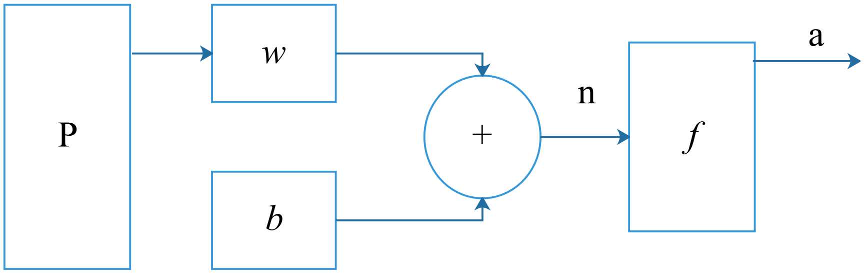

Figure 18. Simple neural network

The dimension of the input vector p is R × 1, where R is the number of features of input data. It is determined by the external statement of the problem. The dimension of the weight matrix w is S × R where S is the number of interested outputs. The dimension of the bias vector b and transfer function f is S × 1. Finally, the dimension of the output a is S × 1. The output can be written as follows (13):

$$a = f(wp + b) \tag{13}$$

3.4.1.2 Layers and Network

The typical structure of a three-layer neural network is shown in Figure 19. It can be noted that the superscripts do not mean the power but indicate the different layers.

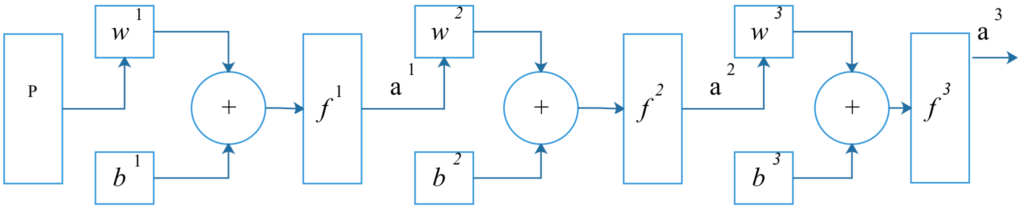

Figure 19. 3-layer neural network

The output of the first layer is a^1, which can be obtained in the following equation:

$$a^1 = f^1(w^1p + b^1) \tag{14}$$

Likewise, a^2 and a^3 can be calculated by (15) and (16) respectively:

$$a^2 = f^2(w^2p + b^2) \tag{15}$$

$$a^3 = f^3(w^3 p + b^3) \tag{16}$$

By substituting (14) and (15) into equation 16, the general output equation of the three-layer network can be written as follows:

$$a^3 = f^3(w^3 f^2(w^2 f^1(w^1 p + b^1) + b^2) + b^3) \tag{17}$$

Multilayer networks are more robust than single-layer systems because the users get more control over varying the functions with multilayer networks. However, it is not possible in the single-layer neuron networks [57].

3.4.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a set of supervised learning algorithms and is widely used for both regression and classification problems. However, in most cases, it is used in the classification problem. SVM classifier is also known as a discriminative classifier because it separates data points using a hyperplane commonly known as decision boundary. SVM can have several decision boundaries that accurately divide the data points. Unlike other classification algorithms, SVM selects the decision boundary which has maximal margin from the nearest points of all the classes. SVM doesn't merely find a decision boundary; it sees the most optimal decision boundary. The most optimal decision boundary is the one which has maximum margin from the nearest points of all the classes. These points are called support vectors as shown in Figure 20. In the case of support vector machine, the optimal decision boundary is often referred to as the maximum margin classifier or hyperplane.

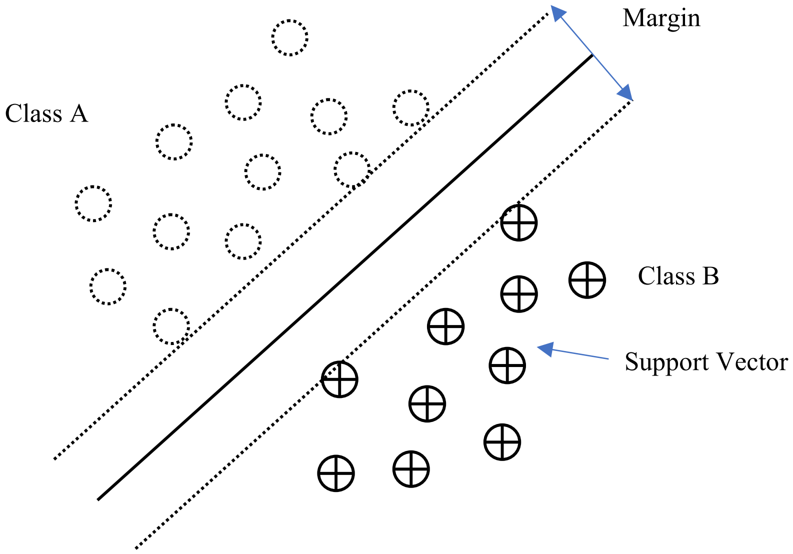

Figure 20. Simple SVM classifier

Support vectors identification is one of the biggest challenges which involves a complex mathematical algorithm. The algorithm computes the best margin between the optimal decision boundary and support vectors, thus maximizing the margin. Finding the decision boundary for linearly separable data is very easy and can be done using simple SVM algorithm. Nevertheless, when it comes to non-linearly separable data, a simple algorithm cannot be used because a single straight line cannot separate a non-linear dataset precisely. In this case, a remodeled version of SVM is used, commonly referred to as kernel SVM or kernel trick of SVM [58]. There are several kernel methods available: linear kernel, polynomial kernel, gaussian kernel, exponential kernel, laplacian kernel, sigmoid kernel, circular kernel, etc. In this thesis, the linear kernel, polynomial kernel, gaussian kernel, and sigmoid kernel were used.

3.4.2.1 Linear Kernel

The Linear kernel is the most straightforward kernel function. It works well when there are many features because mapping the data to a higher dimensional space does not always improve the performance. Also, the linear kernel is faster than with another kernel because it has less parameter to optimize. The linear kernel function is given below:

$$(x, y) = x^T * y + c \tag{18}$$

Here, $k(x, y)$ is a linear kernel where $< x, y >$ is the dataset, c is the regularization parameter and the only parameter that need to be optimized for SVM linear trick. The kernel measures the distance between new data and the support vectors [59].

3.4.2.2 Polynomial Kernel

The polynomial kernel is an example of the non-stationary kernel. When the training data is normalized, polynomial kernel works very well. The polynomial kernel function is given below:

$$k(x, y) = (\alpha x^T * y + c)^d \tag{19}$$

Here, the adjustable parameters are the slope alpha (α), the constant term c, and the polynomial degree. The classification algorithm always specifies the polynomial degree. Polynomial kernel behaves like a linear kernel when its value equal to one.

3.4.2.3 Gaussian Kernel

The gaussian function is also known as a radial basis function, is a popular kernel tricks used in machine learning especially. The function can be expressed as follows:

$$k(x, y) = e^{(-\frac{||x-y||^2}{2\sigma^2})} \tag{20}$$

Also, the function can be written as follows:

$$k(x, y) = e^{(-\gamma||x-y||^2)} \tag{21}$$

The adjustable parameter sigma (σ) or gamma (γ) should be carefully optimized as it plays a crucial role in the execution of the kernel. When the data is overestimated, the higher-dimensional projection loses its non-linear power, and the exponential function behaves like a linear function. On the contrary, if the data is underestimated, the feature lacks systematization, and the decision boundary becomes more and more sensitive to noise.

3.4.2.4 Hyperbolic Tangent (Sigmoid) Kernel

The hyperbolic tangent kernel is often referred to as the sigmoid kernel. It is also known as the multilayer perception kernel as it is derived from the neural network. The function can be expressed as follows:

$$k(x, y) = \tanh(\alpha x^T * y + c) \tag{22}$$

Two adjustable parameters of the sigmoid kernel are: (1) the slope alpha (α) and (2) the intercept constant c. A common value for α can be expressed as $1/N$, where N is the dimension of the data. As this kernel originally comes from neural network theory, it is very popular for support vector machine as well [60].

3.4.2.5 Confusion Matrix

A confusion matrix is a method to evaluate the performance of a classification algorithm on a set of test data for which the actual values are known. It is a summary of the prediction results of a classifier. The number of correct and incorrect predictions of a classifier are recapitulated with count values and broken down by each class [61]. Table 4 shows a simple structure of a confusion matrix.

Table 4. Confusion matrix

Class	Class 1 Predicted	Class 2 Predicted
Class 1 Actual	True Positive (TP)	False Negative (FN)
Class 2 Actual	False Positive (FP)	True Negative (TN)

This example has two classes, i.e., class 1 and class 2. Let us consider, class 1 denotes some activity positive and class 2 indicates the opposite, i.e., negative. The definition of the terms is given below:

True Positive (TP) - Observation is class 1 and is predicted to be class 1.

False Negative (FN) - Observation is class 1 but is predicted class 2.

True Negative (TN) - Observation is class 2 and is predicted is class 2.

False Positive (FP) - Observation is class 2 but is predicted class 1.

Classification Accuracy is given by the following relation:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{23}$$

However, the drawback with accuracy is that it assumes equal costs for both kinds of errors. As a result, very high efficiency can be better, moderate, or poor depending on the particular problem. This problem can be solved by using recall. Recall can be explained as the ratio of the total number of correctly classified positive examples divided by the total number of positive samples. The high value of recall indicates the class is accurately predicted. The following relation expresses the recall:

$$Recall = \frac{TP}{TP+FN} \tag{24}$$

In this thesis, the recall has been calculated for the SVM kernel to evaluate the classification performance.

4. EXPERIMENTAL ANALYSIS

A laboratory-based test bench system has been designed and developed to detect the leak, predict leak size and make the model effective and efficient. This system consists of U-shaped pipelines made of PVC pipes and is shown in Figure 21. Water was moved around in the system by a common water pump capable of providing up to 15 PSI of pressure. Three leaks were created in the pipe sections using hose bibb in three different locations. The first leak was created between sensor 1 & 2, the second leak was between sensor 3 & 4, and the third leak was between sensor 5 & 6. Sensor 1 & 2 are 180cm apart, sensor 3 & 4 are 100cm apart, sensor 5 & 6 are 180cm apart, sensor 2 & 5 are 500cm apart and Sensor 1 & 6 are 860cm apart. Leak-1 is located halfway between sensor 1 & 2, leak-2 is located halfway between sensor 3 & 4, and leak-3 is located halfway between sensor 5 & 6. The overall system is shown in Figure 21.

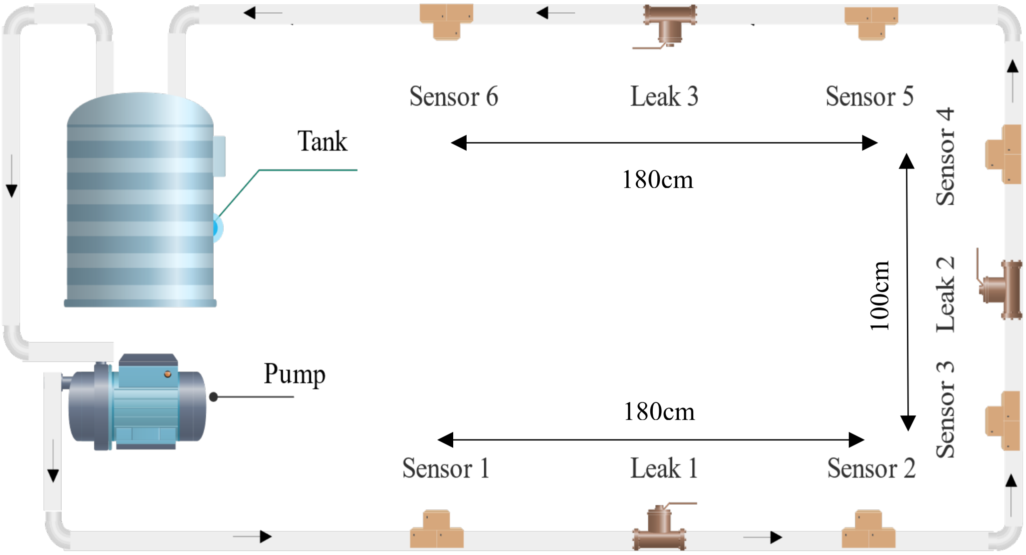

Figure 21. Overall system

A large number of datasets have been collected to verify the system. We used PVC pipes with three different diameters: 0.75-inch, 1-inch, and 1.5-inch; three leak locations and five different leak sizes: 0.5-inch, 0.4-inch, 0.3-inch, 0.2-inch, and 0.1-inch to collect data with different conditions, as illustrated in Figure 22.

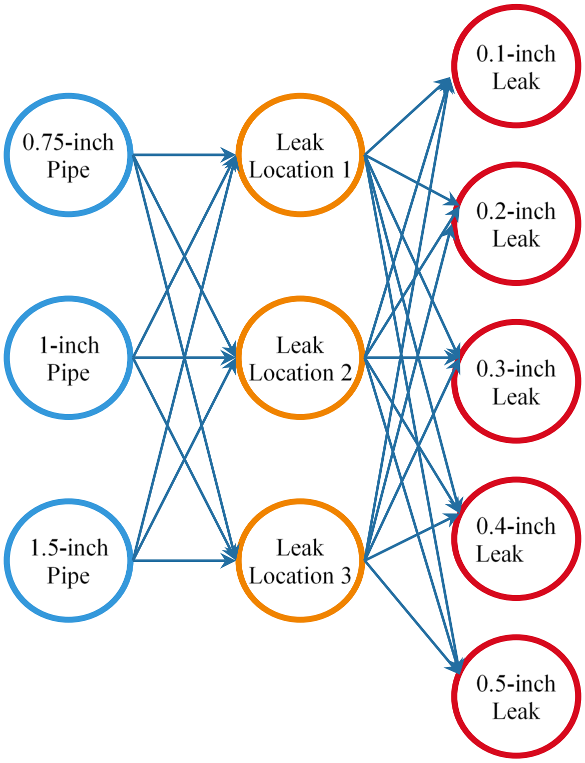

Figure 22. Data collection conditions

Figure 22 indicates that, for every pipeline, data was collected from three different locations by creating three leaks with five types of leak diameter variation. A total of 1180 datasets were obtained, and 900 leak datasets were separated from non-leak datasets. Next, the 900 leak data sets were analyzed further to detect leak location. The whole leak location identification procedure is given in Figure 23.

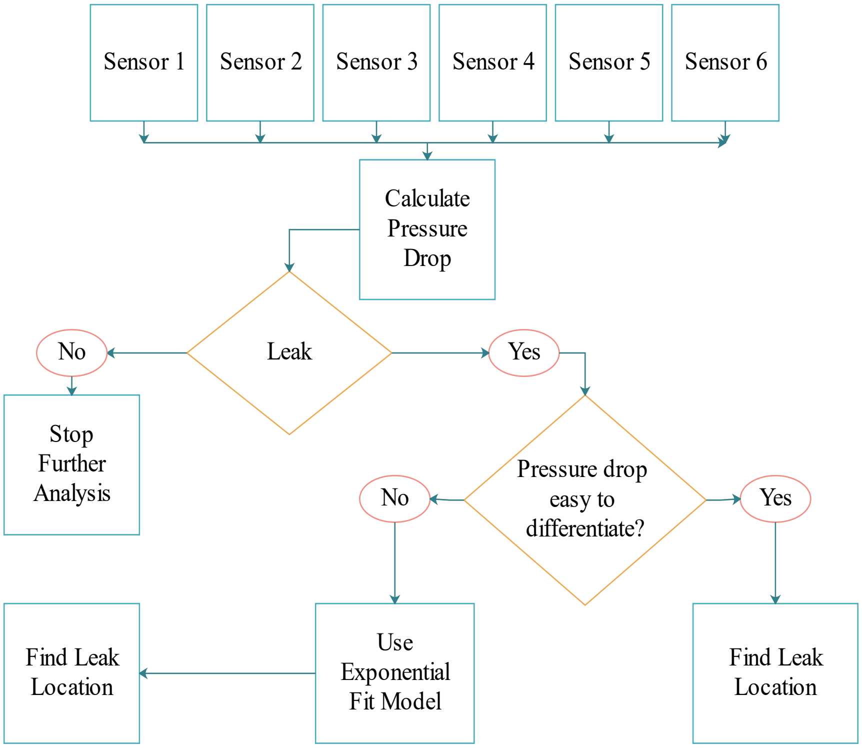

Figure 23. Overall leak location identification methodology

The whole experimental analysis can be divided into two sections: pressure drop analysis and exponential curve fit study.

4.1 Pressure Drop Analysis

4.1.1 Leak between sensor 1 & 2 with leak size of 0.5-inch

The following Figure 24 is drawn to explain pressure drop theory. Here, we had only two sensors 860cm apart. The other four sensors were not considered this time to explain the idea of calculating pressure drop. A leak in size of 0.5-inch was created using hose bibb near sensor-1. Sensor-1 was located before the leak, and Sensor-2 was situated after the leak. The hole was 90cm away from sensor-1 and 750cm from sensor-2.

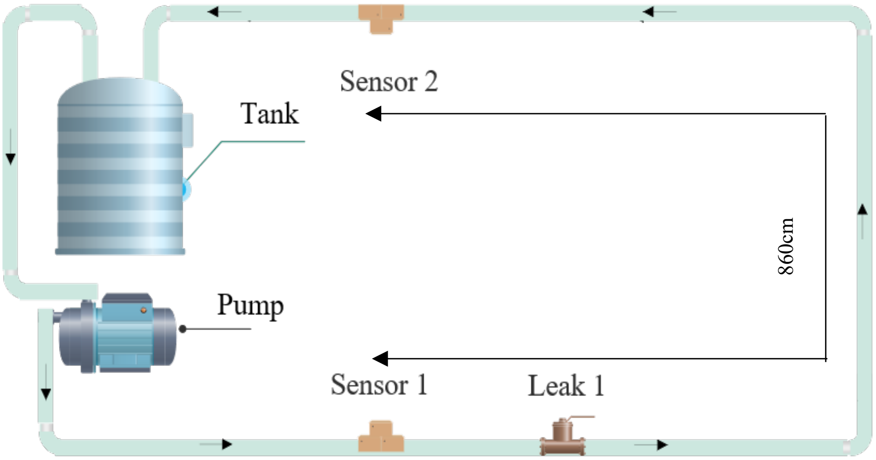

Figure 24. A system with one leak with size 0.5-inch and 2 sensors

Data from the pressure sensors were recorded using Arduino at 500ms intervals and transmitted through Zigbee. Raw data from the sensors is illustrated in Figure 25.

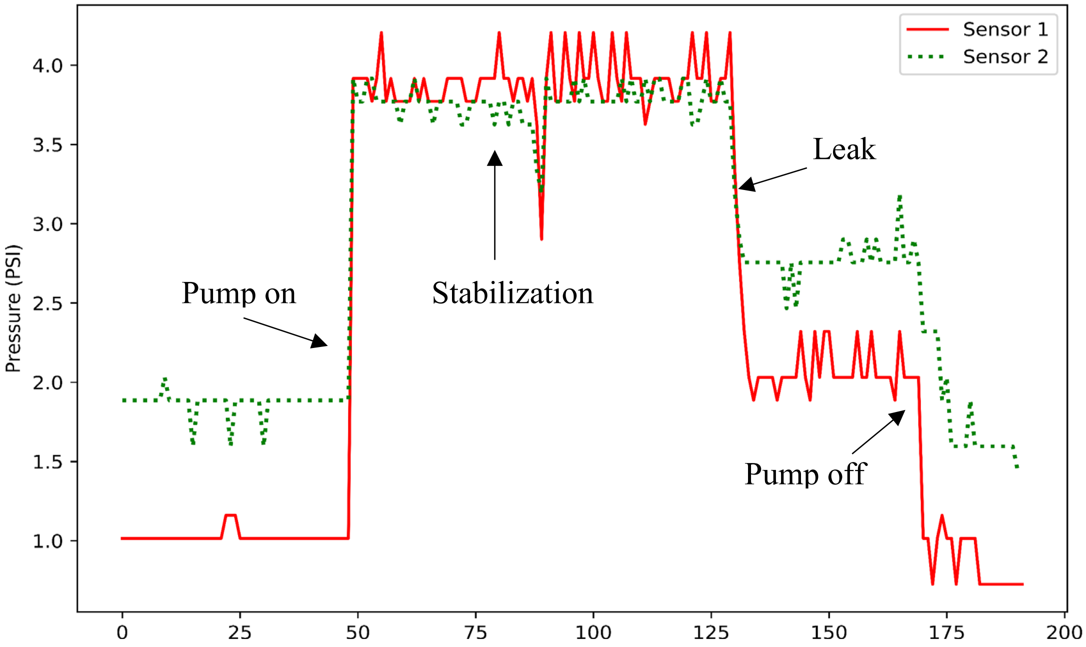

Figure 25. Raw data of two sensors with a 0.5-inch leak between sensor 1 & 2

This process can be divided into four main phases: the pumps start, stabilization, leak, and pump-off are visible from the output of the pressure sensors. Figure 25 includes the pressure before the pump was switched on, the increase in force when the water pump was switched on, the reduction in pressure value due to the development of the leak and finally the drop-in pressure due to the switching off the pump shortly after the occurrence of the hole.

As the leak was created close to sensor-1, it should show more downstream pressure profile than sensor-2. However, based on Figure 25, it is difficult to determine the exact position of the leak because both pressure sensors appear to respond similarly to the leak. A close-range pressure profile of these two sensors is shown in Figure 26.

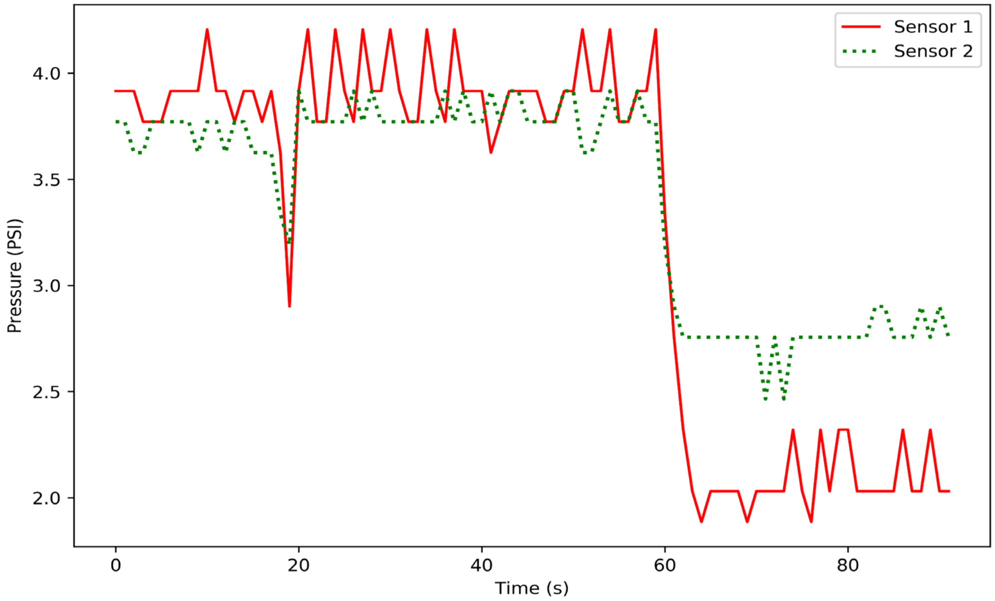

Figure 26. Close-range pressure profile of 2 sensors with 0.5-inch leak

It can be noticed from Figure 26 that the pressure profile of sensors 1 and 2 are somewhat different. The pressure profile of sensor 1 drops more sharply than sensor 2. This indication can be used to determine the approximate location of the leak, i.e., it is somewhere near sensor 1 in this case. However, it is also challenging to pinpoint leak location by looking at downstream pressure drop especially when the leak size is small. To overcome this problem pressure drop has been calculated to pinpoint leak location is shown in Figure 27. The average pressure value of the ‘stabilization’ part (Figure 25, p. 48) of every sensor was calculated and subtracted from every pressure datapoint of an individual pressure sensor to determine pressure drop. According to the hypothesis, sensor 1 should exhibit more pressure drop than sensor 2 as it is close to the leak.

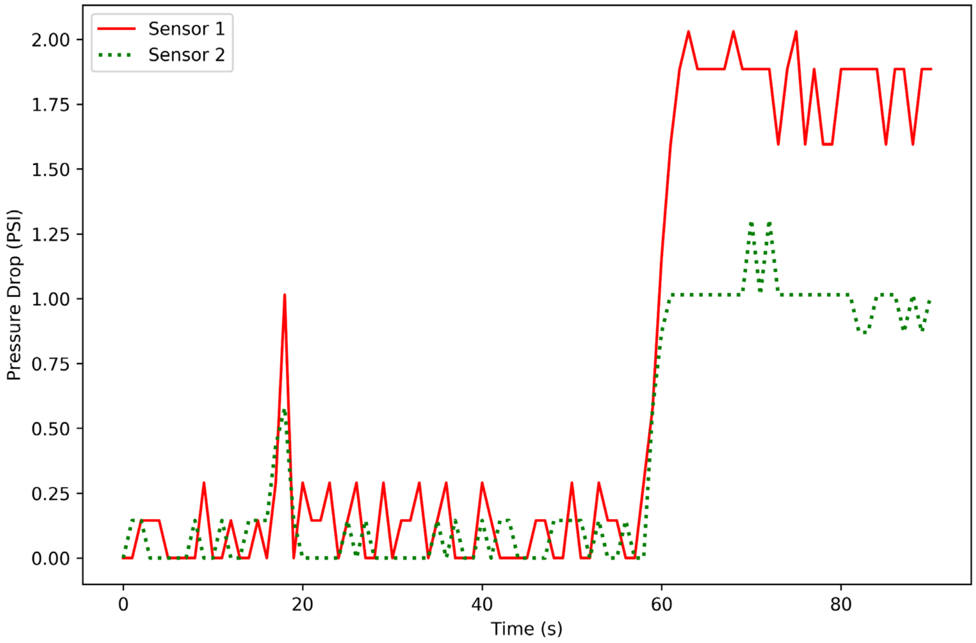

Figure 27. Pressure drop of 2 sensors with a 0.5-inch leak between sensor 1 & 2

It is easily understandable from Figure 27 that sensor 1 exhibits more pressure drop than sensor 2. So, we can assume that the leak is located somewhere near sensor 1. This idea can be explained more precisely when more sensors are implemented. Figure 21 (Chapter 4, p. 45) shows six pressure sensors attached to the PVC pipe sections. Leak 2 and leak 3 were disregarded in this case which means the value of these two leaks system was closed. Sensors 1 was located before the hole with a diameter of 0.5-inch, and sensors 2, 3, 4, 5, & 6 were placed after the leak. Data from the pressure sensors were recorded at 500ms interval. Raw data from the sensors are illustrated in Figure 28.

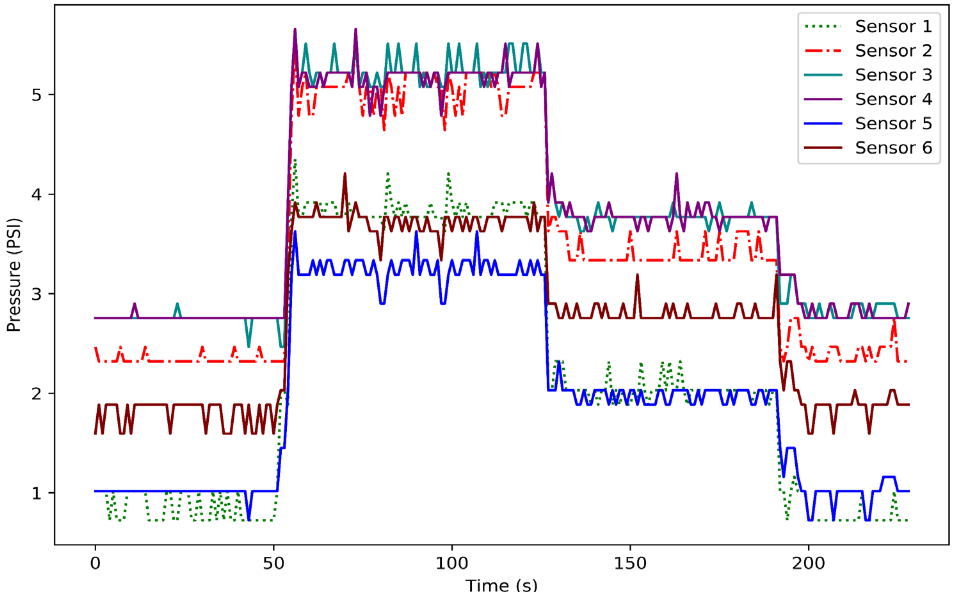

Figure 28. Raw data of six sensors with a 0.5-inch leak between sensor 1 & 2

Based on Figure 28, it is difficult to determine which sensors show more downstream profile. A close-range pattern of all pressure sensors could solve the problem and is shown in Figure 29. According to the hypothesis, sensor 1 and sensor 2 should exhibit more downstream pressure profile than other sensors because the leak is located between them. However, close-range profile analysis cannot provide useful insight about the downstream pattern of sensors. This is because the pressure value of all six sensors drops immediately after the leak. So, it is hard to understand the difference by looking at the pressure profile of all six sensors.

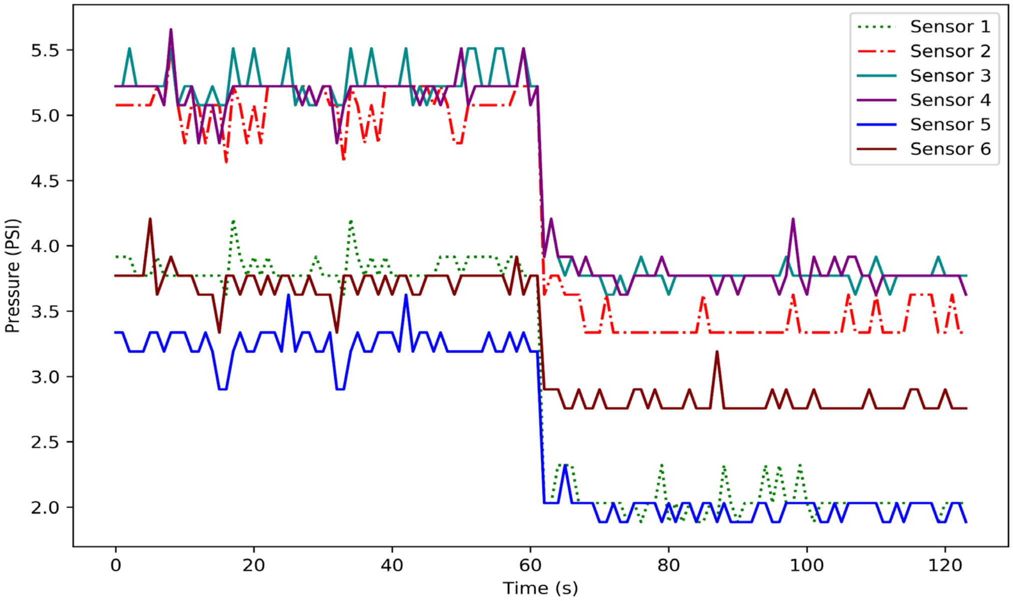

Figure 29. Close-range pressure profile of 6 sensors with 0.5-inch leak

As the close-range pressure profile analysis does not help much to identify leak location, pressure drop analysis can be a vital method to find leak location. Figure 30 presents a pressure drop analysis of 6 sensors with the 0.5-inch leak between sensor 1 & 2.

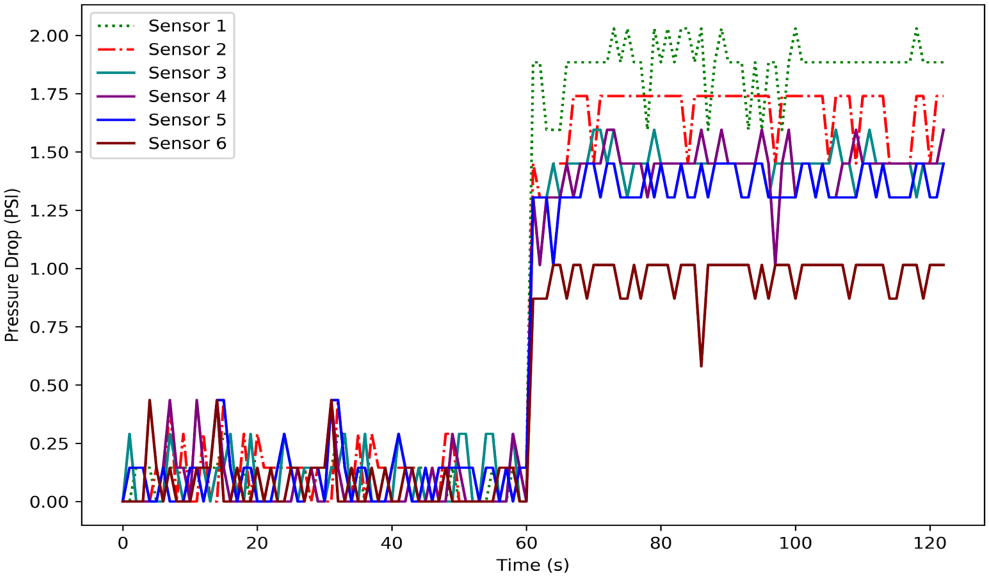

Figure 30. Pressure drop of 6 sensors with a 0.5-inch leak between sensor 1 & 2

It is clear from Figure 30 that sensor 1 & 2 exhibits more pressure drop than sensor 3, 4, 5, 6. So, it can be said that the leak is located between sensor 1 and 2.

4.1.2 Leak between sensor 3 & 4 with leak size of 0.5-inch

This time, the leak was created between sensors 3 & 4 as shown in Figure 21 (Chapter 4, p. 45). Leak 1 and leak 3 were disregarded in this case meaning the valves of these two leaks were closed. Sensors 1, 2, & 3 were located before the hole with a diameter of 0.5-inch, and sensors 4, 5, 6 were located after the leak. Data from the pressure sensors were recorded at 500ms intervals. Raw data sensor and close-range profile analysis have been omitted from this section because these analyses do not help to identify leak location. Figure 31 shows a pressure drop of six sensors when a leak is created between sensor 3 & 4. According to the hypothesis, sensor 3 and sensor 4 should exhibit more pressure drop than other sensors because the hole is located between them.

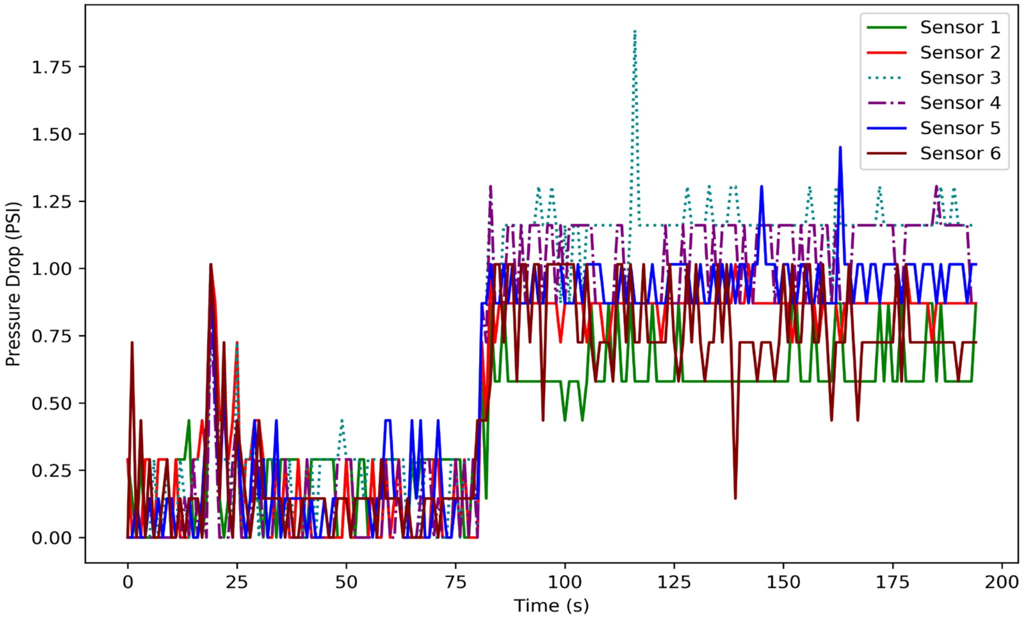

Figure 31. Pressure drop of 6 sensors with a 0.5-inch leak between sensor 3 & 4

It is clear from Figure 31 that sensor 3 & 4 exhibits more pressure drop than sensor 1, 2, 5, & 6. So, it can be said that the leak is located between sensor 3 and 4.

4.1.3 Leak between sensor 5 & 6 with leak size of 0.5-inch

This section follows the same method as the previous two sections. A leak was created between sensors 5 & 6 to justify proposed system accuracy with different leak location is shown in Figure 21 (Chapter 4, p. 45). Leak-1 and leak-2 were disregarded in this case meaning the valves of these two leaks system were closed. Sensors 1, 2, 3, 4, & 5 were located before the leak with a diameter of 0.5-inch, and sensor 6 was located after the leak. Data from the pressure sensors were recorded at 500ms intervals. According to our hypothesis, sensors 5 and 6 should exhibit more pressure drop than other sensors because the leak is located between them. Figure 32 depicts pressure drop analysis when a hole is created between sensors 5 & 6.

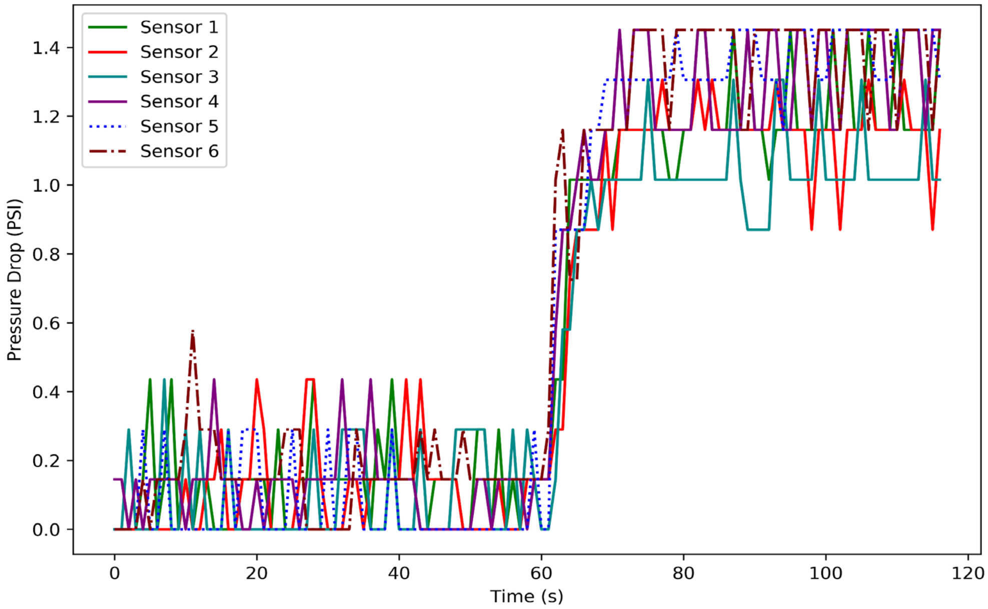

Figure 32. Pressure drop of 6 sensors with a 0.5-inch leak between sensor 5 & 6.

It is clear from Figure 32 that sensors 5 & 6 exhibit more pressure drop than sensors 1, 2, 3, 4. So, it can be said that the leak is located between sensor 5 and 6.

So, leak location can be identified by using simple pressure drop analysis. However, when the leak size is small, the pressure drops are significantly less for every sensor. Therefore, it becomes difficult to determine leak location by calculating pressure drop for each sensor. This phenomenon is explained in the next section.

4.1.4 Other sizes of a leak between sensors 1 and 2

Here, the leak was created between sensors 1 & 2 and the leak diameter was changed to 0.4-inch using hose bibb. Data from the pressure sensors were recorded at 500ms intervals, and the pressure drop was calculated. Figure 33 exhibits a pressure drop analysis with 0.4-inch leak sizes while the leak was created between sensor 1 & 2.

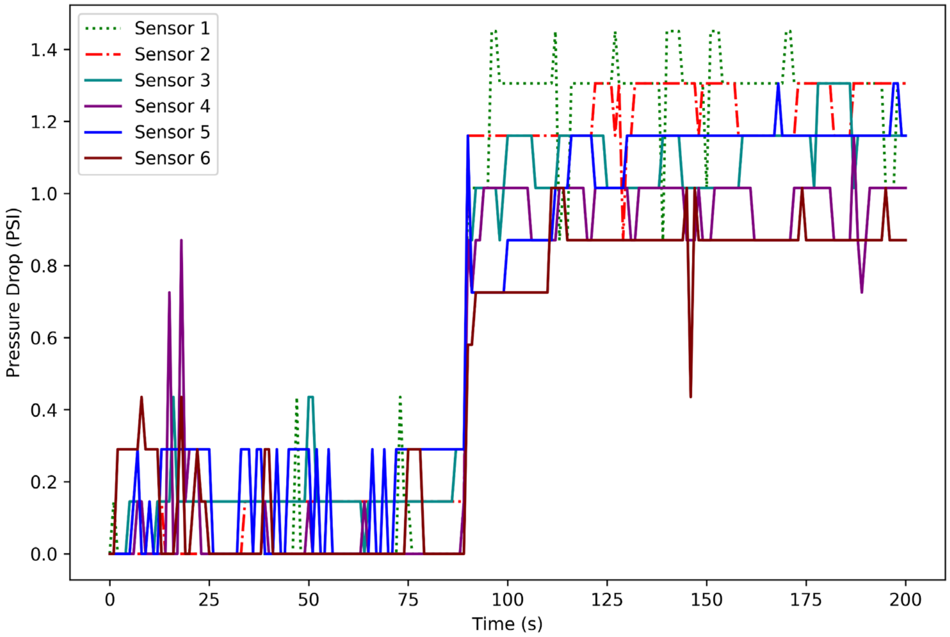

Figure 33. Pressure drop of 6 sensors with a 0.4-inch leak between sensor 1 & 2.

It is clear from Figure 33 that sensor 1 & 2 exhibit more pressure drop than sensors 3, 4, 5, 6. So, it can be said that the leak is located between sensor 1 and 2. Thus, the 0.4-inch leak can be detected by pressure drop analysis.

The leak size was further changed to 0.3-inch while keeping the same leak location, i.e., between sensor 1 & 2. Data from the pressure sensors were recorded at 500ms intervals, and the pressure drop was calculated. Figure 34 exhibits a pressure drop analysis with 0.3-inch leak sizes while the leak was created between sensor 1 & 2.

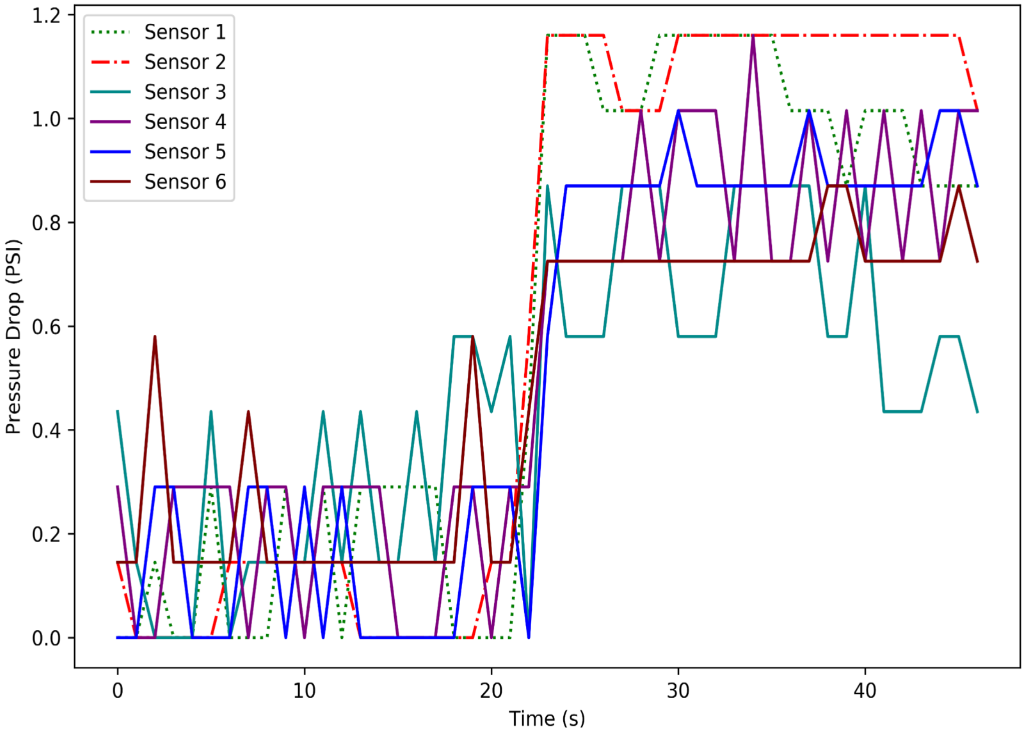

Figure 34. Pressure drop of 6 sensors with 0.3-inch leak between sensor 1 & 2

It is clear from Figure 34 that sensor 1 & 2 exhibits more pressure drop than sensor 3, 4, 5, 6. So, it can be said that the leak is located between sensor 1 and 2. Thus, the 0.3-inch leak can be detected by pressure drop analysis.

The analysis becomes more interesting with the decrease of leak size. Small leak produces less significant pressure drop, therefore, it is more difficult to distinguish pressure drops from sensors. Figure 35 exhibits a pressure drop analysis with 0.2-inch leak sizes while the leak was created between sensor 1 & 2. Data from the pressure sensors were recorded at 500ms intervals, and the pressure drop was calculated.

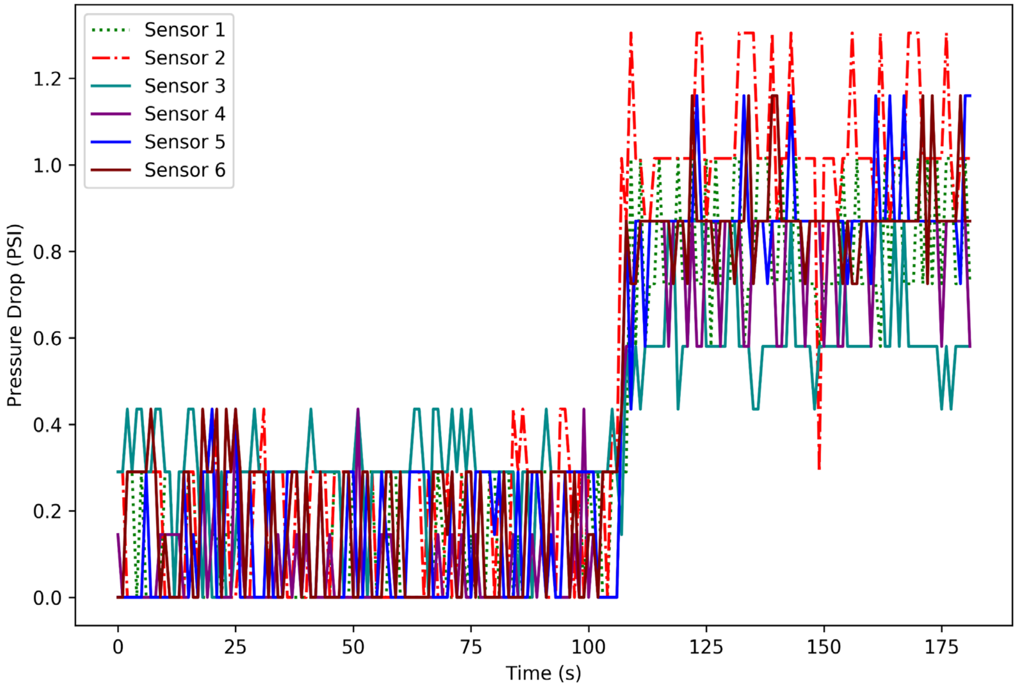

Figure 35. Pressure drop of 6 sensors with a 0.2-inch leak between sensor 1 & 2

It is not entirely clear from Figure 35 that which sensors exhibit more pressure drop. So, the leak location is getting difficult to identify when leak size is small. Figure 36 shows a pressure drop with 0.1-inch leak sizes while the leak was created between sensor 1 & 2. It is hard to interpret from the picture which sensors exhibit more pressure. So, the leak location cannot be detected by the pressure drop method especially when leak size is small.

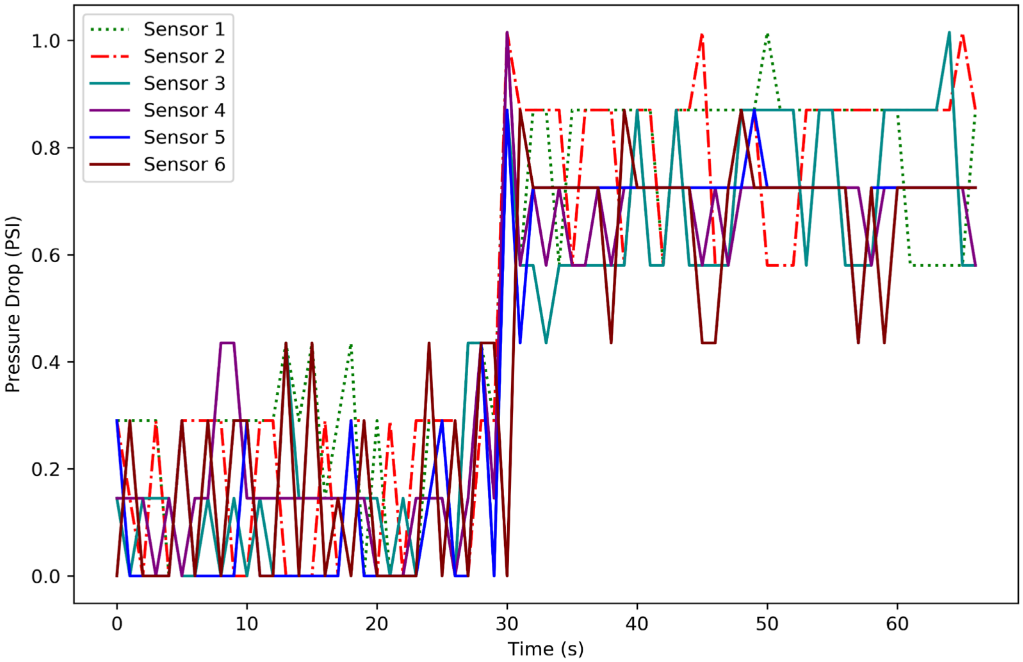

Figure 36. Pressure drop of 6 sensors with a 0.1-inch leak between sensor 1 & 2

4.2 Exponential Curve Fit Analysis

Many exercises in nature depend on exponential expression. For instance, the decrease of the temperature over time and the growth of the bacterial colony over time can be represented by the exponential relationship. In this thesis, the reduction in pressure value over time can also be shaped by this mathematical tool.

A leak of 0.5-inch was created between sensor 1 and 2. Sensor 1 was located before the hole, and sensors 2, 3, 4, 5, 6 were located after the leak. Data from the pressure sensors were recorded at 500ms intervals. Exponential curve fitting model was then applied using MATLAB as illustrated in Figure 37. The curve fit model was applied separately on each of the sensors.

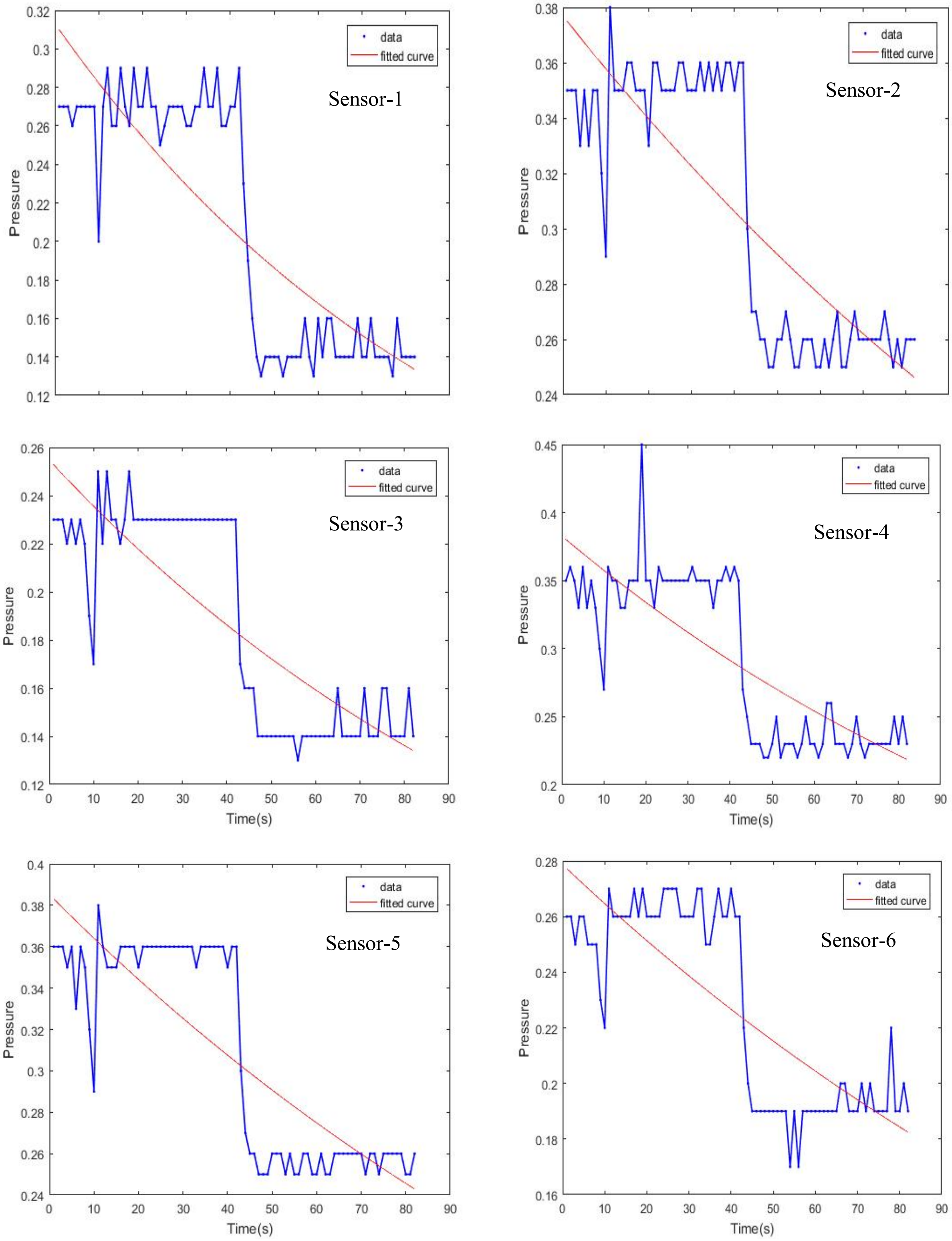

Figure 37. Curve fit analysis with a leak size 0.5-inch.

In Figure 37, an exponential curve fit was applied to six sensor's data that gives six separate decay rates. In other word, the six sensors produce six decay rates. The decay rate of six sensors is provided in Table 5.

Table 5. Decay rate for a leak size 0.5-inch

Sensor	1	2	3	4	5	6
Decay constant 'b'	0.0104	0.0078	0.0068	0.0056	0.0052	0.0050

It is clear from Table 5 that sensor 1 and 2 have a larger decay rate than other sensors. So, we can assume that the leak is located between sensor 1 and 2.

Now, the idea of comparing decay rates can be justified if it can detect a leak with a small diameter. In the next experiment, a hole of 0.1-inch was created between sensor 1 and 2. Sensor 1 was located before the leak, and sensors 2, 3, 4, 5, & 6 were located after the leak. Data from the pressure sensors were recorded at 500ms intervals. An exponential curve fitting model was then applied using MATLAB as illustrated in Figure 38. The curve fit model was applied separately on each of the sensors. The decay rate of 6 sensors for the 0.1-inch leak is given in Table 6.

Table 6. Decay rate for leak size 0.5-inch

Sensor	1	2	3	4	5	6
Decay constant 'b'	0.001000	0.0009013	0.0009000	0.0008040	0.0007011	0.0006001

It is clear from Table 6 that the sensor 1 and 2 have more decay rate than other sensors. More significant digits have been calculated to distinguish every sensor's decay rate precisely.

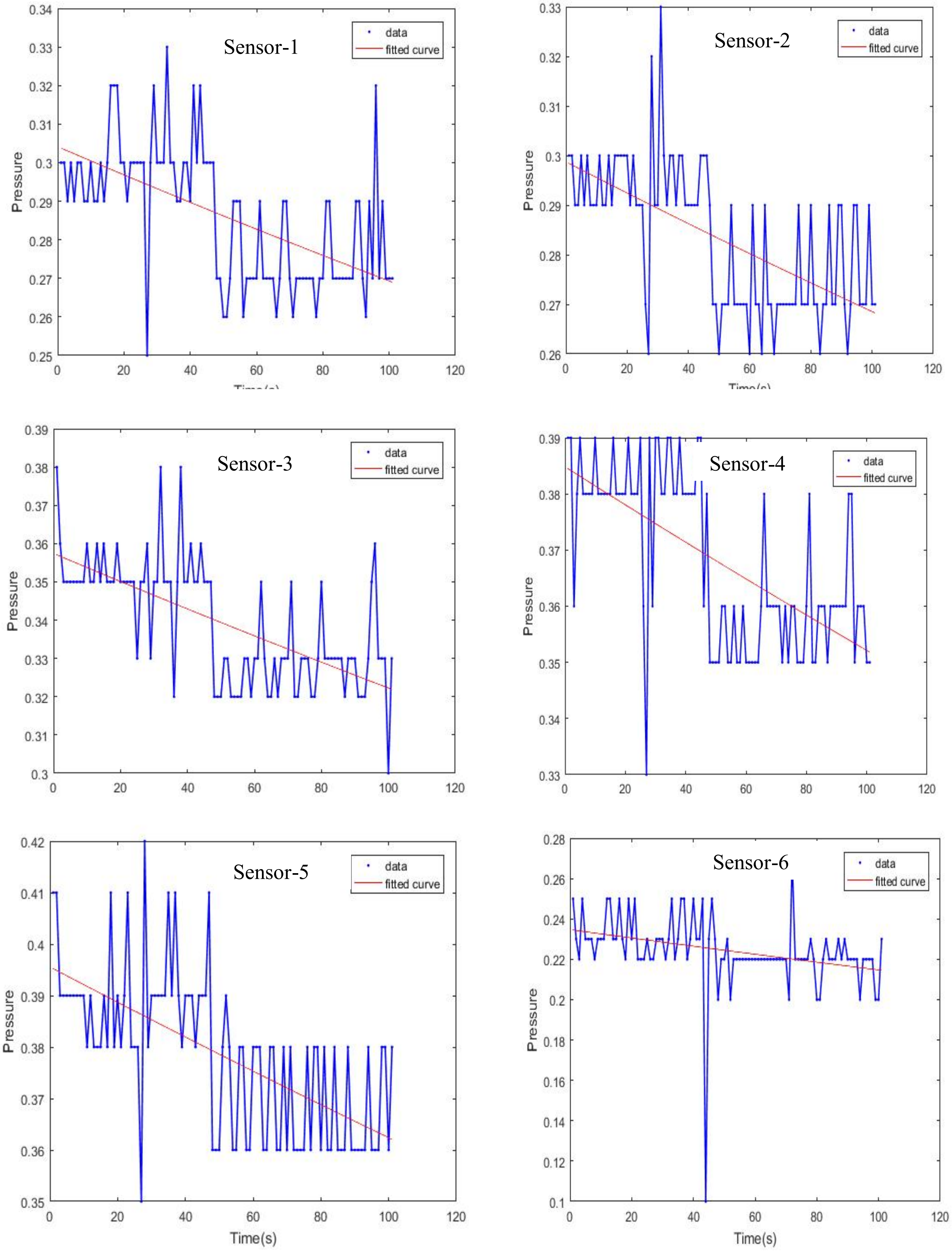

Figure 38. Curve fit analysis with a leak size 0.1-inch

Although decay constant of sensors 2 & 3 is very close, a comparison between them can be used to differentiate them more precisely. For example, sensor 2 value is 14.44% bigger than sensor 3. So, we can now claim that the leak is located between sensor 1 and 2.

So, the trend in a dataset can be captured using curve fitting by allocating a single function across the entire range. In this thesis, decay constant 'b' was calculated using the curve fitting method. Here, b is a negative constant which can be called the decay rate that denotes how rapidly the pressure value decreases with time. The magnitude of b had been taken during curve fit analysis to make data analysis more manageable. In this experiment, one of the essential parts of data analysis is to separate datasets that contain a leak from datasets that do not include the leak. The exponential curve fitting model was able to distinguish between these two types of data sets. To illustrate the exponential curve fit in a non-leak data set, data was taken continuously from all six sensors. Data from the pressure sensors were recorded at 500ms intervals. The exponential curve fitting model was then applied using MATLAB. Curve fitting on only one sensor for the non-data set is shown in following Figure 39. The decay rate of 6 sensors for the non-leak dataset is given in Table 7.

Table 7. 'b' value for non-leak dataset

Sensor	1	2	3	4	5	6
Constant 'b'	0.0003	0.0002	0.0002	0.0001	0.0001	0.0001

It is noticed from Table 7 that, the value of 'b' is close to zero for the non-leak dataset. However, it is not possible to distinguish leak and non-leak data set by only looking at the magnitude of ‘b’ value. Instead of taking the magnitude, the actual value of ‘b’ needs to be considered to separate the two types of datasets.

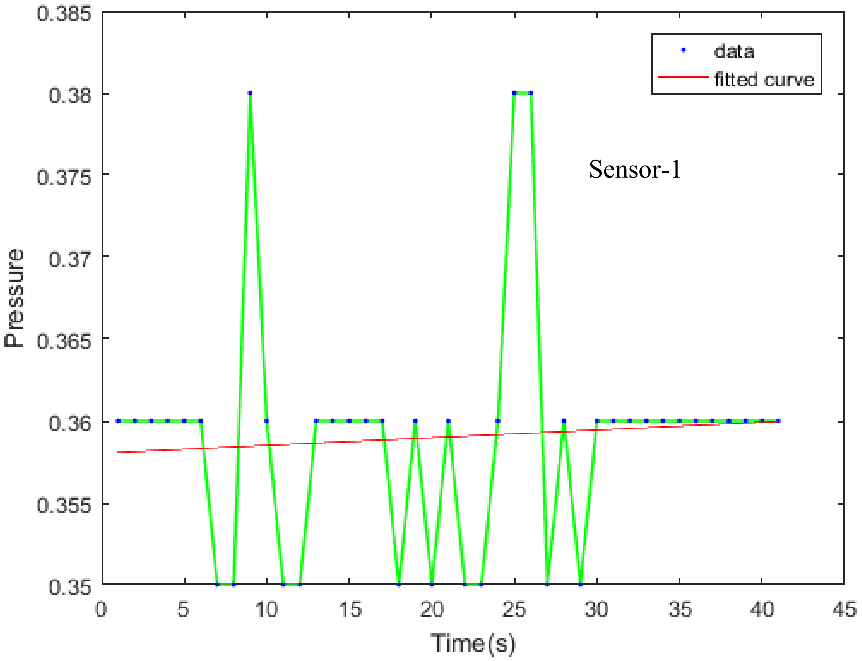

Figure 39. Curve fit analysis for non-leak data

The actual value of b is always negative for the leak dataset because data goes down immediately after the leak. However, in the case of the non-leak dataset, the actual b value is positive. It was previously mentioned that, for this experiment, the magnitude of ‘b’ was considered for the ease of analysis. However, to separate leak datasets from the non-leak datasets, the actual value of b must be considered. Datasets that contain leaks, provide negative b value and non-leak datasets usually give positive value. A few examples of b values for leak and non-leak datasets are shown in Table 8.

Table 8. ‘b’ value for non-leak dataset

No	Sensor-1	Sensor-2	Sensor-3	Sensor-4	Sensor-5	Sensor-6	Type
Actual ‘b’ value	-0.0104	-0.0078	-0.0068	-0.0056	-0.0052	-0.0050	Leak
	-0.0083	-0.0060	-0.0052	-0.0050	-0.0048	-0.0046	Leak
	-0.0032	-0.0025	-0.0023	-0.0020	-0.0018	-0.0015	Leak
	0.0003	0.0002	0.0002	0.0001	0.0001	0.0001	Non-Leak
	0.0001	0.0002	0.0001	0.0001	0.0002	0.0002	Non-Leak
	0.0003	0.0003	0.0002	0.0002	-0.0001	0.0001	Non-Leak

So, the actual value of b is always negative for the leak dataset because pressure goes down immediately after the leak. However, in the case of the non-leak dataset, the actual b value is positive for most of the trials and close to zero for all trials because data remains almost the same during the stabilization state. By this way, a leak dataset can be separated from a non-leak dataset using the curve fitting method for further analysis.

Although leak location can be determined using an exponential curve fit, leak size cannot be predicted by this method. However, this can be done using classification techniques such as Support Vector Machine (SVM) and Artificial Neural Network (ANN).

The features need to be extracted from the dataset to do classification. Exponential curve fitting method gives six decay rates for six sensors. These six types of decay constant were considered as six features. Furthermore, as pipes with three different diameters have been considered for the system, one extra element, i.e., pipe diameter needs to be added for the classification of data to determine leak size. So, a total of seven items were used as feature vectors for the classification. An example of feature vectors is given the Table 9.

Table 9. Example of feature values

Sensor-1	Sensor-2	Sensor-3	Sensor-4	Sensor-5	Sensor-6	Pipe Diameter	Leak Diameter
0.09	0.08	0.08	0.08	0.07	0.06	0.75	1
0.16	0.15	0.15	0.15	0.14	0.14	0.75	2
0.24	0.18	0.18	0.17	0.15	0.14	0.75	3
0.38	0.29	0.27	0.25	0.22	0.2	0.75	4
0.77	0.6	0.51	0.43	0.4	0.38	0.75	5
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
0.04	0.04	0.03	0.02	0.01	0.01	1	1
0.1	0.11	0.11	0.1	0.09	0.08	1	2
0.15	0.15	0.15	0.14	0.13	0.12	1	3
0.23	0.22	0.21	0.22	0.2	0.19	1	4
0.35	0.3	0.31	0.29	0.29	0.28	1	5
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
0.11	0.1	0.1	0.08	0.08	0.07	1.5	5
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…

First seven features were considered as prediction variables by which data was trained for SVM and ANN. The last column was considered as a response variable which is the outcome of the classifications. The outcome of predicting leak sizes using SVM and ANN will be analyzed in Chapter 5.

5. RESULTS

In this thesis, data have been collected from the test-bench system with several conditions such as: (1) Three kinds of pipe diameters: 0.75-inch, 1-inch, and 1.5-inch; (2) 3 leak locations; and (3) Five leak sizes: 0.5-inch, 0.4-inch, 0.3-inch, 0.2-inch, and 0.1-inch. The data collection system is illustrated in Figure 40.

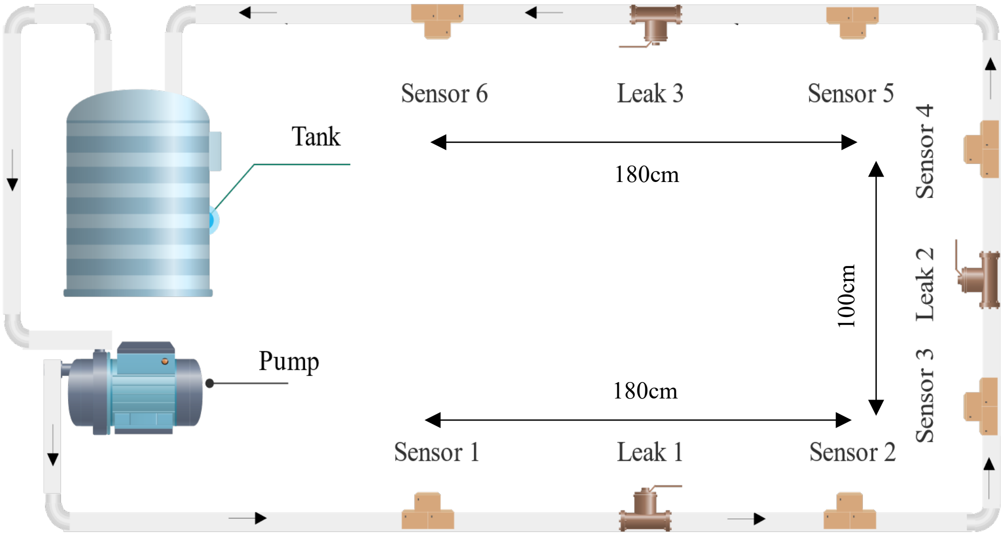

Figure 40. Data collection system.

A total of 1180 sets of data has been collected including datasets that have a leak and datasets that do not have a leak as well. Data were stored and then analyzed to separate leak data sets from non-leak data sets. Next, a total of 900 sets of leak data were analyzed to localize leak and predict leak size. In this chapter, the result of the data analysis is discussed.

5.1 Separating leak and non-leak dataset

The exponential curve fitting model was used to separate leak data sets from non-leak data sets. The accuracy of the separation of leak data sets from non-leak data sets is shown in Table 10.

Table 10. Accuracy to separate leak and non-leak data

Data sets	Test datasets	Correctly predicted	Accuracy
Leak data sets	100	98	98%
Non-leak data sets	100	100	100%

We examined 200 datasets containing all types of leak size using an exponential curve fitting method to determine accuracy. The analysis was performed to evaluate the effectiveness of the technique. The result shows an exponential curve fit model does a wonderful task in detecting a leak.

5.2 Measuring the right distance between sensors

The accuracy was calculated using 900 leak data sets with three different leak locations, i.e., leak 1 which is between sensor 1 & 2, leak 2 which is between sensor 3 & 4, and leak 3 which is between sensors 5 & 6. Results are shown in Table 11.

Table 11. Overall Accuracy Rate

Leak Location	1	2	3
Overall Accuracy	82.5%	52.5%	79.1%

The data in Table 11 indicates that when leaks were created between sensors 1 & 2 and sensors 5 & 6, our model gave better accuracy for leak location identification compared to when leaks were formed between sensor 3 & 4. The distance between sensor 1 & 2 and 5 & 6 is identical, hence providing almost similar accuracy. However, the distance between sensors 3 & 4 is less compared to the distance between other sensors mentioned above and the model is less accurate.

The distance between sensors 3 & 4 is 100cm. For any leaks between these sensors, the model shows significantly less efficiency in detecting leak location. So, it can be concluded that 100cm is not the right distance between sensors. To verify this hypothesis, the effect of sensor 3 and 4 has been nullified and data were collected only from sensors 1, 2, 5, & 6. Now the leak is between sensor 2 & 5 and both sensors are 500cm away is shown in Figure 41.

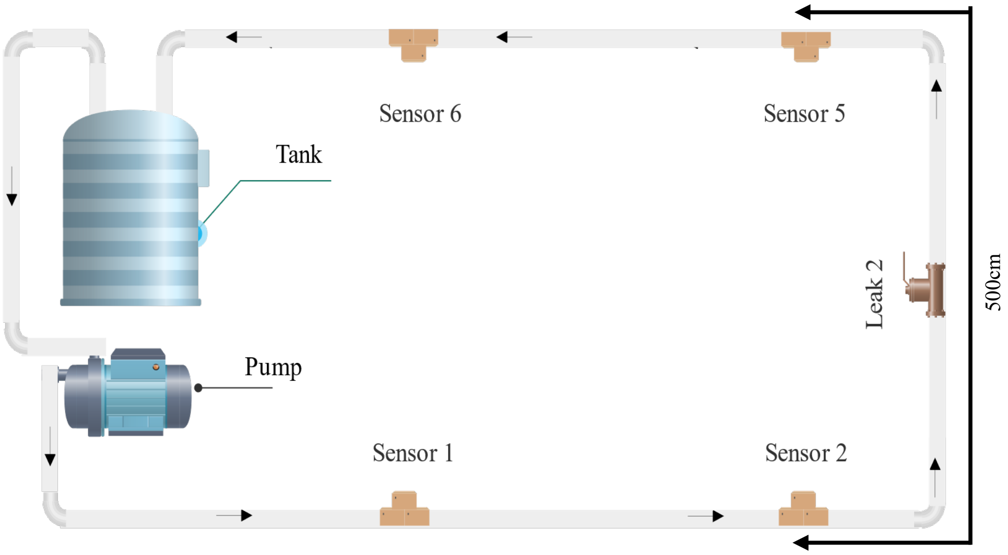

Figure 41. Leak between sensor 2 & 5

A total of 125 datasets have been collected and then the efficiency of leak location identification has been calculated. In this case, the accuracy of 85.6% is achieved. So, the distance between sensors plays a pivotal role in detecting leak location accurately. Thus, it can be concluded that distance between sensors should be 180cm or more to get an acceptable efficiency in identifying leak location.

5.3 Accuracy with different leak sizes

As data sets related to leak-2 showed less efficiency because of the incorrect distance between sensors, so datasets associated with the leak-1 and leak-3 were considered only. A total of 750 datasets had been collected with different conditions. Leak location identification accuracy has been calculated separately for pipelines with three different diameters. Table 12 represents leak location identification accuracy for the pipeline with a 1.5-inch diameter.

Table 12. Accuracy Rate for 1.5-inch pipe

Leak Size	Total Dataset	Correctly detected	Incorrectly detected	Accuracy
0.1-inch leak	50	38	12	76%
0.2-inch Leak	50	36	14	72%
0.3-inch Leak	50	40	10	80%
0.4-inch Leak	50	41	9	82%
0.5-inch Leak	50	43	7	86%
Overall	250	198	52	79.20%

Table 12 indicates that the leak location identification accuracy increases with the increased leak sizes. A large leak in the pipeline creates more pressure drop inside the pipe, thus giving high decay constants ‘b’ which are easily differentiable compared to smaller leak size. In this case, the only exception is the dataset related to 0.2-inch leak where it produces less accuracy than dataset contains the 0.1-inch hole. The overall efficiency achieved to detect leak location for pipelines with 1.5-inch diameter is 79.20%. Next, leak location identification accuracy for 1-inch pipe diameter is shown in Table 13.

Table 13. Accuracy Rate for 1-inch pipe

Leak Size	Total Dataset	Correctly detected	Incorrectly detected	Accuracy
0.1-inch leak	50	38	12	76%
0.2-inch Leak	50	38	12	76%
0.3-inch Leak	50	40	10	80%
0.4-inch Leak	50	42	8	84%
0.5-inch Leak	50	43	7	86%
Overall	250	201	49	80.4%

It is clear from the Table 13 that leak location identification accuracy increases with the increased leak sizes. The highest efficiency is observed as 86% when leak size is 0.5-inch. The overall efficiency achieved to detect leak location for pipelines with 1-inch diameter is 80.4%. Leak location identification accuracy for 0.75-inch pipe diameter is shown in Table 14.

Table 14. Efficiency Rate for 0.75-inch pipe

Leak Size	Total Dataset	Correctly detected	Incorrectly detected	Accuracy
0.1-inch leak	50	39	11	78%
0.2-inch Leak	50	40	10	80%
0.3-inch Leak	50	41	9	82%
0.4-inch Leak	50	43	7	86%
0.5-inch Leak	50	44	6	88%
Overall	250	207	43	82.8%

Table 14 indicates that leak location identification accuracy increases with the increased leak sizes. The highest accuracy is observed as 88% when leak size is 0.5-inch. The overall accuracy achieved to detect leak location for pipelines with 0.75-inch diameter is 82.8%.

5.4 Overall leak location identification accuracy

All the accuracy comparison between 5 leak sizes is listed in the following Table 15.

Table 15. Accuracy comparison

Leak Size	1.5-inch pipe	1-inch pipe	0.75-inch pipe
Overall	79.2%	80.4%	82.8%
0.1-inch leak	76%	76%	78%
0.2-inch Leak	72%	76%	80%
0.3-inch Leak	80%	80%	82%
0.4-inch Leak	82%	84%	86%
0.5-inch Leak	86%	86%	88%

This comparison can be easily understood using the bar chart is shown in Figure 42. It is clear from the figure that leak detection accuracy increases with the decreased pipe diameter. Generally, the smaller diameter pipe creates more water pressure inside it when the water flow rate remains the same. In another words, the accuracy increases with the increased pressure inside the tube. For instances, in this thesis, the 0.75-inch pipe has the highest overall accuracy because it has more water pressure inside it.

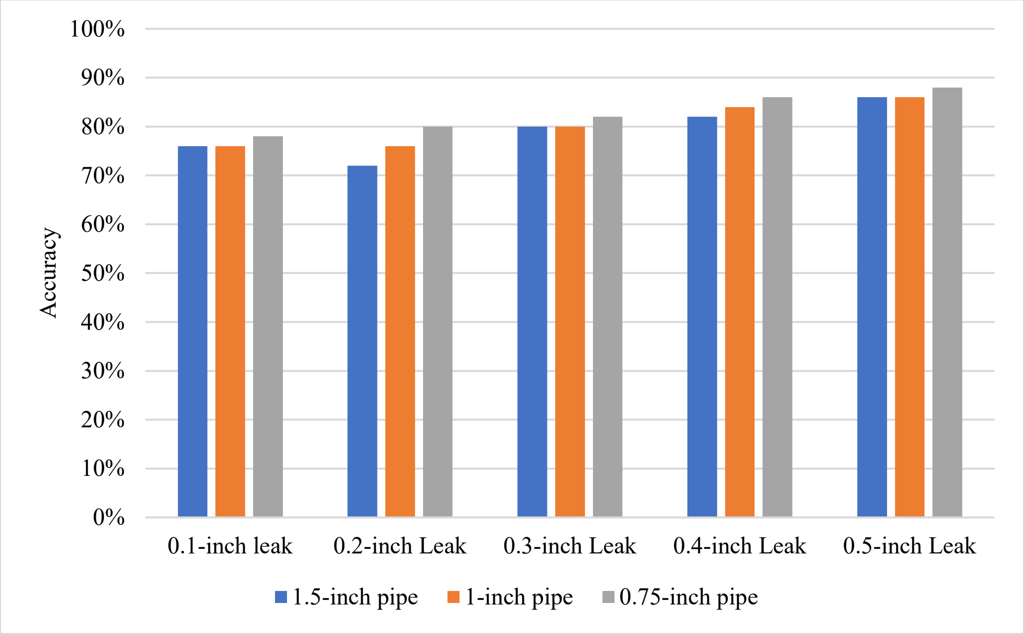

Figure 42. Accuracy comparison chart

5.5 Leak size prediction accuracy using SVM

SVM and MLP neural network algorithm were used to predict leak size. Figure 43 shows the procedure involving SVM and MLP classification. SVM and MLP are two popular strategies for data classification. Both methods are efficient depending on the project type. Data need to be trained and tested for both classification techniques. In this thesis, 600 and 129 datasets were used to train and test the classification systems respectively.

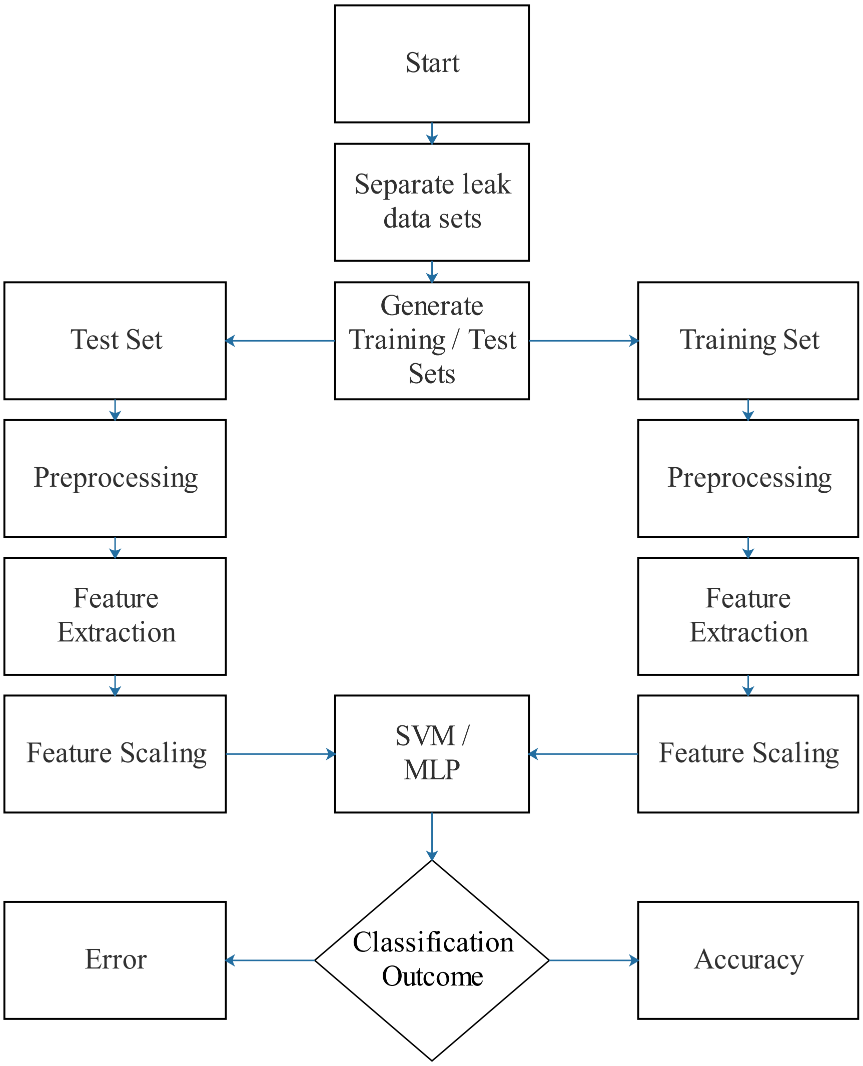

Figure 43. Data classification methodology

In the case of SVM, different types of SVM kernel tricks such as linear, polynomial, Gaussian and sigmoid are used to predict the leak size using Python's Scikit-Learn library. The accuracy of predicting leak size using a linear, polynomial, Gaussian and sigmoid kernel with different normalization technique is calculated separately and shown in Table 16.

Table 16. Classification accuracy using SVM

Normalization	SVM Kernel	Accuracy
Min-max normalization	Linear	40.31%
	Sigmoid	37.99%
	Polynomial	39.53%
	Gaussian	39.5%
z-score normalization	Linear	51.16%
	Sigmoid	31.01%
	Polynomial	30.11%
	Gaussian	31%
Decimal scaling	Linear	90.69 %
	Sigmoid	65.89 %
	Polynomial	62.01%
	Gaussian	77.52%

Table 16 indicates that, in the case of the min-max normalization method, all the SVM kernels show almost the same accuracy. But when it comes to z-score normalization technique, linear SVM kernel shows better accuracy than other kernel methods. And, in the case of decimal scaling, linear kernel trick shows outstanding accuracy. Figure 44 shows a comparison between different normalization techniques with different SVM kernel tricks. It is seen from this figure; linear kernel algorithm shows a higher accuracy over other algorithms for different normalizations. Also, decimal scaling normalization technique outperforms other normalization methods. The highest efficiency of 90.69% was obtained using decimal normalization technique and linear SVM kernel.

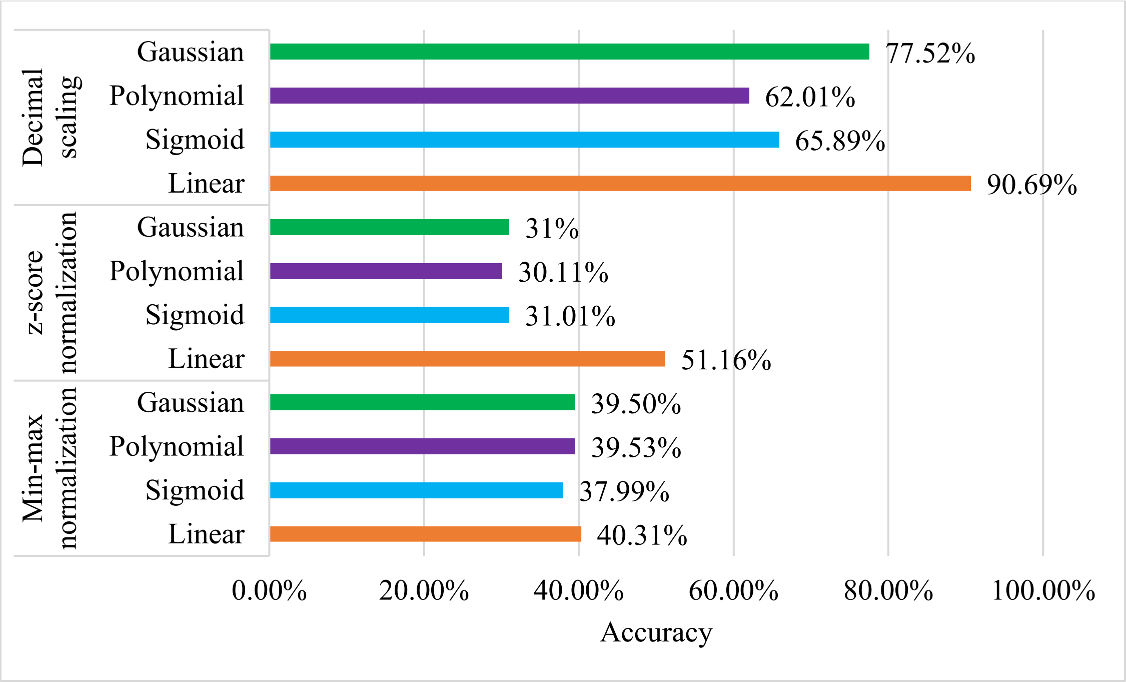

Figure 44. Accuracy comparison for SVM kernel

There is a reason behind the high accuracy of predicting leak size using decimal scaling. Table 9 (Chapter 4, p. 65) shows the data of the first seven columns is between 0 to 1.5. However, data of the first six columns are very distinct compared to data of the 7th column. In this condition, data was plotted closely to each other on a graph, and it was hard for SVM classifier to separate data by drawing a boundary. The data of the first six columns were then multiplied by 10^2 to make it easily distinguishable. Then SVM was then successful in drawing an optimum boundary to separate data which gave an output of 90.69%.

Further analysis was done by analyzing a confusion matrix. As the decimal scaling provides higher SVM accuracy, 129 test datasets scaled using decimal were considered for confusion matrix analysis. Figure 45 represents a confusion matrix.

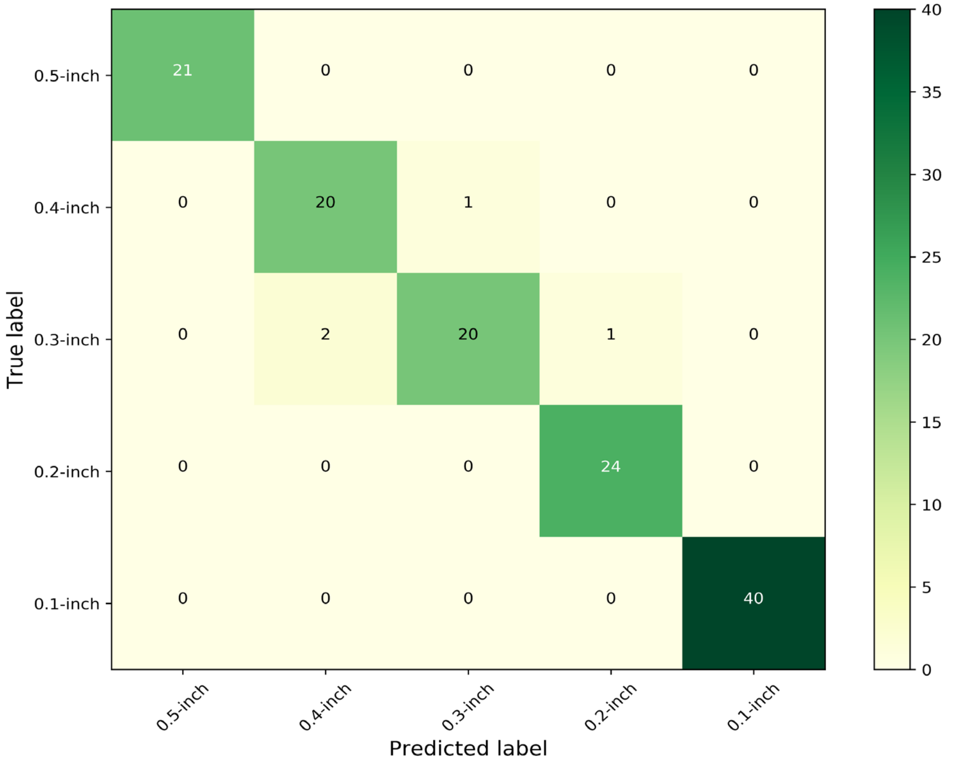

Figure 45. Confusion matrix

Figure 45 indicates that SVM classifies 0.5-inch, 0.4-inch, and 0.1-inch dataset precisely. However, there are some errors while predicting other classes of data. This idea can be better explained using Figure 46 where recall is calculated. It is clear from the figure that SVM does an excellent job in separating classes. The highest 'recall' obtained from the classification is 100%, and the lowest is 87%. The lowest recall is observed during the sorting of 0.3-inch class.

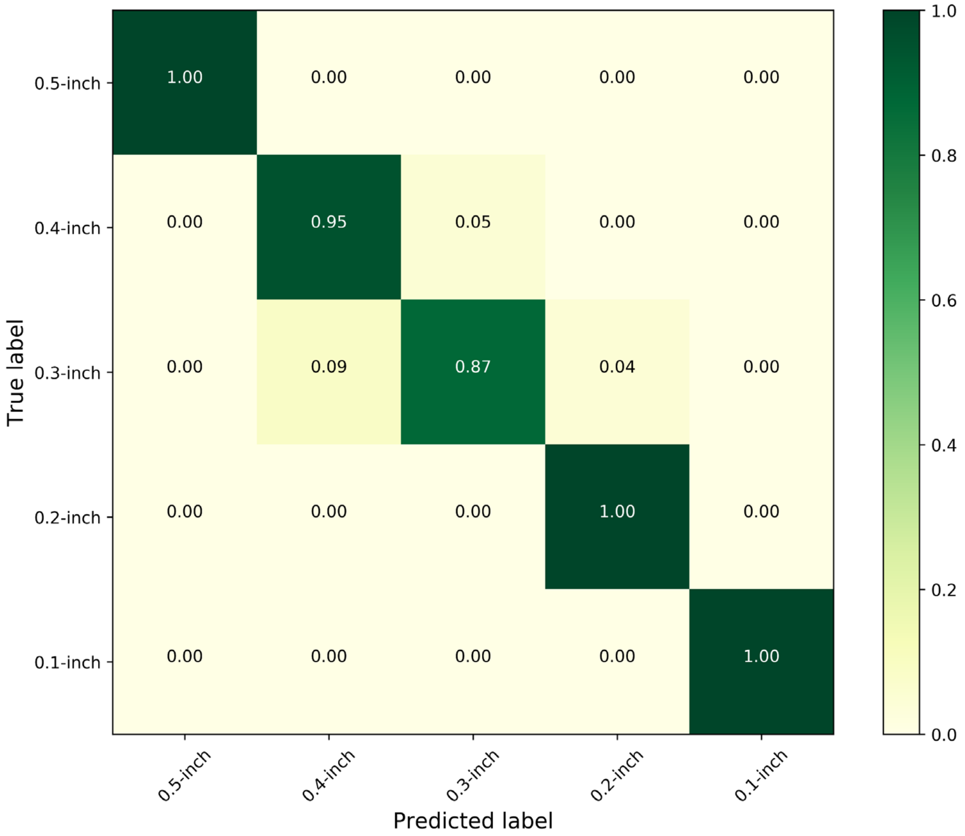

Figure 46. Recall in the confusion matrix

5.6 Leak size prediction accuracy using MLP neural network

Next, we developed an MLP neural network trained with 600 datasets and tested with 129 datasets. The layer number and hidden nodes have been varied to obtain maximum efficiency as shown in Table 17. The highest accuracy of 96.89% was obtained by a 3-layers neural network having 30, 10, and 10 nodes respectively. Decimal scaling was used to normalize the feature vectors to get this accuracy. As only three layers of MLP neural network gives the highest efficiency, the data training and processing become more straightforward and less time-consuming.

Table 17. Classification accuracy using SVM

Sl. No	Layer-1 Nodes	Layer 2 Nodes	Layer 3 Nodes	Layer 4 Nodes	Layer 5 Nodes	Accuracy
1.	10	-	-	-	-	92.24%
2.	20	10	-	-	-	93.79%
3.	10	20	-	-	-	91.47%
4.	20	30	10	-	-	95.34%
5.	30	10	10	-	-	96.89%
6.	20	40	30	10	-	96.12%
7.	20	30	10	40	-	94.57%
8.	20	10	40	30	50	92.84%
9.	20	30	10	40	20	92.37%

5.7 Comparison between SVM and MLP classifier

The overall accuracy in predicting leaks is calculated by MLP neural network and ANN using min-max normalization, and z-score normalization methods as shown in Table 18. The result indicates that MLP neural network is robust compared to SVM with the same number of training datasets and test datasets.

Table 18. Classification accuracy using SVM & MLP

Normalization	Classifier	Accuracy	Error
Min-max normalization	SVM	40.31%	59.69%
	MLP	63.12%	36.88%
z-score normalization	SVM	51.16%	48.84%
	MLP	39.52%	60.48%
Decimal scaling	SVM	90.69%	9.31%
	MLP	96.89%	3.88%
No Normalization	SVM	59.99%	40.01%
	MLP	95.22%	4.78%

Figure 47 shows a comparison between the MLP neural network and SVM using different normalization methods. It is seen that MLP and SVM were not successful in predicting leak size when trained and validated with min-max and z-score normalization techniques. However, when it comes to decimal scaling normalization technique, both technique SVM and MLP neural network performed very well with an accuracy of 90.69% an 96.89% respectively.

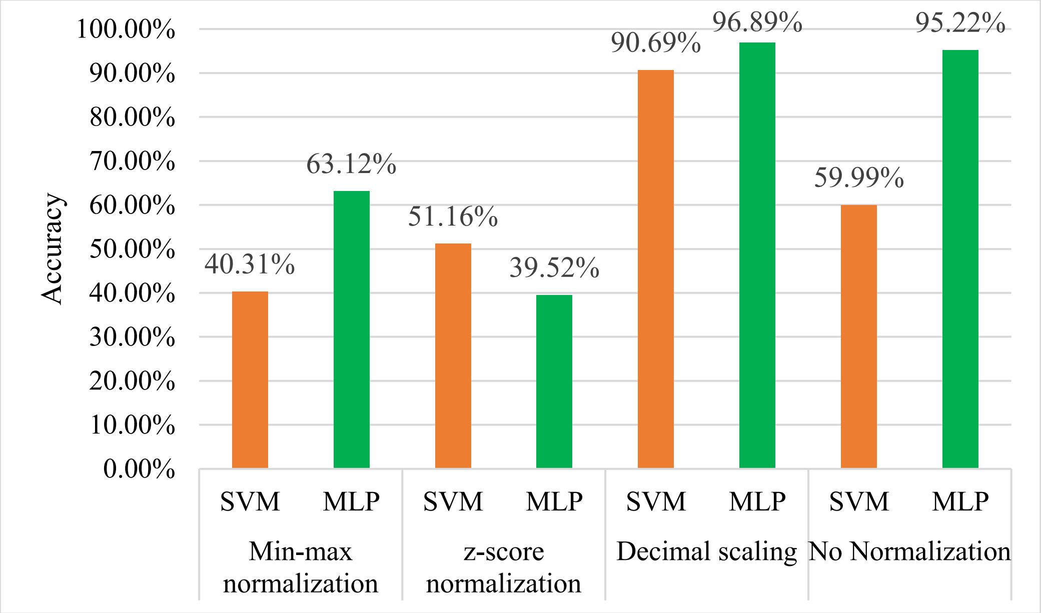

Figure 47. SVM vs. MLP

MLP neural network performed better in predicting leak size compared to SVM. Moreover, even with the absence of any normalization technique, MLP performed better than SVM by a significant margin. Thus, in this thesis, MLP neural network outperformed SVM in predicting leak size with the same number of training and testing datasets.

6. CONCLUSIONS

6.1 Discussion

Table 15 (Chapter 6, p. 71) shows 0.1-inch and 0.2-inch leak have produced less accuracy. In one case, 0.2-inch leakage exhibited less efficiency than the 0.1-inch leak. So, there are some inconsistencies in accuracy when the leak size is small. The probable reason for this mismatch is the internal pressure loss in the system.

Piping systems are aimed to fulfill fluid pressure at different junctions within an industrial application. So, the pressure loss must be considered while designing a pipeline system. The pressure loss is often viewed as friction loss because it is generated from frictional forces applied on a fluid in a pipeline system which resists the flow. Several factors that can cause friction loss such as gravity, fittings, valves, bends, joints [62]. The loss was calculated for our system using open source friction loss calculator [63]. Table 19 presents friction loss with different sensor position for 0.75-inch pipe diameter.

Table 19. Friction loss vs sensor positions

Sensor	Distance from source (feet)	No. of 90-degree elbows (left)	Pressure Loss (PSI)
Sensor 1	7	1	0.19
Sensor 2	12	1	0.3
Sensor 3	18	2	0.47
Sensor 4	22	2	0.56
Sensor 5	28	3	0.73
Sensor 6	34	3	0.87

Figure 48 shows average pressure and pressure loss of every sensors when the pump is on.

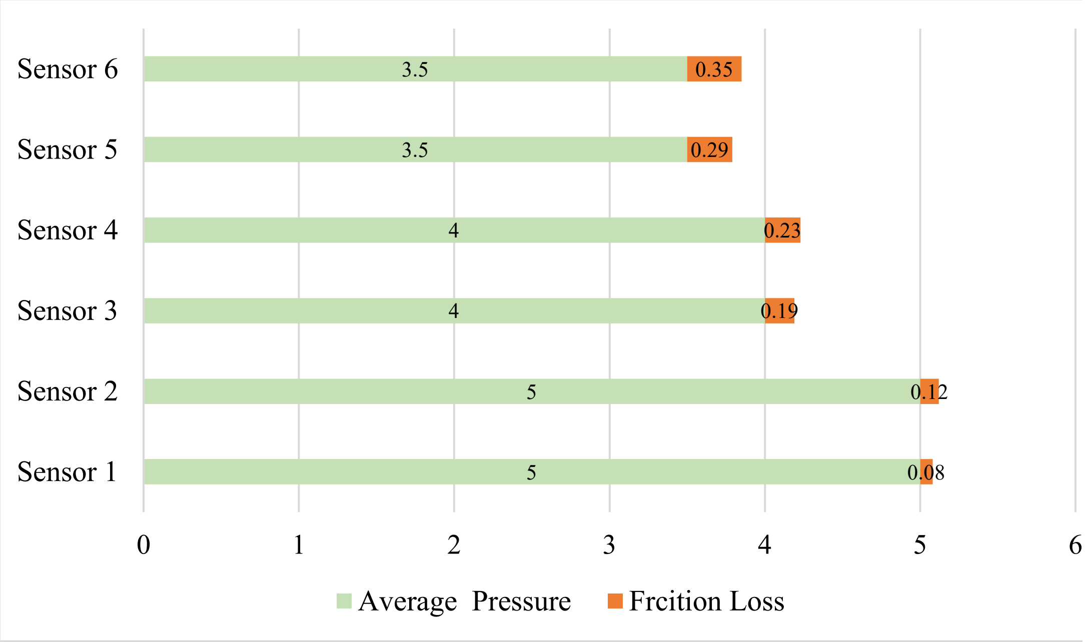

Figure 48. Friction loss with different sensors

It is clear from Figure 48 friction loss is responsible for some pressure losses. Figure 49 shows effect of friction loss in our system.

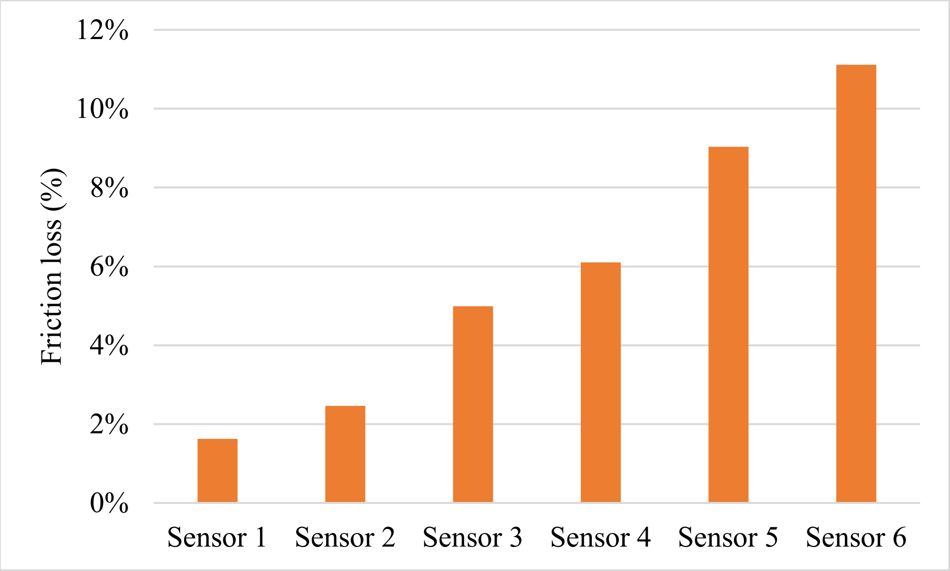

Figure 49. Friction loss effect

It is clear from Figure 49 that, friction loss increases with distance and number of valves in the pipeline system. Table 15 (Chapter 6, p. 71) shows the maximum leak location identification accuracy obtained was 88% when the leak size was 0.5-inch and minimum efficiency was 72% when leak size was 0.1-inch. Friction loss is one of the reasons why we have got low accuracy particularly when the leak size is small. Water pressure reduces with the friction loss. So, when the leak size is small, the friction loss makes it difficult for the pressure sensors to measure the subtle change in pressure, thus, leak localization accuracy reduced. Furthermore, Table 11 (Chapter 6, p. 67) shows, leak location 3 has less accuracy than leak location 1. It is because friction loss is more near at the end of the pipe which made the leak localization job difficult between sensor 5 & 6.

6.2 Summary

Having a reliable leak detection method in a pipeline distribution system is very important to prevent disaster. A laboratory-based test bench system has been developed to achieve the goal. Data were collected in different conditions using wireless sensor networks and then analyzed using programming languages, i.e., Python and MATLAB. A novel leak location identification method was then proposed and implemented. Next, data classification algorithms were used to predict leak sizes.

The leak location identification method was based on the exponential curve fitting method. Exponential curve fit method finds the decay constant of every pressure sensor. Comparison between decay rates of each sensor helped us to identify leak location. The system efficiency under some practical circumstances was then calculated.

The exponential curve fitting method cannot predict leak sizes. But data classification algorithms, i.e., SVM and MLP neural network performed very well in predicting leak sizes with an accuracy of 90.69% and 96.89% respectively. Researchers can use this method in leak detection analysis and, in other systems, wherever it best suits them.

6.3 Future works

Firstly, multiple leak locations cannot be detected using the proposed system. In future work, the network models can be extended considering numerous leaks in a network. Also, only PVC pipe was considered for this thesis. Therefore, the system can be examined using other types of pipe, i.e. galvanized steel and copper pipes in the future.

Additionally, only pressure sensors were used to detect leak location and its size in this system. However, system efficiency can be improved further by using multi-sensors. Such could be another future work. Furthermore, other applications that could get the advantage of our approach include detecting leaks and their location in oil and gas pipeline systems.

Moreover, in this paper, leak location can only be identified between the two sensors. However, the system cannot tell the exact spot between the two sensors. Pinpointing exact location can be another future work. Moreover, Raspberry Pi can be used instead of a computer to process and display data. Also, the mesh network can be developed using Zigbee to cover more area and get a better data transmission system and accuracy.

Finally, in the future, power consumption can be studied to improve the lifetime of sensor nodes.

APPENDIX SECTION

APPENDIX A. Arduino code for data collection from sensors

```
void setup () {
   Serial.begin (9600);
}
void loop () {
int sensorVal1 = analogRead(A1);
float voltage1 = (sensorVal1*5.0)/1024.0;
float pressure_pascal1 = (3.0*((float)voltage1-0.47)) *1000000.0;
float pressure_bar1 = pressure_pascal1/10e5;
       Serial.print("Pressure_Bar SNI: ");
       Serial.println(pressure_bar2);
int sensorVal2=analogRead(A2);
float voltage2 = (sensorVal4*5.0)/1024.0;
float pressure_pascal2 = (3.0*((float)voltage2-0.47)) *1000000.0;
float pressure_bar2 = pressure_pascal2/10e5;
       Serial.print("Pressure_Bar SNII: ");
       Serial.println(pressure_bar4);
int sensorVal3=analogRead(A3);
float voltage3 = (sensorVal3*5.0)/1024.0;
float pressure_pascal3 = (3.0*((float)voltage3-0.47)) *1000000.0;
float pressure_bar3 = pressure_pascal3/10e5;
```

```
        Serial.print("Pressure_Bar SNIII: ");
        Serial.println(pressure_bar3);
int sensorVal4=analogRead(A4);
float voltage4 = (sensorVal4*5.0)/1024.0;
float pressure_pascal4 = (3.0*((float)voltage4-0.47)) *1000000.0;
float pressure_bar4 = pressure_pascal4/10e5;
        Serial.print("Pressure_Bar SNIV: ");
        Serial.println(pressure_bar6);
int sensorVal5=analogRead(A5);
float voltage5 = (sensorVal5*5.0)/1024.0;
float pressure_pascal5 = (3.0*((float)voltage7-0.47)) *1000000.0;
float pressure_bar5 = pressure_pascal5/10e5;
        Serial.print("Pressure_Bar SNV: ");
        Serial.println(pressure_bar5);
int sensorVal6=analogRead(A6);
float voltage6 = (sensorVal6*5.0)/1024.0;
float pressure_pascal6 = (3.0*((float)voltage6-0.47)) *1000000.0;
float pressure_bar6 = pressure_pascal6/10e5;
        Serial.print("Pressure_Bar SNVI: ");
        Serial.println(pressure_bar6);
delay(500);
}
```

APPENDIX B. Python code for preprocessing raw data

```
'''**********************************************'''
import os
print ("Initially Working Directory -->", os.getcwd() + "\n")
os.chdir("")
'''**********************************************'''
import pandas as pd
x1 = pd.ExcelFile('')
x1.sheet_names
data_frame_raw = x1.parse('')
'''**********************************************'''
import re
def find_word(text, search
  result = re.findall('\\b'+search+'\\b', text, flags=re.IGNORECASE)
  if len(result)>0:
    return True
  else:
    return False
'''**********************************************'''
p_sni = []; p_snii = []; p_sniii = []; p_sniv = []; p_snv = []; p_snvi =
data_frame = data_frame_raw[0:len(data_frame_raw)]
for i in range(len(data_frame)):
   str = data_frame["Data"][i]
```

```
    if find_word(str, "Pressure_Bar"):
        if find_word(str, "sni"):
            count = str.find(":")+2
            p_sni.append(float(str[count:len(str)]))
        elif find_word(str, "snii"):
            count = str.find(":")+2
            p_snii.append(float(str[count:len(str)])
        elif find_word(str, "sniii"):
            count = str.find(":")+2
            p_sniii.append(float(str[count:len(str)]))
        elif find_word(str, "sniv"):
            count = str.find(":")+2
            p_sniv.append(float(str[count:len(str)]))
        elif find_word(str, "snv"):
            count = str.find(":")+2
            p_snv.append(float(str[count:len(str)]))
        elif find_word(str, "snvi"):
            count = str.find(":")+2
'''***********************************************'''
l_p_snvi = len(p_snvi)
print("Total Elements in p_sni: ", l_p_snvi, "\n\n")
'''#####################################################'''
```

```
import numpy as np

m=90; n=190; # m and n are the positions of an array

p_sni_extracted = np.array(p_sni[m:n])

p_snii_extracted = np.array(p_snii[m:n])

p_sniii_extracted = np.array(p_sniii[m:n])

p_sniv_extracted = np.array(p_sniv[m:n])

p_snv_extracted = np.array(p_snv[m:n])

p_snvi_extracted = np.array(p_snvi[m:n])

print("Extracted Data of Pressure Bar", p_sni_extracted, "\n\n")

'''************************************************'''

p_sni_extracted_diff = abs((np.average(p_sni_extracted[]))-

(np.array(p_sni_extracted[1:])))

p_snii_extracted_diff = abs((np.average(p_snii_extracted[]))-

(np.array(p_snii_extracted[1:])))

p_sniii_extracted_diff = abs((np.average(p_sniii_extracted[]))-

(np.array(p_sniii_extracted[1:])))

p_sniv_extracted_diff = abs((np.average(p_sniv_extracted[]))-

(np.array(p_sniv_extracted[1:])))

p_snv_extracted_diff = abs((np.average(p_snv_extracted[]))-

(np.array(p_snv_extracted[1:])))

p_snvi_extracted_diff = abs((np.average(p_snvi_extracted[]))-

(np.array(p_snvi_extracted[1:])))

print("Extracted Drop of Pressure Bar", p_sni_extracted_diff, "\n\n")
```

```
'''**********************************************'''
p_sni_psi = [i * 14.5038 for i in p_sni]
p_snii_psi = [i * 14.5038 for i in p_snii]
p_sniii_psi = [i * 14.5038 for i in p_sniii]
p_sniv_psi = [i * 14.5038 for i in p_sniv]
p_snv_psi = [i * 14.5038 for i in p_snv]
p_snvi_psi = [i * 14.5038 for i in p_snvi]
p_sni_extracted_psi = np.array(p_sni_psi[m:n])
p_snii_extracted_psi = np.array(p_snii_psi[m:n])
p_sniii_extracted_psi = np.array(p_sniii_psi[m:n])
p_sniv_extracted_psi = np.array(p_sniv_psi[m:n])
p_snv_extracted_psi = np.array(p_snv_psi[m:n])
p_snvi_extracted_psi = np.array(p_snvi_psi[m:n])
p_sni_extracted_diff_psi = abs((np.array(p_sni_extracted_psi[0]))-
(np.array(p_sni_extracted_psi[1:])))
p_snii_extracted_diff_psi = abs((np.array(p_snii_extracted_psi[0]))-
(np.array(p_snii_extracted_psi[1:])))
p_sniii_extracted_diff_psi = abs((np.array(p_sniii_extracted_psi[0]))-
(np.array(p_sniii_extracted_psi[1:])))
p_sniv_extracted_diff_psi = abs((np.array(p_sniv_extracted_psi[0]))-
(np.array(p_sniv_extracted_psi[1:])))
p_snv_extracted_diff_psi = abs((np.array(p_snv_extracted_psi[0]))-
(np.array(p_snv_extracted_psi[1:])))
```

```
p_snvi_extracted_diff_psi = abs((np.array(p_snvi_extracted_psi[0]))-
(np.array(p_snvi_extracted_psi[1:])))
'''#################################################'''
import matplotlib.pyplot as plt
from matplotlib import pylab
from matplotlib.font_manager import FontProperties
plt.subplot(3, 1, 1)
'''#################################################'''
fig = plt.figure()
ax = plt.subplot(111)
#ax.plot([i for i in range(len(p_sni))], p_sni, color = 'Green', label='Sensor 1')
ax.plot([i for i in range(len(p_snii))], p_snii, color = 'Blue', label='Sensor 1')
ax.plot([i for i in range(len(p_sniii))], p_sniii, color = 'Red', label='Sensor 2')
ax.plot([i for i in range(len(p_sniv))], p_sniv, color = 'Yellow', label='Sensor 3')
ax.plot([i for i in range(len(p_snv))], p_snv, color = 'Magenta', label='Sensor 4')
ax.plot([i for i in range(len(p_snvi))], p_snvi, color = 'Olive', label='Sensor 5')
plt.xlabel('Time(s)')
plt.ylabel('Pressure')
plt.title('Pressure value in Bar')
ax.legend(bbox_to_anchor=(1, 1))
plt.show()
fig.savefig('image_full.svg', format='svg', dpi=1200)
```

```
'''**********************************************'''
fig = plt.figure()
ax = plt.subplot(111)
#ax.plot([i for i in range(len(p_sni_extracted))], p_sni_extracted, color = 'Green',
label='Sensor 1')
ax.plot([i for i in range(len(p_snii_extracted))], p_snii_extracted, color = 'Blue',
label='Sensor 1')
ax.plot([i for i in range(len(p_sniii_extracted))], p_sniii_extracted, color = 'Red',
label='Sensor 2')
ax.plot([i for i in range(len(p_sniv_extracted))], p_sniv_extracted, color = 'Yellow',
label='Sensor 3')
ax.plot([i for i in range(len(p_snv_extracted))], p_snv_extracted, color = 'Magenta',
label='Sensor 4')
ax.plot([i for i in range(len(p_snvi_extracted))], p_snvi_extracted, color = 'Olive',
label='Sensor 5')
ax.plot([i for i in range(len(p_snvii_extracted))], p_snvii_extracted, color = 'Lime',
label='Sensor 6')
plt.ylabel('Pressure')
plt.title('Pressure value in Bar')
ax.legend(bbox_to_anchor=(1, 1))
plt.show()
fig.savefig('image_half.svg', format='svg', dpi=1200)
```

```
'''**********************************************'''
fig = plt.figure()
ax = plt.subplot(111)
#ax.plot([i for i in range(len(p_sni_extracted_diff))], p_sni_extracted_diff, color =
'Green', label='Sensor 1')
ax.plot([i for i in range(len(p_snii_extracted_diff))], p_snii_extracted_diff, color = 'Blue',
label='Sensor 1')
ax.plot([i for i in range(len(p_sniii_extracted_diff))], p_sniii_extracted_diff, color =
'Red', label='Sensor 2')
ax.plot([i for i in range(len(p_sniv_extracted_diff))], p_sniv_extracted_diff, color =
'Yellow', label='Sensor 3')
ax.plot([i for i in range(len(p_snv_extracted_diff))], p_snv_extracted_diff, color =
'Magenta', label='Sensor 4')
ax.plot([i for i in range(len(p_snvi_extracted_diff))], p_snvi_extracted_diff, color =
'Olive', label='Sensor 5')
ax.plot([i for i in range(len(p_snvii_extracted_diff))], p_snvii_extracted_diff, color =
'Lime', label='Sensor 6')
plt.xlabel('Time(s)')
plt.ylabel('Pressure Drop')
plt.title('Pressure Drop in Bar')
ax.legend(bbox_to_anchor=(1, 1))
plt.show()
fig.savefig('image_drop.svg', format='svg', dpi=1200)
```

```
'''**********************************************'''
p_sni_psi = [i * 14.5038 for i in p_sni]
p_snii_psi = [i * 14.5038 for i in p_snii]
p_sniii_psi = [i * 14.5038 for i in p_sniii]
p_sniv_psi = [i * 14.5038 for i in p_sniv]
p_snv_psi = [i * 14.5038 for i in p_snv]
p_snvi_psi = [i * 14.5038 for i in p_snvi]
'''**********************************************'''
p_sni_extracted_psi = np.array(p_sni_psi[m:n])
p_snii_extracted_psi = np.array(p_snii_psi[m:n])
p_sniii_extracted_psi = np.array(p_sniii_psi[m:n])
p_sniv_extracted_psi = np.array(p_sniv_psi[m:n])
p_snv_extracted_psi = np.array(p_snv_psi[m:n])
p_snvi_extracted_psi = np.array(p_snvi_psi[m:n])
'''**********************************************'''
p_sni_extracted_diff_psi = abs((np.array(p_sni_extracted_psi[0]))-
(np.array(p_sni_extracted_psi[1:])))
p_snii_extracted_diff_psi = abs((np.array(p_snii_extracted_psi[0]))-
(np.array(p_snii_extracted_psi[1:])))
p_sniii_extracted_diff_psi = abs((np.array(p_sniii_extracted_psi[0]))-
(np.array(p_sniii_extracted_psi[1:])))
p_sniv_extracted_diff_psi = abs((np.array(p_sniv_extracted_psi[0]))-
(np.array(p_sniv_extracted_psi[1:])))
```

```
p_snv_extracted_diff_psi = abs((np.array(p_snv_extracted_psi[0]))-
(np.array(p_snv_extracted_psi[1:])))
p_snvi_extracted_diff_psi = abs((np.array(p_snvi_extracted_psi[0]))-
(np.array(p_snvi_extracted_psi[1:])))
'''*********************************************'''
plt.plot([i for i in range(len(p_sni))], p_sni_psi, color = 'Green', label='Sensor 1')
plt.plot([i for i in range(len(p_snii))], p_snii_psi, color = 'Blue', label='Sensor 2')
plt.plot([i for i in range(len(p_sniii))], p_sniii_psi, color = 'Red', label='Sensor 3')
plt.plot([i for i in range(len(p_sniv))], p_sniv_psi, color = 'Yellow', label='Sensor 4')
plt.plot([i for i in range(len(p_snv))], p_snv_psi, color = 'Magenta', label='Sensor 5')
plt.plot([i for i in range(len(p_snvi))], p_snvi_psi, color = 'Olive', label='Sensor 6')
plt.xlabel('Time(s)')
plt.ylabel('Pressure')
plt.title('pressure value in PSI')
plt.legend()
plt.show()
'''*********************************************'''
plt.plot([i for i in range(len(p_sni_extracted_psi))], p_sni_extracted_psi, color = 'Green',
label='Sensor 1')
plt.plot([i for i in range(len(p_snii_extracted_psi))], p_snii_extracted_psi, color = 'Blue',
label='Sensor 2')
plt.plot([i for i in range(len(p_sniii_extracted_psi))], p_sniii_extracted_psi, color = 'Red',
label='Sensor 3')
```

```
plt.plot([i for i in range(len(p_sniv_extracted_psi))], p_sniv_extracted_psi, color = 'Yellow', label='Sensor 4')
plt.plot([i for i in range(len(p_snv_extracted_psi))], p_snv_extracted_psi, color = 'Magenta', label='Sensor 5')
plt.plot([i for i in range(len(p_snvi_extracted_psi))], p_snvi_extracted_psi, color = 'Olive', label='Sensor 6')
plt.ylabel('Pressure')
plt.title('Pressure value in PSI')
plt.legend()
plt.show()
'''*********************************************'''
plt.plot([i for i in range(len(p_sni_extracted_diff_psi))], p_sni_extracted_diff_psi, color = 'Green', label='Sensor 1')
plt.plot([i for i in range(len(p_snii_extracted_diff_psi))], p_snii_extracted_diff_psi, color = 'Blue', label='Sensor 2')
plt.plot([i for i in range(len(p_sniii_extracted_diff_psi))], p_sniii_extracted_diff_psi, color = 'Red', label='Sensor 3')
plt.plot([i for i in range(len(p_sniv_extracted_diff_psi))], p_sniv_extracted_diff_psi, color = 'Yellow', label='Sensor 4')
plt.plot([i for i in range(len(p_snv_extracted_diff_psi))], p_snv_extracted_diff_psi, color = 'Magenta', label='Sensor 5')
plt.plot([i for i in range(len(p_snvi_extracted_diff_psi))], p_snvi_extracted_diff_psi, color = 'Olive', label='Sensor 6')
plt.xlabel('Time(s)')

plt.ylabel('Pressure Drop')

plt.title('Pressure Drop in PSI')

plt.legend()

plt.show()
```

APPENDIX C. Matlab code for exponential curve fitting

```
userpath ('')
for j = 1:n
    filename = ['' int2str(j)];
    data = xlsread(filename);
    start_index = 1;
    end_index = size(data,1);
    time = (1:size(data,1))';
    c = [;];
for I =1:6
    figure(i)
    f = fit(time(start_index:end_index), data(start_index:end_index,i), 'exp1');
    c = [c; coeffvalues(f)];
plot(time(start_index:end_index),data(start_index:end_index,i),'b-')
hold on
plot(f,time(start_index:end_index),data(start_index:end_index,i))
end
%addexp = c(:,4)+c(:,2);
[~,I] = min(c(:,2));
%addexp(I) = 10000;
%[~,J] = min(addexp);
display(['The leak is near sensor  ' int2str(I)])
end
```

APPENDIX D. Python code for SVM and ANN

```
'''**********************************************'''
from sklearn import svm
from sklearn import metrics
'''**********************************************'''
clf = svm.SVC(kernel='linear')  # Create a svm Classifier object and Linear Kernel
clf.fit(tr_features,tr_label)   # train model
y_pred = clf.predict(ts_features)   # predict model
print('SVM Accuracy for Linear: ', metrics.accuracy_score(ts_label, y_pred))
clf = svm.SVC(kernel='poly', degree=2)
clf.fit(tr_features,tr_label)   # train model
y_pred = clf.predict(ts_features)   # predict model
print('SVM Accuracy for Polynomial: ', metrics.accuracy_score(ts_label, y_pred))
clf = svm.SVC(kernel='rbf')  # Create a svm Classifier object and Linear Kernel
clf.fit(tr_features,tr_label)   # train model
y_pred = clf.predict(ts_features)   # predict model
print('SVM Accuracy for Gaussian: ', metrics.accuracy_score(ts_label, y_pred))
clf = svm.SVC(kernel='sigmoid')  # Create a svm Classifier object and Linear Kernel
clf.fit(tr_features,tr_label)   # train model
y_pred = clf.predict(ts_features)   # predict model
print('SVM Accuracy for Sigmoid: ', metrics.accuracy_score(ts_label, y_pred))
'''**********************************************'''
```

```
from sklearn import metrics

n_n = [1,2,3,4,5]      # hidden layers

import itertools

from sklearn.neural_network import MLPClassifier

'''**********************************************'''

for idx,layer in enumerate(tuple(itertools.permutations(n_n))):

   clf = MLPClassifier(solver='lbfgs', alpha=1e-5,hidden_layer_sizes=tuple([x*10 for x

in layer]), random_state=1)

   clf.fit(tr_features,tr_label)

   y_pred = clf.predict(ts_features)

   print(idx, ' : NN Accuracy: ', tuple([x*1 for x in layer])

,metrics.accuracy_score(ts_label, y_pred))

'''**********************************************'''

print(__doc__)


import itertools

import numpy as np

import matplotlib.pyplot as plt

plt.rc('xtick', labelsize=10)

plt.rc('ytick', labelsize=10)

from sklearn.metrics import confusion_matrix

class_names = np.array(['0.5-inch', '0.4-inch', '0.3-inch','0.2-inch', '0.1-inch']

print('finalized model efficiency - ',metrics.accuracy_score(ts_label, y_pred)
```

APPENDIX E. Example of training date sets for SVM and ANN

Table 20. Training set

Data set	Sensor-1	Sensor-2	Sensor-3	Sensor-4	Sensor-5	Sensor-6	pipe diameter	leak diameter
1	0.09	0.08	0.08	0.08	0.07	0.06	0.75	1
2	0.06	0.06	0.06	0.05	0.04	0.02	0.75	1
3	0.07	0.06	0.06	0.06	0.04	0.03	0.75	1
4	0.09	0.09	0.08	0.08	0.07	0.05	0.75	1
5	0.09	0.08	0.08	0.07	0.07	0.06	0.75	1
6	0.06	0.06	0.06	0.05	0.05	0.03	0.75	1
7	0.08	0.06	0.06	0.06	0.04	0.03	0.75	1
8	0.09	0.08	0.08	0.08	0.07	0.05	0.75	1
9	0.15	0.13	0.13	0.12	0.12	0.1	0.75	2
10	0.17	0.15	0.15	0.15	0.13	0.13	0.75	2
11	0.16	0.15	0.14	0.14	0.14	0.13	0.75	2
12	0.15	0.14	0.13	0.12	0.12	0.11	0.75	2
13	0.17	0.15	0.15	0.15	0.14	0.13	0.75	2
14	0.16	0.16	0.14	0.14	0.14	0.13	0.75	2
15	0.14	0.14	0.13	0.13	0.13	0.11	0.75	2
16	0.15	0.14	0.14	0.13	0.12	0.12	0.75	2
17	0.16	0.15	0.15	0.15	0.14	0.14	0.75	2
18	0.16	0.16	0.16	0.15	0.14	0.13	0.75	2
19	0.28	0.24	0.21	0.21	0.2	0.2	0.75	3
20	0.29	0.25	0.22	0.22	0.2	0.18	0.75	3
21	0.26	0.21	0.19	0.19	0.15	0.15	0.75	3
22	0.22	0.17	0.17	0.16	0.16	0.15	0.75	3
23	0.31	0.25	0.23	0.22	0.2	0.18	0.75	3
24	0.27	0.25	2.2	0.19	0.17	0.15	0.75	3
25	0.24	0.19	0.16	0.16	0.15	0.14	0.75	3
26	0.24	0.19	0.16	0.16	0.15	0.14	0.75	3
27	0.24	0.19	0.16	0.16	0.15	0.14	0.75	3
28	0.22	0.16	0.18	0.14	0.14	0.13	0.75	3

Table 20. Continued

29	0.26	0.21	0.19	0.18	0.18	0.16	0.75	3
30	0.27	0.23	0.21	0.18	0.17	0.17	0.75	3
31	0.28	0.24	0.21	0.21	0.2	0.2	0.75	3
32	0.24	0.2	0.18	0.18	0.16	0.14	0.75	3
33	0.44	0.32	0.32	0.3	0.24	0.23	0.75	4
34	0.4	0.28	0.26	0.26	0.23	0.21	0.75	4
35	0.45	0.34	0.33	0.3	0.23	0.21	0.75	4
36	0.4	0.3	0.29	0.28	0.26	0.25	0.75	4
37	0.41	0.29	0.27	0.25	0.22	0.21	0.75	4
38	0.42	0.31	0.28	0.26	0.23	0.22	0.75	4
39	0.39	0.27	0.25	0.23	0.22	0.21	0.75	4
40	0.4	0.28	0.25	0.24	0.22	0.2	0.75	4
41	0.43	0.32	0.27	0.27	0.23	0.22	0.75	4
42	0.71	0.58	0.47	0.42	0.38	0.37	0.75	5
43	0.75	0.6	0.52	0.47	0.43	0.41	0.75	5
44	0.69	0.5	0.46	0.42	0.4	0.37	0.75	5
45	0.66	0.48	0.44	0.38	0.35	0.33	0.75	5
46	0.67	0.5	0.5	0.4	0.38	0.36	0.75	5
47	0.75	0.58	0.47	0.42	0.39	0.37	0.75	5
48	0.78	0.6	0.57	0.51	0.46	0.45	0.75	5
49	0.65	0.49	0.42	0.37	0.34	0.33	0.75	5
50	0.67	0.5	0.46	0.4	0.36	0.35	0.75	5
51	0.06	0.05	0.05	0.02	0.02	0.02	1	1
52	0.05	0.06	0.06	0.04	0.03	0.03	1	1

APPENDIX F. Example of testing data sets for SVM and ANN

Table 21. Testing set

Data set	Sensor-1	Sensor-2	Sensor-3	Sensor-4	Sensor-5	Sensor-6	pipe diameter	leak diameter
1	0.1	0.09	0.09	0.08	0.07	0.05	0.75	1
2	0.09	0.08	0.09	0.07	0.06	0.06	0.75	1
3	0.06	0.06	0.05	0.05	0.05	0.03	0.75	1
4	0.28	0.24	0.22	0.21	0.2	0.2	0.75	3
5	0.28	0.26	0.24	0.23	0.22	0.2	0.75	3
6	0.3	0.28	0.25	0.23	0.2	0.2	0.75	3
7	0.25	0.2	0.19	0.18	0.18	0.16	0.75	3
8	0.19	0.18	0.18	0.17	0.16	0.15	0.75	3
9	0.39	0.27	0.25	0.25	0.23	0.22	0.75	4
10	0.4	0.28	0.25	0.24	0.22	0.22	0.75	4
11	0.92	0.71	0.66	0.55	0.51	0.5	0.75	5
12	0.83	0.65	0.55	0.47	0.44	0.43	0.75	5
13	0.77	0.58	0.51	0.43	0.4	0.38	0.75	5
14	0.7	0.53	0.44	0.38	0.36	0.34	0.75	5
15	0.37	0.31	0.3	0.3	0.27	0.28	1	5

REFERENCES

[1] NEC110414, “NEC, Texas State partner to collaborate on social infrastructure projects: Office of Media Relations: Texas State University,” 08-Jun-2016. [Online]. Available: http://www.txstate.edu/news/news_releases/news_archive/2014/November-2014/NEC110414.html. [Accessed: 10-Sep-2018].

[2] “WHO | Domestic water quantity, service level and health,” WHO. [Online]. Available: http://www.who.int/water_sanitation_health/publications/wsh0302/en/. [Accessed: 20-Sep-2018].

[3] P. R. Hunter, A. M. MacDonald, and R. C. Carter, “Water Supply and Health,” *PLOS Medicine*, vol. 7, no. 11, p. e1000361, Nov. 2010.

[4] “Agricultural Water| Other Uses of Water | Healthy Water | CDC.” [Online]. Available: https://www.cdc.gov/healthywater/other/agricultural/index.html. [Accessed: 09-Oct-2018].

[5] “Handbook of Water and Wastewater Systems Protection - Google Books.” [Online].

[6] “1 Introduction | Drinking Water Distribution Systems: Assessing and Reducing Risks | The National Academies Press.” [Online]. Available: https://www.nap.edu/read/11728/chapter/3#17. [Accessed: 07-Oct-2018].

[7] “Second Edition Handbook of PE Pipe | HDPE Handbook.” [Online]. Available: https://plasticpipe.org/publications/pe-handbook.html. [Accessed: 07-Oct-2018].

[8] “Black Snake in the Grass | A\J – Canada’s Environmental Voice.” [Online]. Available: https://www.alternativesjournal.ca/energy-and-resources/black-snake-grass. [Accessed: 07-Oct-2018].

[9] “Irrigation Engineering: LESSON 15 Underground Pipeline Systems.” [Online]. Available: http://ecoursesonline.iasri.res.in/mod/page/view.php?id=124826. [Accessed: 07-Oct-2018].

[10] “Photo Gallery – PVC Conduit.” [Online]. Available: http://www.primeconduit.com/photo-gallery-pvc-conduit/. [Accessed: 07-Oct-2018].

[11] S. Oven, “Leak Detection in Pipelines by the use of State and Parameter Estimation”, 2014.

[12] Colombo Andrew F. and Karney Bryan W., “Energy and Costs of Leaky Pipes: Toward Comprehensive Picture,” *Journal of Water Resources Planning and Management*, vol. 128, no. 6, pp. 441–450, Nov. 2002.

[13] O. Egeland and J. T. Gravdahl, Modeling and simulation for automatic control, Corr., 2. print. Trondheim: Marine Cybernetics AS, 2003.

[14] “Water main breaks close Ventura Boulevard in Studio City [Updated] | L.A. NOW | Los Angeles Times.” [Online]. Available: http://latimesblogs.latimes.com/lanow/2011/11/two-water-main-breaks-in-studio-city-area.html. [Accessed: 21-Sep-2018].

[15] D. Barer, “Billions of gallons of water lost due to leaky pipes,” *KXAN*, 22-Jul-2015. [Online]. Available: https://www.kxan.com/news/billions-of-gallons-of-water-lost-due-to-leaky-pipes/1156467330. [Accessed: 21-Sep-2018].

[16] U. EPA, “Page Name.” [Online]. Available: https://19january2017snapshot.epa.gov/www3/watersense/pubs/fixleak.html. [Accessed: 21-Sep-2018].

[17] O. US EPA, “EPA’s 6th Drinking Water Infrastructure Needs Survey and Assessment,” US EPA, 30-Mar-2018. [Online]. Available: https://www.epa.gov/drinkingwatersrf/epas-6th-drinking-water-infrastructure-needs-survey-and-assessment. [Accessed: 21-Sep-2018].

[18] A. Sadeghioon, N. Metje, D. Chapman, and C. Anthony, “SmartPipes: Smart Wireless Sensor Networks for Leak Detection in Water Pipelines,” *Journal of Sensor and Actuator Networks*, vol. 3, pp. 64–78, Feb. 2014.

[19] B. V. Hieu, S. Choi, Y. U. Kim, Y. Park, and T. Jeong, “Wireless transmission of acoustic emission signals for real-time monitoring of leakage in underground pipes,” *KSCE J Civ Eng*, vol. 15, no. 5, p. 805, May 2011.

[20] Khulief Y. A., Khalifa A., Mansour R. Ben, and Habib M. A., “Acoustic Detection of Leaks in Water Pipelines Using Measurements inside Pipe,” *Journal of Pipeline Systems Engineering and Practice*, vol. 3, no. 2, pp. 47–54, May 2012.

[21] P. Karkulali, H. Mishra, A. Ukil, and J. Dauwels, “Leak detection in gas distribution pipelines using acoustic impact monitoring,” in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 412–416.

[22] X. Cui, Y. Yan, M. Guo, Y. Hu, and X. Han, “Localization of continuous gas leaks from a flat-surface structure using an Acoustic Emission sensor array,” in *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, 2016, pp. 1–5.

[23] M. Klingajay and T. Jitson, “Real-time Laser Monitoring based on Pipe Detective Operation,” vol. 2, no. 6, p. 6, 2008.

[24] “Intelligent System for Condition Monitoring of Underground Pipelines - Sinha - 2004 - Computer-Aided Civil and Infrastructure Engineering - Wiley Online Library.” [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8667.2004.00336.x?deniedAccessCustomisedMessage=&userIsAuthenticated=false. [Accessed: 20-Sep-2018].

[25] A. Gautam, R. R. Singh, A. Kumar, and V. Priye, “Architecture of optical sensing and monitoring for pipelines using FBG,” in *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, 2016, pp. 337–338.

[26] H. Murayama, “Structural health monitoring based on strain distributions measured by fiber-optic distributed sensors,” in *2015 Opto-Electronics and Communications Conference (OECC)*, 2015, pp. 1–2.

[27] “Permanent Leak Detection on Pipes Using a Fibre Optic Based Continuous Sensor Technology,” ResearchGate. [Online]. Available: https://www.researchgate.net/publication/269133920_Permanent_Leak_Detection_on_Pipes_Using_a_Fibre_Optic_Based_Continuous_Sensor_Technology. [Accessed: 20-Sep-2018].

[28] S. Eyuboglu, H. Mahdi, and H. Al-Shukri, “DETECTION OF WATER LEAKS USING GROUND PENETRATING RADAR,” p. 17.

[29] Z. Liu and Y. Kleiner, “State-of-the-Art Review of Technologies for Pipe Structural Health Monitoring,” *IEEE Sensors Journal*, vol. 12, no. 6, pp. 1987–1992, Jun. 2012.

[30] M. JayaLakshmi and V. Gomathi, “An enhanced underground pipeline water leakage monitoring and detection system using Wireless sensor network,” in *2015 International Conference on Soft-Computing and Networks Security (ICSNS)*, 2015, pp. 1–6.

[31] S. Adsul, A. K. Sharma, and R. G. Mevekari, “Development of leakage detection system,” in *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, 2016, pp. 673–677.

[32] I. F. Akyildiz, Z. Sun, and M. C. Vuran, “Signal propagation techniques for wireless underground communication networks,” *Physical Communication*, vol. 2, no. 3, pp. 167–183, Sep. 2009.

[33] S. Jiang, S. V. Georgakopoulos, and O. Jonah, “RF power harvesting for underground sensors,” in *Proceedings of the 2012 IEEE International Symposium on Antennas and Propagation*, 2012, pp. 1–2.

[34] S. Yoon et al., “Subsurface monitoring using low frequency wireless signal networks,” in *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*, 2012, pp. 443–446.

[35] K. Panjabi et al., “Development and Field Evaluation of a Low-Cost Wireless Sensor Network System for Hydrological Monitoring of a Small Agricultural Watershed,” *Open Journal of Civil Engineering*, vol. 08, p. 166, Jun. 2018.

[36] A. Pettersson, J. Nordlander, and S. Gong, “ZigBee-Ready Wireless Water Leak Detector,” in *2009 Third International Conference on Sensor Technologies and Applications*, 2009, pp. 105–108.

[37] Aditya Engineering College Beed, Maharashtra, P. C. H. Chavan, and M. P. V. Karande, “Wireless Monitoring of Soil Moisture, Temperature & Humidity Using Zigbee in Agriculture,” *International Journal of Engineering Trends and Technology*, vol. 11, no. 10, pp. 493–497, May 2014.

[38] I. Jawhar, N. Mohamed, and K. Shuaib, “A framework for pipeline infrastructure monitoring using wireless sensor networks,” in *2007 Wireless Telecommunications Symposium*, 2007, pp. 1–7.

[39] G. Dang and X. Cheng, “Application of wireless sensor network in monitoring system based on Zigbee,” in *2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA)*, 2014, pp. 181–183.

[40] J. Mashford, D. D. Silva, D. Marney, and S. Burn, “An Approach to Leak Detection in Pipe Networks Using Analysis of Monitored Pressure Values by Support Vector Machine,” in *2009 Third International Conference on Network and System Security*, 2009, pp. 534–539.

[41] 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, pp. 1–4.

[42] 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, pp. 3288–3292.

[43] O. US EPA, “EPANET,” US EPA, 24-Jun-2014. [Online]. Available: https://www.epa.gov/water-research/epanet. [Accessed: 08-Oct-2018].

[44] “Common Types of Pressure Sensors.” [Online]. Available: https://www.thomasnet.com/articles/instruments-controls/pressure-sensors. [Accessed: 23-Sep-2018].

[45] “Arduino - Home.” [Online]. Available: https://www.arduino.cc/. [Accessed: 23-Sep-2018].

[46] “Arduino Mega 2560 R3 - DEV-11061 - SparkFun Electronics.” [Online]. Available: https://www.sparkfun.com/products/11061. [Accessed: 08-Oct-2018].

[47] B. Uddin, A. Imran, and M. A. Rahman, “Detection and locating the point of fault in distribution side of power system using WSN technology,” in *2017 4th International Conference on Advances in Electrical Engineering (ICAEE)*, 2017, pp. 570–574.

[48] T. Kumar and P. B. Mane, “ZigBee topology: A survey,” in *2016 International Conference on Control, Instrumentation*, Communication and Computational Technologies (ICCICCT), 2016, pp. 164–166.

[49] “How XBee devices communicate.” [Online]. Available: https://www.digi.com/resources/documentation/Digidocs/90001456 13/concepts/c_how_xbees_communicate.htm. [Accessed: 23-Sep-2018].

[50] “XBee Buying Guide - SparkFun Electronics.” [Online]. Available: https://www.sparkfun.com/pages/xbee_guide. [Accessed: 08-Oct-2018].

[51] “Data Processing | Meaning, Definition, Steps, Types and Methods,” Planning TankTM, 15-Jun-2017.

[52] “About Feature Scaling and Normalization,” Dr. Sebastian Raschka, 11-Jul-2014. [Online]. Available: https://sebastianraschka.com/Articles/2014_about_feature_scaling.html. [Accessed: 04-Oct-2018].

[53] “Z-Score: Definition, Formula and Calculation,” Statistics How To. [Online]. Available: https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/z-score/. [Accessed: 04-Oct-2018].

[54] Sifium, “Types of classification algorithms in Machine Learning,” Medium, 28-Feb-2017.

[55] L. Dormehl, “What is an artificial neural network? Here’s everything you need to know,” Digital Trends, 13-Sep-2018. [Online]. Available: https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/. [Accessed: 04-Oct-2018].

[56] F. Yang, Water Leak Detection and Localization Using Multi-sensor Data Fusion. 2012.

[57] M. T. Hagan, Neural Network DesignPaperback. Martin Hagan, 2002.

[58] “Implementing SVM and Kernel SVM with Python’s Scikit-Learn,” Stack Abuse, 17-Apr-2018. [Online]. Available: https://stackabuse.com/implementing-svm-and-kernel-svm-with-pythons-scikit-learn/. [Accessed: 13-Sep-2018].

[59] J. Brownlee, “Support Vector Machines for Machine Learning,” Machine Learning Mastery, 19-Apr-2016.

[60] “Kernel Functions for Machine Learning Applications – César Souza.”.

[61] J. Brownlee, “What is a Confusion Matrix in Machine Learning,” Machine Learning Mastery, 17-Nov-2016.

[62] “Pipe Pressure Drop Calculations Formula, Theory and Equations.” [Online]. Available: https://www.pipeflow.com/pipe-pressure-drop-calculations. [Accessed: 07-Nov-2018].

[63] “On-Line Friction Piping Loss.” [Online]. Available: http://www.freecalc.com/fricfram.htm. [Accessed: 07-Nov-2018].