Water Quality Monitoring Using Internet of Things (IoT) and Machine Learning (ML) for Domestic Application

By

Ahmed Mahad Mohamed
21371010

A thesis submitted to the Department of Electrical and Electronic Engineering in partial fulfillment of the requirements for the degree of M.Eng. in Electrical and Electronic Engineering

Electrical and Electronic Engineering
Brac University
August 2022

Declaration

It is hereby declared that

1. The thesis submitted is my own original work while completing my degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material that has been accepted or submitted, for any other degree or diploma at a university or other institution.
4. I have acknowledged all main sources of help.

Student's Full Name & Signature:

Ahmed Mahad Mohamed
21371010

Approval

The project **"Water Quality Monitoring Using Internet of Things (IoT) and Machine Learning (ML) for Domestic Application"** was submitted by

1. Ahmed Mahad Mohamed (21371010)

of Summer, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of M.Eng. in Electrical and Electronic Engineering on **07/08/2022**.

Examining Committee:

Supervisor:
(Member)

____

Dr. Abu S. M. Mohsin, PhD
Associate Professor
Department of Electrical and Electronic Engineering,
Brac University

Internal Expert Examiner:
(Member)

____

Dr. Mohammed Belal Hossain Bhuian, PhD
Associate Professor
Department of Electrical and Electronic Engineering
Brac University

Departmental Head:
(Chair)

____

Dr. Md. Mosaddequr Rahman, PhD
Professor and Chairperson,
Department of Electrical and Electronic Engineering
Brac University

Abstract/ Executive Summary

Water is one of the greatest blessings that nature has to offer, and it is necessary for the survival of all living things, including humans, animals, and plants. At this time, challenges are being faced all around the world in meeting the requirements for water that can be consumed. The most severely impacted by this crisis are those in underdeveloped countries. Because there is not enough adequate monitoring of water quality and quantity, the current water issue is just going to get worse as the population continues to grow.

With the recent advancement in information and communication system technology, there is a growing interest in the development of smart and cost-effective solutions for water quality monitoring system (WQMS). To get a lasting solution to the above problem, we built a system containing eight sensors (PH, Temperature, TDS, Turbidity, Pressure, Volume, Color, and Flow) that can monitor the quality and quantity of the household water. All these sensors were connected with an esp32 Wi-Fi module, and all the data collected from the sensors were stored into a cloud-based server. Afterwards water quality data was visualized in the Thingspeak server in real time. Additionally, we employed two machine learning algorithms such as FB prophet and SARIMA ARIMA and predicted the future values.

The primary goal of this project is to develop a water quality monitoring system (WQMS) so that the consumer can monitor the water quality in real-time and can use the safe water for drinking. This low-cost monitoring system uses the new technology Internet of Things, and Machine Learning, all of which have the potential to replace the conventional methods of water quality monitoring. Customers, vendors, and the government all can benefit from this system as communication can move in both directions. This approach will enable the government not only to obtain taxes but also help to regulate the water situation, which will lead to an increase in customer trust in the government and the suppliers. The developed system can surely use for water quality monitoring systems. To check the validation of the developed system I use two different sources (normal water source and contaminated water source) the result was as expected different sources have different results.

Keywords: WQMS, Internet of Things, Water sensors, Machine learning algorithms, RSME, FB Prophet, and Sarima Arima analysis.

Dedication

My dissertation is dedicated to my family and friends. A particular thank you to my darling Mother Khadra Mohamud, who has always supported, encouraged, and shown me the way to success, as well as my wonderful Uncle, who has helped me become the man I am today.

Acknowledgement

I would like to convey my gratitude to my supervisor, Dr. A. S. M. Mohsin, Assistant Professor, Department of Electrical and Electronic Engineering, BRAC University, Dhaka, Bangladesh, who has provided me with countless recommendations, guidance, and help during this process.

I'd also want to thank the excellent faculty members and staff of the Department of Electrical and Electronic Engineering at BRAC University for their prolonged assistance and collaboration.

I would like to acknowledge the "ICT Ministry Bangladesh-Innovation Grant" for the financial support.

In addition, I want to use this opportunity to thank everyone who has contributed to my education at BRAC University.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Water covers almost 75% of the earth's surface, demonstrating how important water is to living organisms. Every living organism requires water to thrive and stay alive. Body function maintains and retains hydration because every cell, organ, and tissue require water to assist control temperature. Water also serves as a lubricant and a cushion for your joints. The throne of Allah is defined as the standing of the Qur'an, and it is mentioned in the verse, which is likely the most cited verse among the verses, as water is a vital element that permits living organisms to live. Even though water is the source of all life. Unfortunately, only around 3% of the total volume of fresh water is suitable for human use. In recent decades, the human community has faced the wrath of water scarcity, which has been exacerbated by uncontrolled urbanization and industrialization, which has further tainted the scarce quantity accessible for consumption. Water contamination has increased in recent years, making water unclean and unfit to drink. Furthermore, many impoverished families live near lakes or rivers, and they routinely consume water that is unsafe to drink since they have no means of knowing whether it is safe or not. This may lead to the consumption of water of unknown amounts and quality. So, to find a solution to this problem, we proposed to develop a system that will address the issues listed above. The proposed system can monitor the water quantity and quality[1]. It will also allow suppliers and customers to keep a constant eye on each other. For the development of that system, we propose to use the Internet of Things and deploy machine learning for future data prediction purpose. This technology will assist us in monitoring the water, determining whether it is safe to drink, and establishing a genuine relationship between water suppliers.

1.2 Source of Contamination

Since water makes up around two-thirds of the Earth, protecting it should be a top responsibility for everyone on the planet. Contamination of water sources by microbes and chemicals reduces water quality and may even render the water poisonous. Negative effects on human health, ecosystem health, and the economy can result from water contamination.

Water contamination is a major problem all around the world nowadays. "Water pollution" refers to the presence of harmful elements in aquatic environments. It occurs when toxins and untreated wastes infiltrate water systems. Because of this, water sources such as lakes, rivers, the sea, reservoirs, ponds, and even groundwater can get contaminated.

Source of water contamination can be classified into underground water contamination sources and surface water contamination sources.

1.2.1 Underground water contamination sources

Groundwater is a key supply of fresh water for the world's population and it is used for home, agricultural, and industrial purposes. Water that seeps deep into the ground is called groundwater.

Groundwater contamination is a global issue with serious consequences for human health and ecological services. The studies reported in this special issue focus on contaminants in groundwater that are both geogenic and anthropogenic in origin, and they are spread across a wide geographic range, with contributions from researchers studying groundwater contamination in India, China, Pakistan, Turkey, Ethiopia, and Nigeria[2].

Bangladesh has the world's worst groundwater Arsenic (As) contamination problem. Because surface water is mishandled, 97 percent of the country's population relies on groundwater for drinking and domestic use. In Bangladesh, high amounts of Arsenic (As) in groundwater are causing widespread poisoning. The World Health Organization (WHO) recommends a safe limit of 10 g L1 for As in drinking water[3].

1.2.2 Surface of water contamination Sources

Any body of water that is above ground is considered to be surface water, and this includes creeks, wetlands, lakes, rivers, and streams. Despite its saltwater content, the ocean is regarded as surface water.

The pollution of aquatic systems that are above ground, such as streams, lakes, and rivers, is referred to as surface water pollution. Rainwater runoff introduces toxins into the water, causing these waters to become contaminated. The pollutants carried by runoff include nutrients and fertilizers from farms and lawns as well as salts and chemicals from municipal and highway traffic.

Nutrients, bacteria, plastics, and chemicals including antibiotics, heavy metals, and pesticides are frequently to blame for surface water contamination. These contaminants' effects on the environment vary. For instance, too many nutrients can cause hypoxia and toxic algal blooms in rivers and coastal oceans. River pathogens are dangerous to people's health. Toxic impacts from chemical contamination are possible. Surface waterways frequently experience the combined effects of several contaminants[4].

Bangladesh, one of the world's most densely populated nations, has plenty of water sources, yet these sources are constantly being polluted. Different contaminants include coliforms, harmful trace metals, and other organic and inorganic pollutants are found in both surface water and groundwater sources.

1.2.3 Types of Drinking water Contaminants

Natural contaminant kinds and concentrations are determined by the geological materials through which groundwater travels and the quality of the recharge water.

The effects of these natural contaminations depend on the types and amounts of the compounds that groundwater passing through sedimentary rocks and soils may pick up, including magnesium, calcium, and chloride, arsenate, fluoride, nitrate, and iron, as well as other substances. Water can also get contaminated by naturally occurring elements that are present in unsafe amounts[5],[6],[7].

Hazardous chemicals, colors, and substances like insecticides and fertilizers are other man-made byproducts of industry and agriculture, along with heavy metals like mercury, copper, chromium, and lead.

Ground water pollution can result from improper handling, storage, or disposal of home chemicals such paints, synthetic detergents, solvents, oils, medications, disinfectants, pool chemicals, pesticides, batteries, gasoline, and diesel fuel[8],[9].

Pathogens such as bacteria, viruses, and parasites such as microscopic protozoa and worms are examples of microbiological pollutants. These living organisms can be transmitted by human and animal wastes, either intentionally or unintentionally.

By examining the water's color, odor, turbidity, and flavor, some toxins are simple to spot. The majority, however, are difficult to spot and need for testing to determine whether water is poisoned or not. As a result, the pollutants may have an unpleasant taste or odor, cause discoloration, or have other negative effects on health.

Table 1: Types of Contaminants found in drinking water

Name of the Contaminant	Description	Reference
Inorganic/Chemical Contaminants	These contaminants could be natural or man-made. Nitrogen, bleach, salts, pesticides, metals, bacterial toxins, and human or animal medications are examples of chemical contaminants.	[10]
organic/Physical Contaminants	Contaminants largely affect the physical appearance or other physical aspects of water. Sediment or organic material suspended in the water of lakes, rivers, and streams due to soil erosion are examples of physical pollutants.	[11]
Biological Contaminants	These are organisms found in water. They are also known as microorganisms or microbiological contaminants. Bacteria, viruses, protozoa, and parasites are all examples of biological or microbiological pollutants.	[12]
Radiological Contaminants	Chemical elements having an unequal number of protons and neutrons produce unstable atoms capable of emitting ionizing radiation. Cesium, plutonium, and uranium are examples of radioactive pollutants.	[13]

By examining different water parameters, we can easily identify the components of water and quality of drinking water by identifying its contaminant's level by using individual contaminant with its particular parameters. These parameters also identify the safety of drinking water, each of these has parameters has range of drinking water safety, if the water goes beyond that range its means the safety of water is not suitable for drinking. These parameters are listed in Table2 with their average values.

Table 2: Parameters responsible for designing water quality monitoring system

Parameter name	Reference Value
Arsenic[14]	0.01mg/L
Fluoride[15]	4.0 mg/L
Mercury[16]	0.002 mg/L
Copper[17],[18]	1.3 mg/L
Nitrate[19]	10 mg/L
Lead[20]	0.015 mg/L
Selenium[21]	0.01 mg/L
PH[22]	6.5-8.5
Temperature[23]	20-30
Color[24]	100
Turbidity [25]	1
TDS[26]	300 mg/L
Pesticides[27]	0.07 mg/L
Uranium[28]	2 Mcg/L

1.3 Literature Review

Many methods have been proposed, and many systems have been developed for water management and smart billing. This section has reviewed some research literature on water quality, quantity, billing, and monitoring systems.

Ryu [1] shows how UASWQP (a UAS-based real-time water quality monitoring, sampling, and visualization platform) may be used with cutting-edge technologies and cloud web services. For real-time water quality sampling, monitoring, and displaying at the urban-rural interface's environmental hotspot, this application utilizes off-the-shelf unmanned aircraft systems (UAS, often known as drones). Using Verizon LTE wireless connection, the planned UASWQP allows open water bodies to communicate real-time water quality data to the ThingSpeak IoT Cloud web service. Water pH, water temperature, Electrical Conductivity (EC), and dissolved oxygen are among the data collected. The proposed UASWQP platform can obtain raw data at a range of waterways, mainly where access is limited and/or where hazards exist.

Jha et al. [29] have provided a SWMS (Smart Water Monitoring System) solution based on the Raspberry Pi and Arduino UNO for measuring water quality and quantity. This digital monitoring system uses this uploaded data to compare with industrial standards and recommends usage through the Thingspeak cloud.

Using Arduino Uno, Node MCU, and sensors, Jain et al.[30] described a Water Quality Monitoring and Management System that detects untimed water supply, polluted water entering the tanks, tank overflowing, and unobserved daily water usage. As data is uploaded to the Adafruit cloud, the system tells the consumer about water quality and daily water consumption level via the IFTTT app, SMS, or email alert.

Saraswathi V[31] has introduced a system using the Internet of Things (IoT) to overcome water wastage. The proposed system approach to forecast and monitor water consumption consists primarily of a flow meter, microcontroller, and cloud infrastructure. The water flow rate was measured using a Hall effect-based flow meter, while Arduino Uno and Raspberry Pi were utilized as microcontroller-based devices. The Cloud interface was set up to visualize the data by end-users. In this model, alert messages will be sent to the user after a certain usage level so that the user can acknowledge and act accordingly.

Das Anirudh et al. posed a paradigm for smart water supply to improve drinking water quality. This system could also prevent access to water, reuse, and preserve underground water. Anirudh used a water conductivity sensor to measure the concentration of ions in the water solution, thus measuring the impurities of the water. The researcher also advised that the mineral content of the filtered water from the substation be replenished using a mineral cartridge. The smart water supply system would notify the user surface when it is time to replace the cartridge.

An IoT-based framework has been proposed[32] for a water distribution network that would collect real-time water usage information. Its main disadvantage is that if the instructions are not followed, it results in a limited water supply.

Sarker et al.[33] designed a model that would heat the water automatically at a specific temperature. The model will automatically turn the pump motor on, and the water will flow from the reserve tank to tank 1. Tank 1 contains two solenoid valves that pump water in two directions: the water filtration tank and the other to tank 2. Tank2 has a heater activated by clicking a button on the mobile app. The temperature is displayed on the LCD when it hits 35 degrees Celsius and then goes off automatically. The application makes use of Arduino and the C/C++ programming language to allow the water level to be turned on and off automatically based on temperature detection and stopping. Although this system has the advantage of automatic water heating, it lacks water quality assessment and cooling abilities.

Yazhini and Joan[34] demonstrated an IoT-based water management system that measures the storage tank's water level and water quality. The proposed system employs a web application to control all the city's water storage tanks and provide real-time water level and quality monitoring. It includes sensors for sensing data, a microprocessor for processing the data, and platform tools for storing and displaying it. Water levels are measured by the ultrasonic sensor, while water quality is measured by the conductivity sensor, turbidity sensor, and PH sensor. The data from the control unit is sent to Carriots, an IoT platform that saves and analyzes the data stream in JavaScript Object Notation (JSON) format. It also has an alert system that notifies the authority when the water level rises above a specified threshold or when the water quality deteriorates.

Das et al. [35] present an IoT-based Water Quality Monitoring System that monitors water quality and sends out alerts via the GSM module. Several water parameters are measured using a Zigbee module to transfer data to the microcontroller and a GSM module to send data to a smartphone or PC.

Shah [36] presents a model for smart water distribution which monitors water quality using a pH sensor, connectivity sensor, and temperature sensor. The values of the sensors are uploaded to the cloud by the Raspberry PI controller through the internet surface at a random interval. These values can be tracked in real-time by location. Using this method, we can monitor each endpoint for proper water supply and take prompt action when a problem is diagnosed.

Get et al.[37] offer an automated system based on sequential logic constructed utilizing a flip flop and an automatic water level sensor with a controller. The water motor is controlled by the system, which detects the amount of water in the water tank.

After I understood the importance of a smart water grid and its usage at the time of a water crisis from the existing literature. I want to develop a system that will monitor the quality and the quantity of water supply at the same time. I will use a machine-learning algorithm to predict future water quality using FB prophet and Arima Sarima time series analysis. This project will help the government to control water wastage and get some tax benefits and create a trustable zone between customers, suppliers, and the government.

1.4 Motivation

Cleaning up our waterways relies heavily on monitoring the quality and quantity of the water flowing through them. It will give real-time and long-term data on the health and composition of streams, rivers, and lakes. It is impossible to overestimate the importance of monitoring changes in water quality: human health and livelihoods are dependent on clean, reliable water supplies.

With so much demand for water, countries all over the world are grappling with water shortages and maintaining control over its use, cleanliness, and protection.

Many academics are working to find answers to the water crisis by monitoring individual parameters such as pH, temperature, pressure, flow, and so on. They were unable to implement a procedure that allows for simultaneous monitoring of quality and quantity, and some of their systems were unsuitable for real-time monitoring. I want to create and implement a system that can simultaneously monitor real-time quality and quantity monitoring so that individuals can use real-time by creating an account and receiving their water status and billing system. Because the usage of water is increasing every day, resulting in a water crisis, emerging countries will be forced to spend money on water and regulate it. As a result of this implementation, the government or policymakers will be able to regulate water quality and create a charging system for it, allowing them to readily track their taxi percentage.

1.5 Objectives

To obtain quantitative and qualitative information on home water for biological, physical, and chemical usable amounts per household, government cost savings, and to establish a genuine partnership between government, supplier, and customers.

The following are the project's primary goals:

1. To keep track of the quality and quantity of water in the house

2. To be aware of the safety of the household's drinkable water

3. Calculate the amount of water consumed

4. To forecast the future situation of today's home drinking water

5. To collect data about water's physical, chemical, and biological properties in numerical form

1.6 Limitation of the sensors

To identify domestic water quality monitoring there many parameters which are necessary to use it as shown in Table2, for my case, some of these important parameters are missing due to the unavailability of the commercial market or their price was unfordable and more expensive, for this reason, I didn’t use some most necessary parameters such as arsenic, calcium, chlorine, fortunately, for their commercial availability and affordable price I use some other parameters which also important for the quality of water monitoring such as PH.

1.7 Thesis structure overview

Chapter1, I discussed the amount of water that the human body contains, the amount of water that is required to drink per day, and cited some verses from the Holy Qur'an where Allah says that everything is created by water, I also discuss the importance of accessing clean water to human health and the impact (different water borne diseases) that may human doesn't get enough clean water, which is not possible these days due to the water clogging, I also talk about source of water contamination and their types. I identified some researchers that have worked on this subject, as well as the parameters they assess, platforms, modules, and other methodologies they employ, and the gaps between their findings and mine in the literature review. I have looked at the primary differences between my system and older systems, the extra work modifications I made, the benefits of my implementation, and so on as incentives. The necessity of my implementation was discussed, as well as strategies to alleviate the water problem, obtain drinkable water, and establish a trustworthy community.

The establishment, modules, and operating principles of my system will be detailed in Chapter 2, as well as distinct hardware components with their explanations, circuit connections, and real-world application.

In Chapter 3, I discussed the type of internet network, the type of server I used, who owns this server, how to establish this server, data storage procedures, and data extraction methods.

The theory of machine learning algorithms, regressions, particularly linear and logistic regressions, error calculation, FB Prophet and its calculation technique, and SARIMA ARIMA and its calculation procedure are all covered in Chapter 4.

Chapter 5 covered experimental data analysis topics such as data source, data collection technique, data analysis by Fb Prophet with results, data analysis by SARIMA ARIMA with results, and so on. Fb Prophet and SARIMA ARIMA Result Comparison. Chapter 6 is outlined in the system's conclusion and future work.

Chapter 2

Hardware Component and Methodology

2.1 Introduction

This Chapter will discuss the hardware and the working principle of the system, the functional block diagram will show the sensors collect data from the environment (water storage tank) and transfer these data to the server to analyze it and notify the concern. This chapter will also present individual sensor work with experimental figures. The last part of this chapter has shown the connection of sensors and hardware implementation before and after their connection respectively.

2.2 Functional Block Diagram

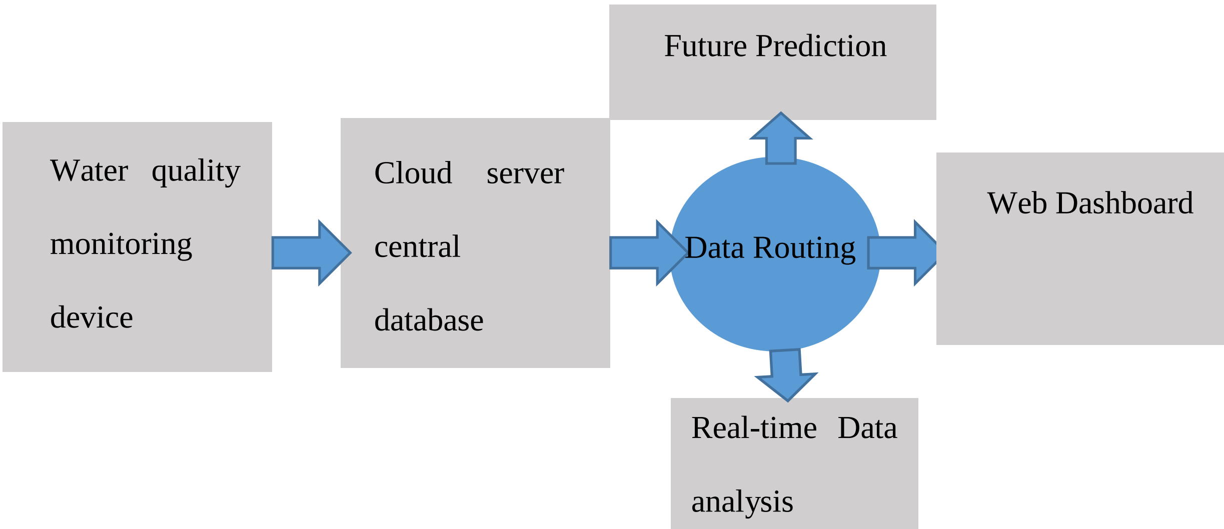

Fig2. 1: Functional Block Diagram

The above Fig illustrates the functional block diagram of the system, this functional block contains five blocks with their processing block, the first block is water quality monitoring devices, this block contains different sensors such as PH, Turbidity, Temperature and so on. Collect the water parameters, this Wi-Fi module gather data from different sensors and transfer to the cloud server which comes to second block, this block cloud server is used to store the data and visualize it, after storage the data it become for analyzing their future prediction and calculating different error values. The Web Dashboard block is where the user visualizes and monitor the data and how the process is going on. To store the data and visualize it, it is used Thingspeak as a cloud server.

2.3 Hardware components

A variety of hardware components are required to complete this suggested system, which is listed and discussed Table2

Table 3:Hardware Components Description

Component Name	*Purpose*
Esp32 Wi-Fi Module	*To collect the data from the environment gather it and transfer it to server (Thingspeak)*
PH Sensor	*It is an indicator used to measure the acidity, normality and alkalinity of water*
Turbidity	*It is an indicator used to find the amount of suspended sediment in water such as clay (composed of silicates metals like aluminum) and silt (composed of silicate minerals like silicon or oxygen)*
TDS	*It is an indicator used to find the amount of total dissolved substance n water such as inorganic salts (Calcium, Potassium, Magnesium, Zink, Copper and Sodium) and some of organic compound like (Carbonate, Bicarbonate, Nitrates, sulfates and chlorides)*
Color	*It is used for water quality monitoring to provide evidence that there is some of contamination*
Temperature	*it is used Because of its influence of water chemistry; the rate of chemical reactions generally increases at higher temperature*
Volume sensor	*It is used for water quantity monitoring to gauge and manage water levels in a water tank*
Flow sensor	*It is used for water quantity monitoring used to measure the amount of water that residential building use*

Volume and flow sensor are mostly related to water quantity part, which I do not focus much about it, for this system was mostly focusing on water quality monitoring.

For more details about the sensors has been discussed below

2.3.1 PH sensor

The pH scale indicates how acidic or basic a solution is. The range is from 0 to 14, with 7 representing the middle ground (Neutral). A pH reading of Less than 7 denotes acidity, whereas a pH reading higher than 7 indicates the presence of a base. In reality, pH is just a measurement of the proportion of free hydrogen ions to hydroxyl ions that are present in the water. Water that has a greater number of free hydrogen ions is considered to be acidic, whereas water that contains a greater number of free hydroxyl ions is considered to be basic.

Each number denotes a change in the water's acidity or basicity that is 10 times greater than the previous number. The difference in acidity between water with a pH of five and water with a pH of six is that water with a pH of five is ten times more acidic[38].

pH ranges from 0 to 14, with 7 being neutral. PH less than 7 is acidic while PH greater than 7 is alkaline (basic) as shown in Fig2.3, The Fig has also shown that the drinkable water is in the range of 6.5-8.5.

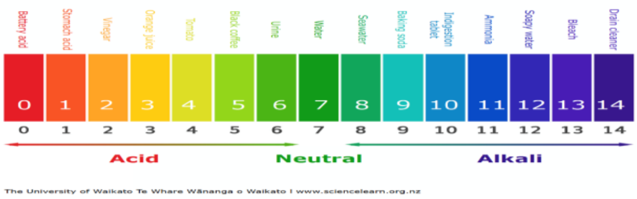

Fig2. 2:PH Scale [38]

Fig2. 3: Waterproof PH Sensor

2.3.2 Temperature Sensor

We use various things in our daily lives that may be necessary to continue our lives safely, but we don't know their temperature because it's not readable, or we don't have access to their reading, or we don't know how to read it, so we can overcome this challenge with the help of advanced technologies, particularly the Internet of Things (IoT). By receiving Notifications, these new technologies enable us to be aware of everything connected to our daily lives. In my case, I have been looking at the temperature of the water in the household, which is usually between 20 and 25 degrees Celsius. To pass the inspection, we'll need a waterproof temperature sensor that can be submerged in water and provides us with a probe. This sensor measures the degree of hotness or coolness and transforms it into a readable unit via an electrical signal[39].

Fig2. 4: Waterproof Temperature Sensor

2.3.3 Pressure Sensor

Pressure is a type of force that is delivered at a right angle to the outside of an object and dispersed over a unit area. It is crucial to understand pressure because it is one of the most important aspects of our daily lives. It may be used to monitor human blood pressure and other necessities. It is easy to update and monitor the pressure of vital objects thanks to new technologies. In my situation, I employ a pressure sensor to keep track of the amount of water in the house[40]. I created a sharp hollow outside the tank and then applied the sharp side of the saline tube to a pressure sensor that had another side of the saline.

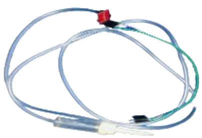

Fig2. 5: Water Pressure Sensor with Saline tube

2.3.4 Flow Sensor

Flow rate refers to the amount of water or other liquid materials that flow through an inlet or outlet. Humans may encounter such situations in their daily lives. Water for the home or industry, oil for automobiles or industries, and urine in hospitals for diabetic patients require observation or keeping an eye on things, and we can't be able to know our condition unless we have an appropriate and satisfactory sensor with modern technology. We can monitor and be alert without putting too much pressure on ourselves thanks to flowing sensors and advanced technology[41].

I observed an online server while working on water flow for the residence. Plastic covers the intake and outlet valves of the water flow sensor, which contains a hall effect sensor, turbine wheel, and magnet. The sensor's intake is connected to a water tank tap, and the sensor's exit is connected to a pipe. Water goes in through the inlet and out through the outlet from the tap. During this procedure, the turbine will begin to rotate, allowing water to flow. The speed of the water flowing through this pipe is measured by the sensor. This will establish a genuine connection between water providers and their customers. It will also help to alleviate the water shortage because users will not be able to use water beyond their limits.

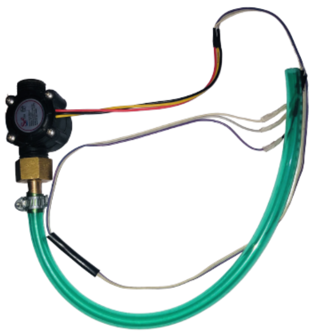

Fig2. 6: Water Flow Sensor with pipe

2.3.5 Color sensor

The color of the material is detected using a color sensor; white light is made up of three separate colors with various wavelengths. These three sensors are combined to create various color hues. When white light strikes any surface, some wavelengths of light are absorbed while others are reflected depending on the material qualities of the surface.

Fig2. 7: Color Sensor

2.3.6 Turbidity Sensor

The human eye has limitations; there are some things that it cannot see that are potentially harmful to human health. Microorganisms are unseen entities that can only be seen with the use of a microscope, which is not always ideal. We can observe the state of our necessities before we use them thanks to new technologies. This improves the healthcare system's safety, security, and trustworthiness. The light dispersed by a turbidity sensor analysis suspended solids in water. It determines the density of particles in water. It observes by sending light beams into the water, with the light detector positioned at a right angle to the light source. The amount of light reflected in it is then calculated[42].

The amount of reflected light is used to calculate the density of particles in the water. In my case, I use a turbidity sensor to check the water quality in the residence.

Fig2. 8: Turbidity Sensor

2.3.7 Total Dissolved Solid (TDS)

Total dissolved solids (TDS) are the number of dissolved minerals, metals, organic material, and salts in water, given in mg/L. It affects water quality and cleanliness, especially in water purification systems.

All living things, as well as anything that consumes or utilizes water, are impacted when total dissolved solids levels rise. As a result, it is necessary to measure it to guarantee the performance in industrial environments including pipelines, valves, and other equipment to the quality of the drinking water.

In addition to these, total dissolvable solids (TDS) can be derived from rocks as well as air that contains nitrogen, sulfur, calcium bicarbonate, and other types of mineral compounds. As water travels through the pipes that are used to bring water to consumers, it has the potential to pick up copper, lead, and other metals along the way. The ability of a water purification system to remove TDS may become less effective with time. Because of this, the quality of the membranes and filters should be monitored, and they should be replaced if necessary.

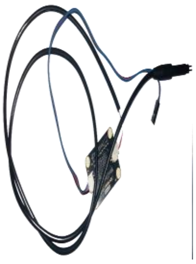

Fig2. 9: TDS Waterproof Sensor

2.3.8 Ultrasonic Level Sensor (Volume Sensor)

The challenges that trouble our ordinary routine, particularly those at home, are developing day by day because of a lack of awareness, leading to social, psychological, and financial insecurity. To solve this, it's crucial to keep track of all the essentials for human survival. This is easier these days because we may take advantage of the new modern world by checking even if we are unable to be present at the appropriate time. In my case, I am here to keep an eye on how much water is left in the house tank. Users can keep an eye on things via the internet. Ultrasonic sensors are used to help with this. Ultrasonic sensors monitor the level of any media, liquid or solid, using high-frequency ultrasonic waves. The sensor transmitter is oriented downward and placed on top of the tank. It transmits the waves and calculates the time it takes for the return signal from the water to reach the sensor. This will assist both the user and the administrator in determining how much water is remaining in the tank[43].

Ultrasonic waves travel at a significantly faster rate than audible waves. It is divided into two portions, each of which measures the distance between the sensor and the item, The transmitter sends sound waves to the item, and the receiver receives waves from the object, making it easy to compute the time it takes to complete this operation, we must apply the below-given formula to compute the distance between the sensor and the object[43].

$$.Distance = \frac{1}{2} \times time \times speed\ of\ sound \qquad (1)$$

Fig2. 10: Ultrasonic Level Sensor

2.3.9 I2C display

Industrial managers, students, waste management staff, and others may monitor their job even if they are not present. This is due to advanced technology, which may come at a large cost. Fortunately, if we don't have that much money, we can utilize other transceiver modules such as RF or NRF24L01. These will allow you to monitor your employees without having to use the internet, but because we don't have a web server, we may not be able to see what they're saying. To overcome this, we'll need to use an advanced electronic display device called an LCD, which uses a flat display as its primary mode of operation. It has 16raws and 2 columns in it. We converted the normal LCD into an I2C display in our instance because it requires a lot of cables and we wanted to save some wiring space. The I2C display operates on a serial data bus, whereas the typical LCD operates on a parallel data bus. I2C displays only have four pins to connect to the outside world[44].

Fig2. 11: I2C Display

2.3.10 Esp32 Wi-Fi

The ESP32 can connect to different networks when configured as a Wi-Fi station (like your router). In this case, the router assigns your ESP board a unique IP address. By referring to the ESP's unique IP address, you can communicate with the ESP using other devices (stations) that are also connected to the same network.

We may use the ESP32 board to request information from the internet, such as data from APIs because the router is connected to the internet (water data, for example), You can connect to your ESP32 board using any device with Wi-Fi capability without having to connect to your network if you set it up as an access point. When you configure the ESP32 as an access point, it creates its Wi-Fi network to which neighboring Wi-Fi devices (stations), such as a smartphone or computer, can connect. To control it, we don't need to be connected to a router [45]

Fig2.13[45] shows the accessing point of Esp32 with a smartphone and computer.

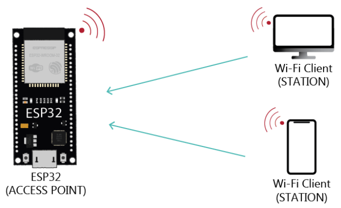

Fig2. 12: Esp32 Accessing with Smart Phone and Desktop [24]

Fig2. 13: Esp32 Wi-Fi Module

2.4 Schematic Diagram

Proteus software was used to create our schematic diagram. All sensors and other devices are connected to Esp32, which performs as the board/father of connection. The connection is presented below.

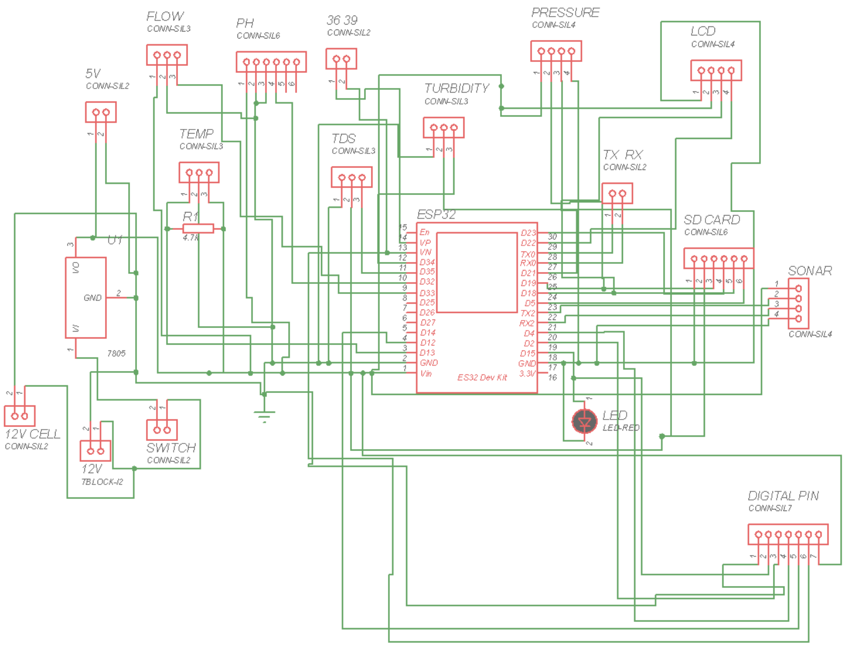

Fig2. 14: Schematic Diagram of The Proposed system

2.5 Hardware Implementation

Finding a strength or a means to manage our everyday lives is critical to being informed about health, education, climate change, the economy, and everything else that affects our environment. As a result, a chip with a variety of instruments that collect data from the environment consolidates it and then transfers it to the desired location where we can watch and receive notifications. Here, we develop a system to track the most crucial substance in our lives: water. Many environmental issues can result in a shortage of water, unsuited water for human life, or other living things in the environment. This tainted water has the potential to induce a variety of dangerous illnesses. To avoid this and to ensure the safety of the environment and its surroundings, we establish a positive and trusting connection between the government, the water supplier, and the community. This system has many sensors that collect data from water and transfer it to a server where it is stored for analysis and future prediction using machine algorithms. Here is my experimental figure, which begins before we connect the sensors to the board and continues after we connect them to the board and observe the water.

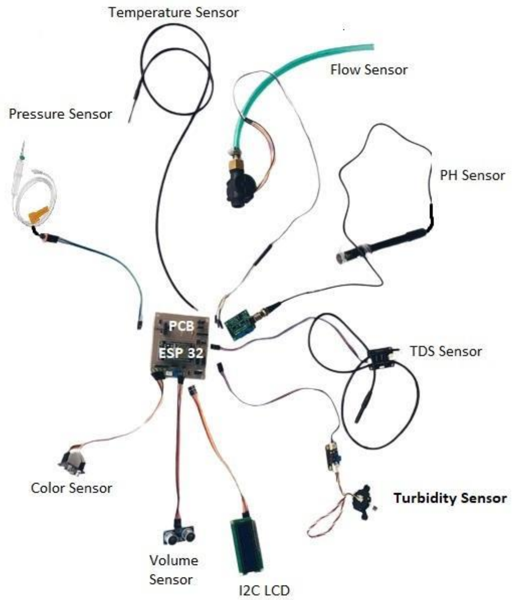

Fig2. 15: Hardware Component of the System before Connection

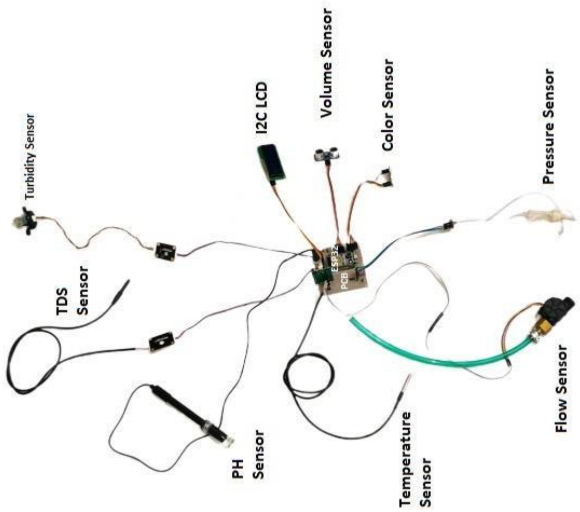

Fig2. 16: Hardware Component of the System After Connection

2.6 Conclusion

Finding a strategy to discover and control the water issues we confront necessitates establishing and maintaining confidence between water suppliers and their consumers, as well as benefiting the government in terms of taxation and water resources. This chapter covers how to construct this system, the tools needed to construct it, and how to monitor and observe it. Additionally, we illustrate in this chapter that different types of sensors are necessary to monitor water parameters, their working principle, advantages, and so on.

Chapter 3

Internet of thing (IoT) and Data Collection

3.1 Introduction

This chapter discusses the overview of IoT, the server Platform which I have used to store the data, how to create and implement this server, and the limitations of this server.

3.2 IoT Overview

The Internet of Things (IoT) is a stage in the growth of the Internet that builds a global infrastructure for human-machine communication. IoT stands for Internet of Things. Building the global infrastructure that will transform everything we do, from health care to manufacturing to agriculture and mining. The Internet of Things (IoT) will be promoted. The latest breakthroughs in the field of Artificial Intelligence require the requisite infrastructure (AI). In modern culture, the Internet of Things provides us with privacy, security, trust, and development.

An Internet of Things node is a piece of hardware with a sensor that transmits sensor data to users or other devices through the Internet. Industrial equipment, medical and mobile devices, wireless sensors, and other items are all covered by an IoT contract.

The connected smart city, smart industry, smart transportation, smart buildings, smart energy, smart manufacturing, smart environmental monitoring, smart life, smart health, smart food, and water monitoring system are the most essential examples of the Internet of Things[35],[46].

Figure 3.1 shows the IoT system's network architecture, which includes various sensors that gather data from the environment and send it to the gateway. The gateway should be a Wi-Fi network device that allows the data to be sent to the server. After that, it will be saved and evaluated, and users will be notified. A mobile app, dashboard, and other tools allow users to keep track of their data. informing them of their system's current state

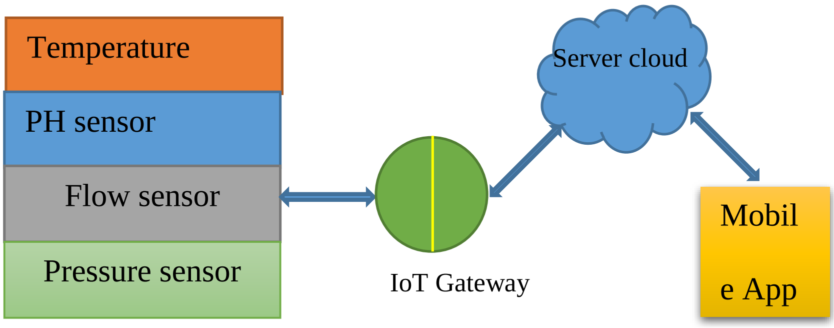

Fig3. 1: IoT Frame network with different sensors

In recent years, the (IoT)Internet of Things has grown in popularity dramatically. This is due to the fundamental characteristics of IoT that have attracted devoted customers: its ability to boost production, maintain (and increase) efficiency, and as well as provide quick results. You may change and move the way you transfer information or data from one device to another thanks to the Internet of Things. It's not only about transmitting information from a tablet to a phone. You can use IoT to give advice or data to your microwave, refrigerator, TV, and other devices.

Smart Home is of essential uses of the Internet of Things, Monitoring House related issues is necessary for all human mankind. Monitoring water and energy helps to save costs and resources. Water monitoring is essential because it discovers the availability of water inside the tank, and some sensors are responsible for notifying the remaining water and their safety[47].

An IoT is a trustworthy system and enables the management of people, equipment as well as transport. It builds good relationships with energy suppliers, government, and societies. For the purpose IoT is used to implement chips inside the human body then doctors can monitor a patient’s situation even if it’s outside of the health center. Likewise, smart cities’ IoT systems use different parameters such as garbage management, people management, traffic management, and so on[48].

The Internet of Things is a network of assigned devices called “things” these devices can transfer data over the internet without taking help from mankind. For connecting and exchanging data with other devices and systems across the internet, these devices are inserted with software, sensors, or other types of technologies. It contains two major components which are IoT devices and IoT gateways. In another word the internet of Things or IoT is a system of connecting devices, computers and digital machines with unique identifiers which are used to gather and exchange real-time data over the internet. In addition, the internet of things devices can be used for monitoring the health centers which can be used for monitoring the patients, monitoring the garbage, and so on, and educational centers, where it can be used to monitor students, faculty/staff members’ health, garbage monitoring, and any other related staff. And for the environment, it can be used to monitor weather, rm, and climate, it supports smart work and more control[49]. In my case, I was monitoring household necessities, particularly water quality quantity in real-time. We are using different sensors which collect data from water and then the Esp32 wi-fi module which acts as an IoT device to collect and transfer to the Thingspeak which acts as an IoT gateway.

3.3 Thingspeak

Thingspeak is an Internet of Things (IoT) analytics platform that is hosted in the cloud and enables users to collect, visualize, and investigate actual data streams. Thingspeak creates real-time visualizations of the data that is posted to Thingspeak. Because Thingspeak is capable of running MATLAB code, you will have the ability to assess and analyze data in real-time as it is being collected. Thingspeak is often utilized in Internet of Things (IoT) systems, particularly those that call for analytics at the prototype and proof of concept stages. IoT solutions have been created for a variety of vertical applications, including environmental monitoring and control, health monitoring, vehicle fleet monitoring, industrial monitoring and control, and home automation[50]. These are only some of the examples of these types of applications.

Fig3.2 can be used to describe numerous IoT systems at a high level. It is necessary to have MathWorks to use and sign up for the ThingSpeak account.

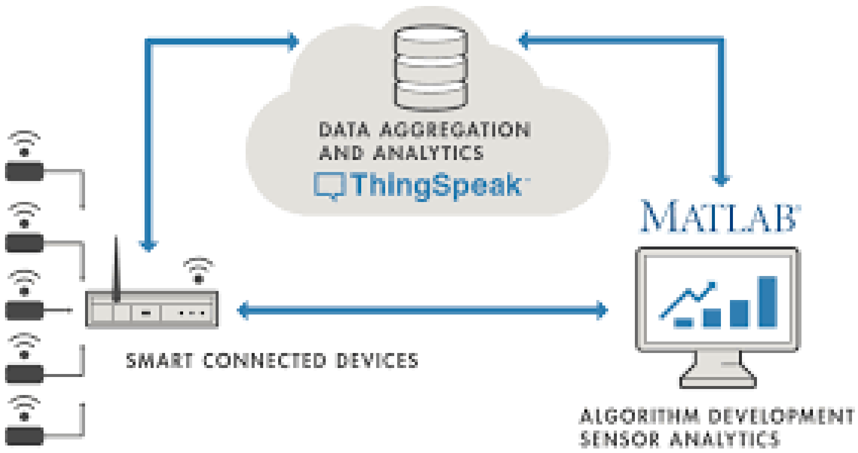

Fig3. 2: IoT Platform Network[51]

3.4 Implementation&Esp32

Here two major components of the internet of things are IoT devices and IoT gateways. In our case, we use Esp32 as an IoT device while Thingspeak is a gateway. To succeed in your goal these two components should be unable to communicate with each other. To do so we need to flow some necessary steps

Thingspeak sign up and channel creating

Engineers and scientists across the world rely on their products to accelerate growth, innovation, and exploration, thus it's important to have a MathWorks account to evaluate the data. This will allow you to connect to the Thingspeak server, which can sign in using your MathWorks Account. Here register for a MathWorks account.

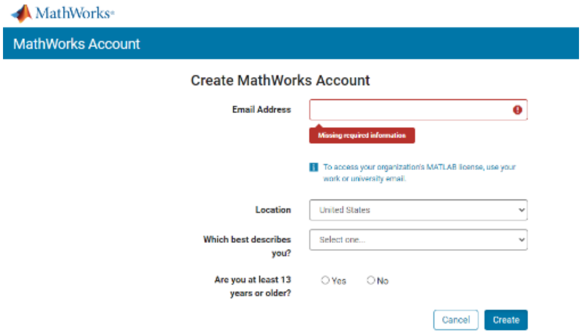

Fig3. 3: MathWorks Account creation

After creating MathWorks sign in to the creating MathWorks account, it will provide a Thingspeak account that works as a server and is used to store and manage data from the environment.

Fig3.4 shows where & how to sign in to the Thingspeak account using the MathWorks account. Use a valid email address with a valid password to sign in to Thingspeak.

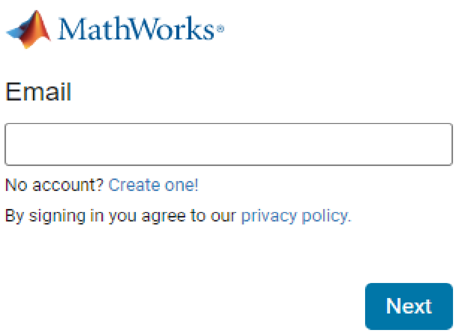

Fig3. 4: MathWorks/ThingSpeak login Window

After successfully signing in to the Thingspeak there will be a window that contains different options.

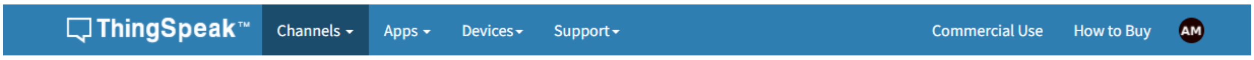

Fig3. 5: Main Window of Thingspeak

After you select a channel there may be seen many options as shown Fig3.6without confusion, select the "my channels" option.

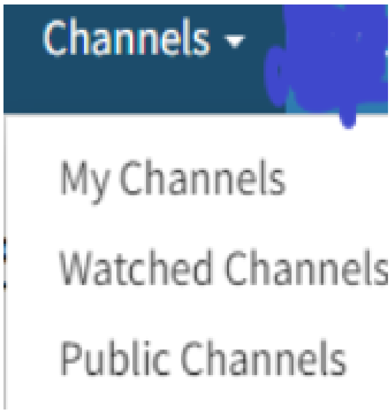

Fig3. 6: Thingspeak Channel Windows [32]

after clicking my Channel, click New Channel to create a new Thingspeak channel, there, fill up the name of the channel, channel description, and field name. field serial number and Channel will be provided from the server. The different channels have different channel IDs. To write the field name just click the given space and it will allow you to write.

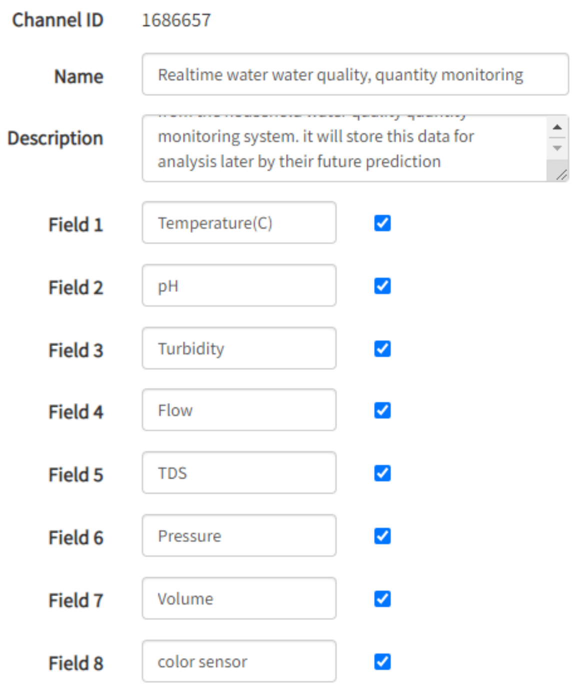

Fig3. 7: Necessary New Channel Filled-up Information

Click the save button, after fill-up all necessary information.

Fig3. 8: Channel saving

Saving the channel transfer me to the main window where all the channel information has been shown, it contains

The name of the channel, channel components, and channel status. Channel status is based on the date it was made, the last time the data, and the number of data entries. Channel settings are based on channel information it is used when it needs to edit the channel.

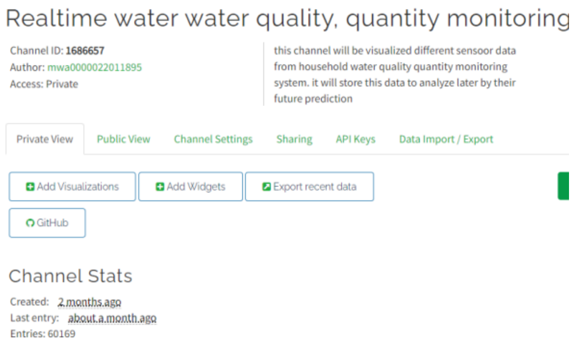

Fig3. 9: Channel Framework

After finishing the channel creation, it is necessary to connect the Wi-Fi device to the server. to do some it should install Arduino IDE software with its all-important libraries. Esp32 is used as an IoT device and Thingspeak as a server. To connect these two, it prepares and gets the below code from Arduino.

Libraries

```
#include <Wire.h>
#include <EEPROM.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include "HX710B.h"
#include <LiquidCrystal_I2C.h>
```

Wi-Fi Connectivity

```
const char* password = "Ahmed123";
```

Domain Name with full URL Path

Service API Key

```
String apiKey = "JOVGRYWNC8CYJCYD ";
```

Wi-Fi setup

```
  WiFi.begin(ssid, password);
 Serial.println("Connecting");
 while(WiFi.status() != WL_CONNECTED) {
   delay(500);
   Serial.print(".");
 }
 Serial.println("");
 Serial.print("Connected to WiFi network with IP Address: ");
```

The API key is from the Thingspeak channel at the Write API Key baton showing fifty columns of fig3.9. This will help to send data to a specific channel.

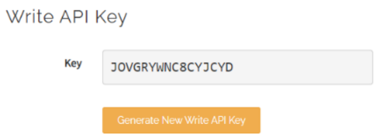

Fig3. 10: Write API key service of Thingspeak

After the connection process finishes the IoT device (Esp32) starts to collect data from the sensor/environment and transfer it to the server (ThingSpeak). When data reaches the server, it will organize as plots with names that are filled in fig3.7. the result plots are as below

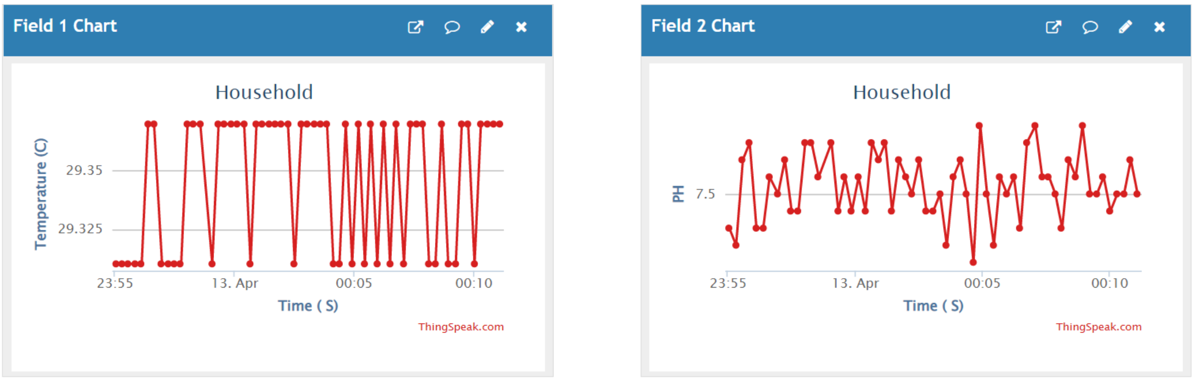

Fig3. 11: Household storage data of temperature and Ph raw data Graph from Thingspeak

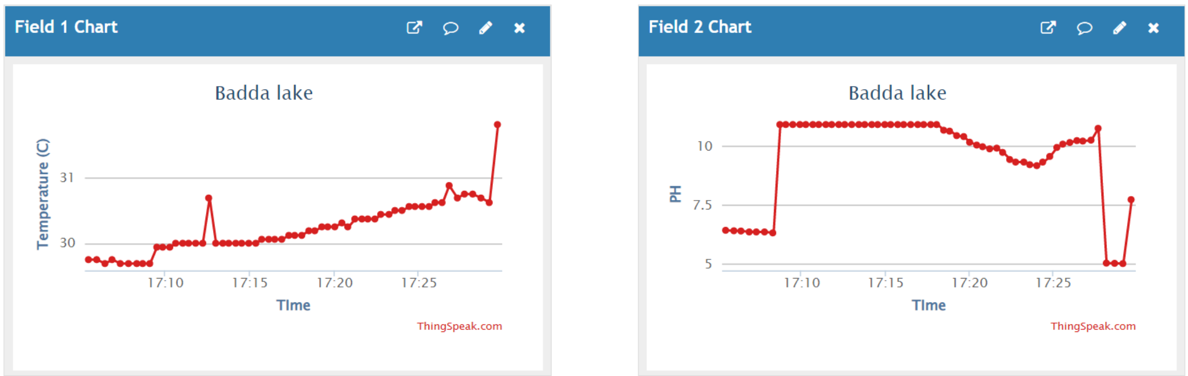

Fig3. 12: Badda storage data of temperature and Ph raw data Graph from Thingspeak

The above two Figures shows the data stores in the ThingSpeak server, the Fig3.11 represent storage data from household while Fig3.12 represents the storage data from one of Bangladesh lakes (Badda lakes). due to the time limitation, the storage data of lake is less than the storage data of the household.

3.5 Limitation of Thingspeak

As a result, nothing is perfect. The ThingSpeak server may have various limitations, such as those listed below.

1. The number of channels that can be created under one account is limited to four.
2. It also has a limited number of fields per channel, as seen in fig3.7, allowing just 8 fields per channel.
3. The number of messages that can be sent is limited to 300 million.
4. The repercussions of any errors or faults can be dire.
5. It's a little difficult to get used to the Use Interference (UI) at first.

3.6 Conclusion

The internet of things or IoT is a method of collecting data from various physical devices and transmitting it over the internet. To accomplish so, two primary components are required: an IoT device and an IoT gateway. In this case, Esp32 serves as an IoT device, while Thingspeak serves as a gateway. The ESP32 Wi-Fi module collects data from physical devices and transmits it to a server. The server is where we store and monitor data from physical devices, and while Thingspeak appears to be a nice server, it does have limits, such as the inability to use lifetime grants and the necessity to purchase in some circumstances. However, it has aided us in storing data from our water. We will examine the future state of these waters using these data and algorithms to predict the future. The analysis of their results and machine learning techniques will be detailed in the next chapter.

Chapter 4

Theory of Machine Learning Algorithm

4.1 Introduction

In this chapter, I have discussed in detail about machine learning algorithms, categories of machine learning, Regression, types of regression, and two methods of time series analysis Fb Prophet and sarima arima with their calculation procedures respectively.

4.2 Machine learning algorithms

The concept of "manual" is developing at an advanced level nowadays. Image recognition, surgery, and statistical arbitrage are intelligent examples of the benefits of machine learning algorithms. We live in a time of ongoing technological advancement, and we can see what the future holds by looking at how computers have evolved over time.

The accessibility of computer tools and processes is one of the most prominent components of this revolution. A new generation of data processing machines has been produced by data scientists in the last five years. They are effortlessly executing modern methodologies, achieving astonishing outcomes, and achieving success because of this. As there are so many algorithm names thrown around, it can be difficult to keep track of what each one is and where it fits. The first is a classification of algorithms based on how they learn, and the second is a collection of algorithms based on their similarities in form or function[52].

Machine learning algorithms are classified into four categories:

1. Supervised

2. Unsupervised

3. Semi-supervised learning

4. Reinforcement Learning

For supervised machine learning algorithms to work, they need to be supervised by a developer or programmer. The algorithm is given input and output data specified by the algorithm's developer. Moreover, data sets that have no labels will be used to train unsupervised machine learning algorithms. These algorithms are used when finding patterns, trends, or groupings in a dataset that aren't already known. Finally, semi-supervised machine learning is a combo of supervised and unsupervised methods, as the title indicates[53].

There is some link between the two algorithms. It is used along with incomplete or inaccurately labeled datasets. And lastly, through trial and error, reinforcement machine learning helps systems become better at whatever they're doing through trial and error. In a given situation, the model uses lessons learned in the past to choose the optimal course of action. Fig4.1 demonstrates types of machine learning algorithms

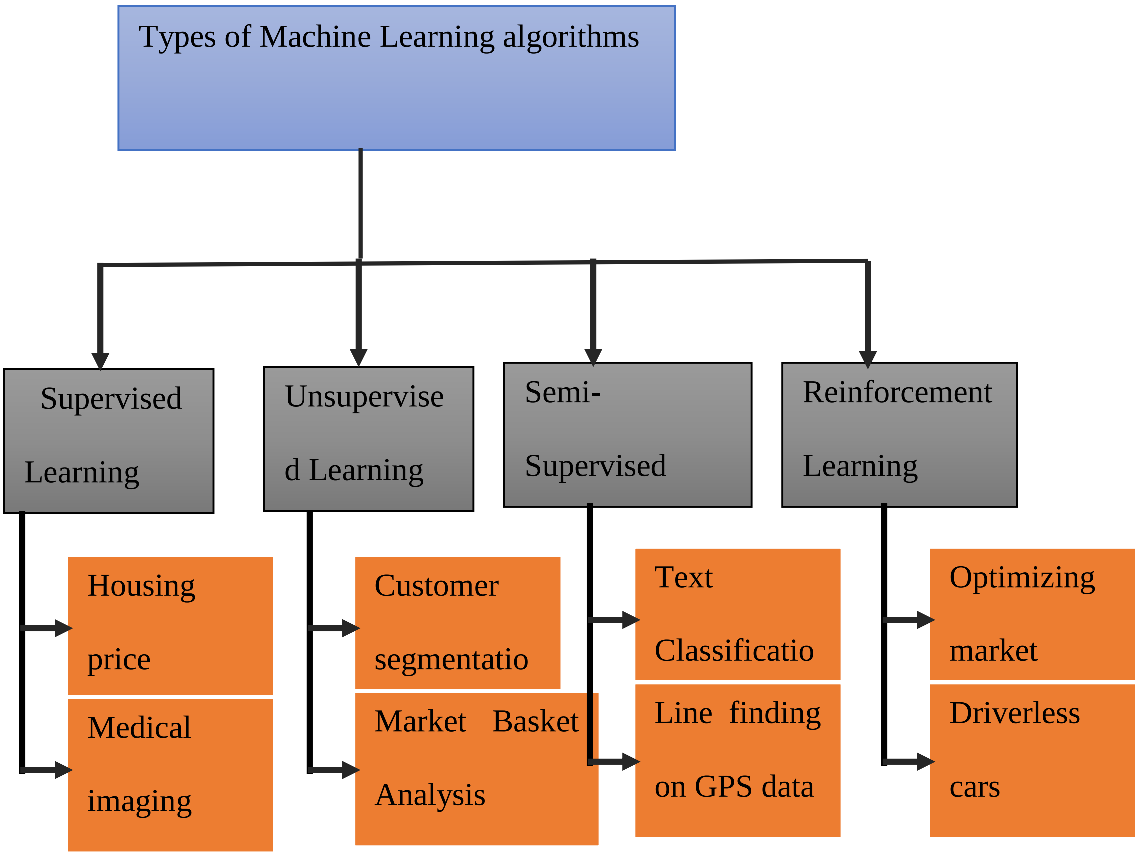

Fig4. 1: Types of Machine Learning[54]

4.3 Regression

One or more independent variables are used in regression analysis to represent the interaction of a dependent and an independent variable. The effect of an independent variable on the value of a dependent variable is investigated using regression analysis, which holds all other independent variables constant. Temperature, age, salary, price, and so on are all examples of continuous/real numbers that can be predicted by this procedure [55].

The most common method for discovering correlations between variables is regression, which can be used to predict an output variable using one or more predictor variables. For forecasting, time series modeling, and causal-effect relationships between variables, it is the most commonly utilized technique.

Regression analysis is another part of regression. The phrase "regression analysis" refers to a set of statistical procedures used to estimate the relationships between a dependent variable and one or more independent variables. It can be used to assess the strength of variables' correlations and forecast relationships. Fig4.2 illustrates branches of regression

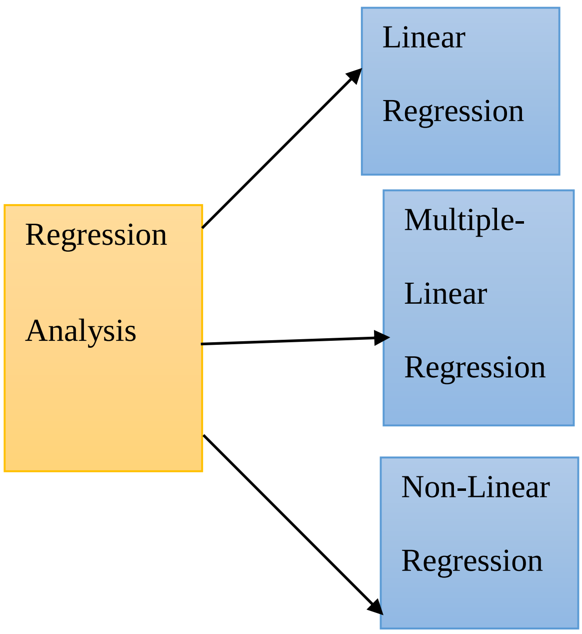

Fig4. 2: Branches of Machine Learning[56]

Regression analysis is a machine learning method for predicting the value of a continuous dependent variable (y) based on the values of one or more predictor variables (x). In a nutshell, the objective of a regression model is to develop a mathematical equation that expresses y as a function of the x variables.

4.3.1 Linear Regression

Linear regression is a straightforward and extensively used machine learning technique. It's a statistical technique for conducting predictive analysis. Linear regression is used to model continuous, real, or quantitative variables such as sales, salary, age, and product price.

To see how a dependent (y) variable and one or more independent (y) variables are related in a straight line, you can use linear regression. As a result, linear regression indicates the existence of a linear relationship, i.e., how the value of one item changes as the value of another thing changes [57].

A curved straight line is produced by the linear regression model, demonstrating the relationship between variables. Linear regression is shown in Fig4.3.

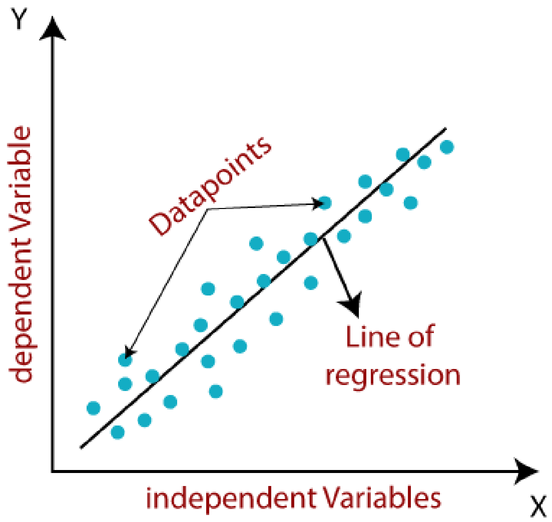

Fig4. 3: Graph of linear regression model [58]

$$y = a0 + a1X + \varepsilon \qquad (2)$$

The term y represents the dependent variable (Target Variable)

The word X represents an independent variable (predictor Variable)

a0 = the intercept of the line (Gives an additional degree of freedom)

a1 = Coefficient of linear regression (scale factor to each input value).

= unintentional error

The values of the x and y variables constitute training datasets for the representation of a Linear Regression model.

4.3.2 Logistic Regression

In the supervised learning algorithm, logistic regression is one of the most commonly used machine learning algorithms. It's an approach for measuring a categorical dependent variable from a collection of independent variables. Linear Regression and Logistic Regression are extremely similar in terms of usage strategy. To solve regression problems, linear regression is employed, while logistic regression is used to solve classification problems. Logistic Regression is shown in Fig4.4.

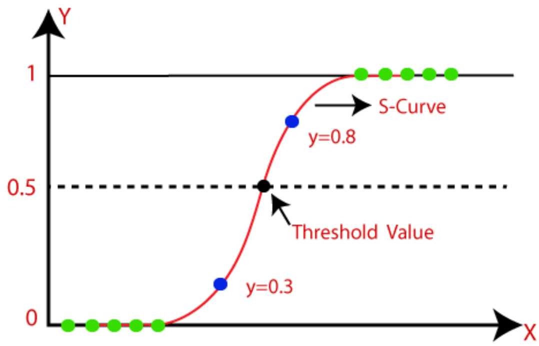

Fig4. 4: Graph of logistic Regression [59]

4.4 Error Calculation

An activity that is inaccurate or incorrect is defined as a mistake. In machine learning, an error is used to measure how effectively our model can predict both previously learned data and new, previously unknown data. Choose the optimal machine learning model for a specific dataset [60].

The act of discovering, observing, and diagnosing erroneous machine learning predictions is known as error calculation, and it helps us understand where the model performs well and where it does not. When it is stated that "the model accuracy is 90%," this may not be the case for all types of data, and there may be some input conditions with which the model does not perform well more frequently. So, now that we've looked at aggregate metrics, we'll go over how to enhance our models in more detail.

This is an example of a model that is good at recognizing dogs in an open environment but not so good in a dark room. This could, of course, be due to biased data. Error analysis can assist determine if this has an impact on the model's performance.

$$MAE \quad = \frac{\sum_{i=1}^{n} |yi - xi|}{n} \tag{3}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(yi - Yi)\text{^}2 \tag{4}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(xi - Xi)^2}{N}} \tag{5}$$

For RMSE xi= Actual observation time, Xi=Estimated time series, N=number of non-missing data

For MSE, yi= Observed values, Yi=Predicted values, n=number of data points

For MAE, yi=prediction, xi=true value, n=Total number of data points

4.5 Fb Prophet

To create time-series models, you can use a free algorithm called Facebook Prophet. It's built around a mix of old and new concepts. This is a fantastic approach to utilize if you need to model a time series with a lot of seasonality.

Prophet FB was developed by Facebook as an algorithm for predicting time series values in-house for a variety of commercial purposes. As a result, it was created expressly to forecast business time series. A Regressive Additive Framework illustrates the Prophet Prediction Models. The equation of the model is given in[61]:

$$y(t) = g(t) + h(t) + s(t) + et \qquad (6)$$

y(t)=Addictive Regressive Model

g(t)=Trend Factor

h(t)= Holiday Component

s(t)=Seasonality Component

et= error term

The explanation of each part is given below:

g(t): It is the trend, and the goal is to obtain the series' overall trend. For example, as more people join Facebook, the number of people who see ads is likely to rise.

s(t): It is the Seasonality part. The number of people who see the ads might also change depending on the time of year. People in the northern hemisphere are more likely to spend more time outside and less time in front of their computers during the summer. These seasonal fluctuations can be very different for different business time series, but they can still happen. The second part is a function that looks at seasonal trends.

h(t): The Holidays Section. When there are holidays that have a big effect on most business time series, we use the information to figure out when they are. Holidays can be different in different years, countries, and so on, so the model needs to know about them.

Random fluctuations are represented by the error term at which the model seems unable to explain. Normal distribution N (0, σ2) with values between 0 and unknown variance.

4.5.1 Calculation procedure using FB Prophet

The prophet is a time-series data forecasting process that uses an additive model to accommodate non-linear trends with yearly, monthly, and daily seasonality, as well as holiday impacts, in non-linear trends. It works well with time series that have a lot of seasonal variation and historical data from several seasons.

The FB prophet's calculation procedure provides two examples. The logistic growth model and the piecewise linear model are both growth models that show how things expand. You can specify a different model if you don't want Prophet to use a piecewise linear model as its default. As a result, a model should be carefully designed because it is dependent on a variety of factors. It depends on the size of the organization, how quickly it expands, and the type of business plan it employs. If the data to be projected is saturated and nonlinear, a logistic growth model is the best option. However, there are certain exceptions. A piece-wise linear model is preferable if the data is linear and has previously grown or contracted [62].

4.6 SARIMA ARIMA

ARIMA is an acronym that stands for Auto-Regressive integrated moving average. And it is an acronym. It's a group of models that can show a lot of different common patterns in time series data, like how quickly things move. It is very specific about how time-series data should be organized. As a result, it is very easy and powerful to use to make good time-series forecasts.

It's a more complex version of the Auto-Regressive Moving Average, with the added idea of integration, making it even more complex [63].

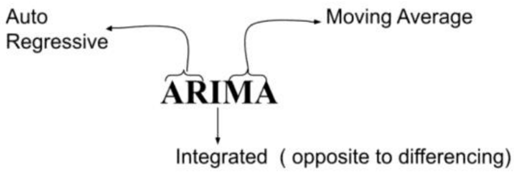

Fig4. 5: ARIMA MODEL [64]

The term "auto-regressive integrated moving average" can also be abbreviated to "ARIMA," which stands for "auto-regressive integrated moving average." Additionally, it is an abbreviation. It is a group of models that can reveal a range of common patterns in time series data, such as the speed at which things move. It outlines in a very particular fashion how the structure of time-series data should be organized. As a direct consequence of this, the process of producing accurate time-series forecasts is both straightforward and effective. In this version of the Auto Regressive Moving Average (ARMA), integration is added, making it a more involved and difficult method [37]. The ARIMA Model change consists of three components, as depicted in Fig4.5[38]. `AR is an abbreviation for autoregression. A model that exploits the dependent relationship between an observation and a set number of delayed observations.

I am an acronym for integrated. The procedure of comparing raw observations (by subtracting one from the previous time step) to stabilize a time series.

MA is an abbreviation for Moving Average. A recursive model utilizes the link between an observation and the residual error of a moving average model when applied to lag observations. many things have gone over the components and now it's time to put them all together to form the ARIMA model.

There are only three variables in ARIMA (p, q, d). Auto Regression gives you p, Moving Averages give you q, and Differencing gives you d. It doesn't matter what order of magnitude d is if it's differentiable. Three parameters, similar to K in K-NN, are hyper-parameters that must be tested. If d = 2, we'll use the yt" model instead of forecasting yt.

ARMA is now available for download (p, q). ARMA is the result of removing I (the integrating portion) from ARIMA. The AR coefficient is p, while q is the MA coefficient [65].

Seasonal changes in time series, on the other hand, can contain periodic models, allowing for more exact forecasting. SARIMA is a seasonal and non-seasonal aspect of the ARIMA model that allows periodic features to be captured. To uncover seasonal and other patterns, trends, and cycles, analyze data using a range of forecasting models. If seasonality is a prominent component of the series, consider models incorporating seasonal adjustments, such as the SARIMA model.

To design a SARIMA model, the trend and seasonal components of a series must be selected as hyperparameters. Three tendencies must be established.

They follow the ARIMA model, in particular:

p: The trend auto regression order

d: Trends are arranged in a different order.

q: Trend moving average order

When it comes to SARIMA in Python, there are three models to follow: define model, fir model, and produce a forecast. As shown in fig4.6

$$SARIMA\underbrace{(p,d,q)}_{non-seasonal}\underbrace{(P,D,Q)_m}_{seasonal}$$

Fig4. 6: SARIMA Equation [66]

4.6.1 Calculation procedure using SARIMA ARIMA

The SARIMA and ARIMA models are paired with a seasonal component. The SARIMA (p, d, q) x SARIMA (p, d, q) x SARIMA (p, d, q) x SARIMA (p, d, q) x SARIMA (p, d (P, D, Q, s). The variables p and seasonal P show how many autoregressive terms there are.

The ARIMA equation, on the other hand, leverages prior observations of time series to predict what will happen in the future. We assign a weight to each of the previous phrases, which can fluctuate depending on how recent they were. AR(x) in the ARIMA equation specifies that the ARIMA model, which is called ARIMA, will use x-lagged error components. ARIMA is a regression model based on auto-regression[67].

Chapter 5

Experimental Data Analysis Using Machine Learning

5.1 Introduction

Here, in this chapter, I discuss the analysis of my stored data from the Thingspeak server. This analysis of future prediction and error calculation occurred through two different methods which are Fb Prophet and sarima arima methods respectively. I analyzed the data using Fb prophet and calculated the errors of the data by following some procedures. After I get the Fb Prophet result, I also analyze the same data using sarima arima and then compare their outcome result and see which analysis method is better.

5.2 Data analysis using Fb Prophet

Facebook is an open-source prophet prophecy obtainable in python and R. prophesy data science duty that is a principle to several enterprises within an administration. For example, wide-ranging firms namely Facebook should take part in position arrangements to effectively assign meager resources and aim to mount to compute discharge parallel to standard.

The diversity of prognosticating issues establishes faith in a huge number of forecasts when they have been generated.

Prophet has been a clue section to upgrade fakebook's potential to generate a huge number of reliable prophecies used for managerial and even in outcome qualities[68].

Data observation can flow hourly, daily, or weekly with a minimum of one month of history, depending on time availability the more data stored the more accurate for this reason most scientists prefer a year. The main advantage of the prophet is that fabricating a reasonable, on-target prophecy prophet makes it much more straightforward. And customizable in techniques that are inherent to non-professionals. By selecting changepoints from data prophet detects changes in trends automatically and results in growing linear or logistic curves.

5.2.1 Data Source/Experiment Details

To investigate and analyze the data, it needs to have storage informality to retrieve this stored information, it requires a place to take it, and it also requires measuring the amount of time it takes. In this particular scenario, I was examining many aspects of domestic water. There, we had a total of eight distinct sensors, all of which could be categorized as either quality or quantity sensors. I employ pressure sensors, flow sensors, and volume level sensors to measure the quantity, while pH sensors, temperature sensors, turbidity sensors, and total dissolved solids sensors are used to measure the quality of the solution. These sensors are linked to the PCB board of the Esp32 microcontroller, and they gather data from the water tank. Once the data is collected, the Esp32 microcontroller receives it and sends it to the cloud server.

In my case, I was using a 35-liter RFL container to install the sensor board and take the water measurement. The area of this RFL container was 7.06m^{2,} while the total storage of the container was 35litter. The data collection period is about fourteen days (14 days) starting from 27. Mar.2022 to 12. Mar.2022, these 14 days the data has been continually collected, and the data from sensors has been combined and transferred to the server after ten-seconds secseconds0 second). For this period the total collected data was sixty hundred one thousand ninety-eight data (601069). The server stores this data as CSV, which is a comma-delimited text file, I download this file and make some changes such as field name, because it just gives serial field number by following Fig3.7, so I gave each serial number by respected field name. The first field name is Temperature as shown in Fig3.7, Second is PH and so on, after this process the data is analyzed and analyzed. Uploading data to the Google Collab which helps us to do analyzing and predictions. The data RMSE calculation and its future prediction needs to follow some steps which will be shown in 5.2.2 (Data Analysis by using FB Prophet).

Fig5.1 shows the RFl container that I use to store the water and install the different eight sensors. There as you seen Fig5.1, some sensors are placed inside the container (Temperature, PH, Turbidity and TDS) and other is placed the edge of the RFL container (Flow sensor and Pressure sensor) while the rest two sensor (Ultrasonic level sensor and color sensor) are placed at the top side of the container. All the sensors are waterproof except the Color sensor.

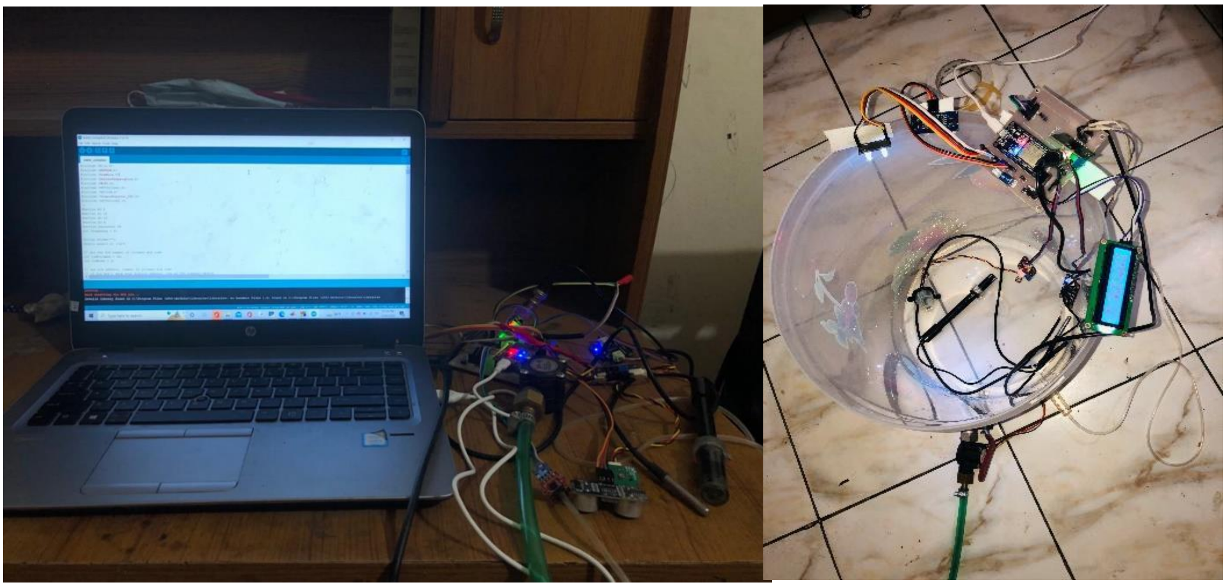

Fig5. 1: Experimental Process before and setup of 35litter of RFL storage container used to take water component measurement with

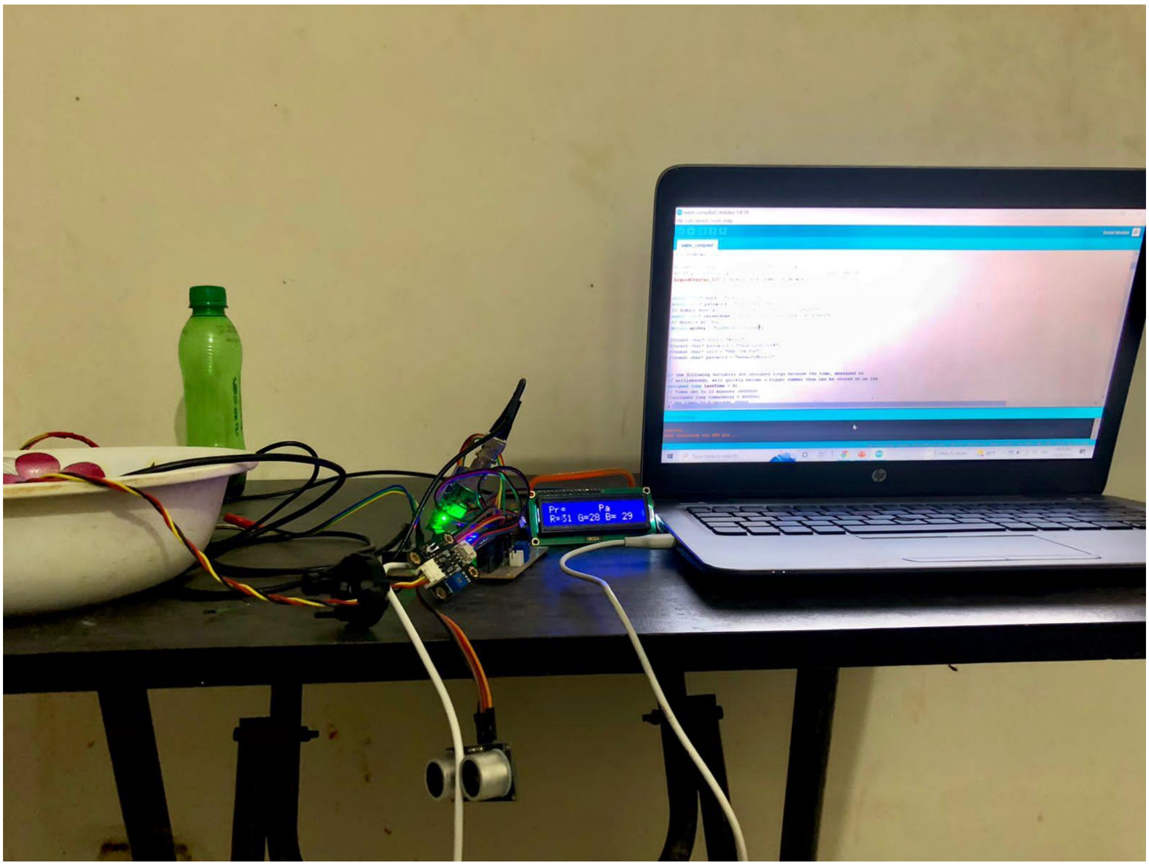

Fig5. 2: Performance of System validation data collection process

The \Fig explains the data collection process of water from Badda lake, the green bottle shows in the image is the container that I took water from the lake. The bowl I use to take the measurement of water parameters. It is easy to place all the sensors inside the bowl as it has flat opening space. I2C display used to see the parameter values manually. The PC is used to power the board as shown in Fig, the while Cable from PC is connected to ESP32 to power the circuit, on the other hand the pc is used to upload the program into Esp32 which allows to collect data and stored it in the server as CSV file.

Fig5. 3: comma-delimited text file from Thingspeak (CSVZ

5.2.2 Data analysis

In Fig5.3 shows the file from the server, and it carries all sensor data specifying the name of this file. The server saves the sensor with a field name, not a specific name while each field represents a specific sensor, so change the field name into the sensor name. After renaming the field names, I create a Google collab account which I can easily analyze my data. Google Collab helps me to write the analyzing code. The writing of the code has followed some rules and steps, I will discuss each step with its code. by upcoming sessions. Starting from installation and importing of libraries.

Step1. Installation and import dependencies

To install libraries and other necessary dependencies

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import random

import seaborn as sns

from fbprophet import Prophet
```

step2. Remove time_zone_label data from csv file

This step imports pandas as pd, removes data to their time_zone_label and generates a new file which can be analyzed with the required parameters.

```
#df=pd.read_csv('WW.csv')
#df.head()

import pandas as pd
def remove_timezone_label(data):
    pos = data.index('+00:00')
    # Slice data to get rid of the tz, then strip leading/trailing whit
espace
```

```
    data = (data[:pos]).strip()

    return data
if __name__ == '__main__':
    df=pd.read_csv("WW.csv")

    # the apply() function is used to carry out a function on the data.
    df["created_at"] = df["created_at"].apply(remove_timezone_label)

    df.to_csv("WW_NEW.csv", index=False)
```

step3. Reading data from csv

This step shows us the parameter's name with their data, this means the user can see there sensor name, carried data, issue of the data and data serial number.

```
Df=pd.read_csv('WW.csv')
df.head()
```

step4. Data execution and visualizing

The step helps us to see the individual/all sensor's related data, in my case, I apply an individual sensor to examine all related issues such as present data, future prediction, and error calculation. Additionally, this step executes only two columns. One is the created time (year, month, and date) and the other column is the sensor name with its data. This will help during prophet analysis as FbProphet only executes two columns.

```
#df=df.drop(['Temperature'],axis=1)
df=df.drop(['PH'],axis=1)
df=df.drop(['Turbidity'],axis=1)
df=df.drop(['Flow'],axis=1)
df=df.drop(['TDS'],axis=1)
df=df.drop(['Pressure'],axis=1)
df=df.drop(['Volume'],axis=1)
df=df.drop(['Colar'],axis=1)
```

```
df=df.drop(['entry_id'],axis=1)

df.head(200)

df.head()
```

step5. Data Description

This step will show the count of data, the number of sensors, the mean, the minimum, the standard deviation, and the percentages of the individual sensor.

```
df.describe()
```

Step6. Visualizing of raw data of the sensor

Before we calculate errors and analyze future predictions, we need to visualize the Raw data of the sensor. of the sensor's original data.

```
df.plot(figsize=(15, 6))
```

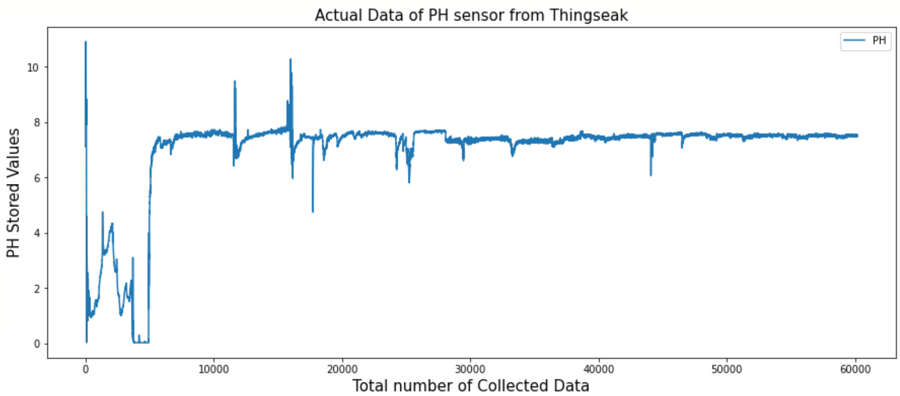

Fig5. 4: Raw data of PH sensor from Thingspeak plotted by using Fb Prophet

The above Fig illustrates how Thingspeak stores PH sensor data. The size of collected data is 60000+. The PH sensor varies from 0.02 which is an acidic state to 10 which is an alkaline state but all the time it gives 7 which is neutral. The drinkable water is in the range of 6.5 -8.5 and the average PH of my data is 7.47 which is better and more suitable to drink.

step7. Preparing the Prophet given dataset

The prophet always accepts a two-column data frame (ds, y). The ds (datestamp) column should be YYYY-MM-DD or YYYY-MM-DD HH:MM: SS. The y column is numeric and indicates the forecasted measurement.

df.columns = ['ds', 'y'] step

df.head()

step8. Import Prophet

To do forecasting we need to get some building functions, so this step will import these functions.

```
dir(Prophet)
```

step9. Initializing the model

It will initialize the model prophet and data columns

```
model = Prophet()
df.columns
```

step10. Model fitting

This step will step and disabling yearly seasonality

```
model.fit(df)
```

Model fitting is a measurement of how well a machine learning model generalizes to data that is comparable to the data on which it was trained. This data is called "fitting data." When it comes to accuracy, nothing beats a model that's been fine-tuned to perfection.

step11. Created Future dates (monthly)

This step I create a future prediction of 6 months (180 days)

```
future_dates = model.make_future_dataframe(periods=180)
```

step12. Create future prediction

This step will predict step10 which is future data and as it is a weekly

```
pred = model. predict(future_dates)
pred.head()
#pred = pred[['ds', 'yhat']]
#pred
model.plot(pred)
```

step13. Cross-validation

To calculate forecast inaccuracy from previous data, calculate forecast inaccuracy from previous data

```
from fbprophet.diagnostics import cross_validation
f_cv = cross_validation(model, initial='12 days', period='8 days', hori
zon='4 days')
#df_cv = cross_validation(model, initial='365 days', period='90 days',
horizon='180 days')
df_cv.head()
```

Step14. Matrix performance

This step is done by calculating errors especially for Root Mean Square Error (RMSE),

```
from fbprophet.diagnostics import performance metrics
df_p = performance_metrics(df_cv)
df_p.head()
```

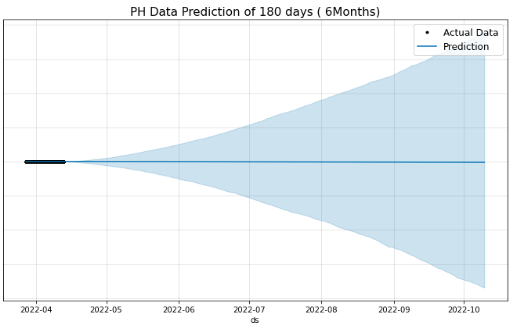

Fig5. 5: PH Model Prediction of six months which carries six hundred one thousand

5.2.3 Result & Discussion (FB Prophet)

I have been working on a household water monitoring system. I implement and build a system that contains eight sensors. I installed these sensors with RFL containers which I used as a water storage tank. The picture of the RFL container with installed sensors has been shown in Fig5.1 section 5.2.1. The data collection period is about fourteen days (14 days) starting from 27. Mar.2022 to 12. Mar.2022, these 14 days the data has been continually collected, and the data from sensors has been combined and transferred to the server after ten-seconds (10 second). For this period the total collected data was sixty hundred one thousand ninety-eight data (601069). of data collection and stored it into ThingSpeak, then I analyzed my data using Fb Prophet. Here are some steps I followed which I listed and explained in 5.2.2.

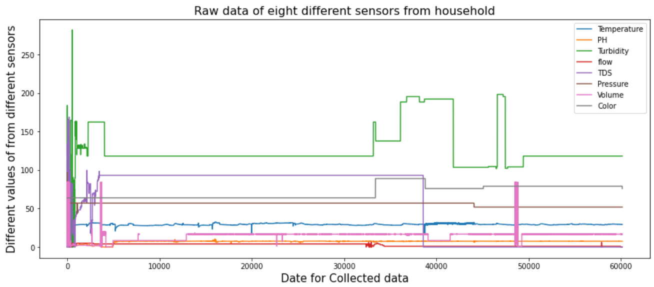

Fig5. 6: Visualization of raw data (domestic) of eight different sensor stored in Thingspeak server

Above Figure illustrates the actual data from eight different sensors which are stored at Thingspeak server, the X label of the figure represents the total number of collected data that are stored in the server while the Y label represents individual sensor data. Turbidity sensors have more value compared to others as shown in the figure.

Table 4: WHO standard value of the drinkable water compared to developed system

Sensor name	WHO standard values	Developed system values
PH (PH unit)	6.5-8.5	7.47
Temperature(C)	25-50C	29.25
Turbidity (NTU)	1.0	127.06
TDS (ppm)	50-150	57.92
Color (units)	15	74.34

In Table4 it shows that five sensors of a developed system, among these five sensors, three of them are within a range while the other two is above the range. This realize that the developed system is reliable for water quality monitoring. For further analysis, I calculated the error values using regression model. A regression model's performance can be evaluated and reported on using three error measures, which are:

a. Mean Squared Error (MSE).

b. Root Mean Squared Error (RMSE).

c. Mean Absolute Error (MAE)

The RMSE, MSE and MAE of all eight sensors are shown in Table2, and I took some RMSE error figures as the sensors have different errors. RSME of pH, Temperature and Flow data analyzed using Fb Prophet are shown Fig 5.6, Fig5. 7 and Fig5.8 respectively.

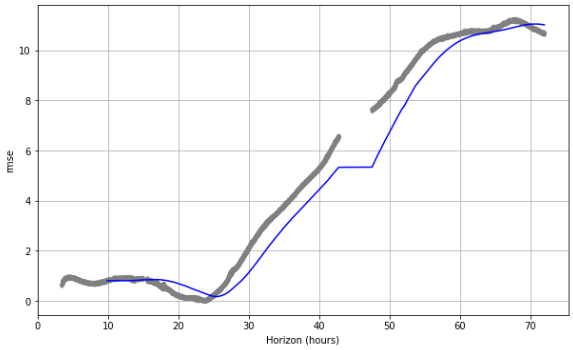

Fig5. 7: Root Mean Square of PH stored data using Fb Prophet

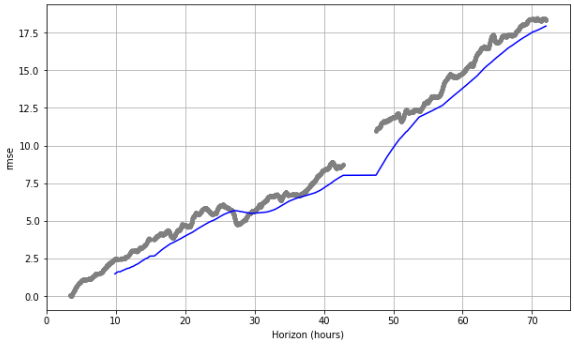

Fig5. 8: Root Mean Square of Temperature stored data using FB Prophet

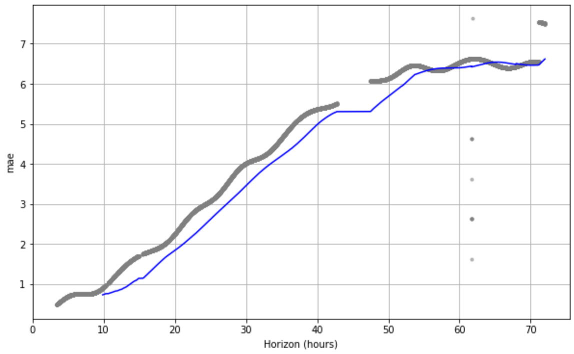

Fig5. 9: Root Mean Square error calculation for flow sensor data using Fb Prophet

With the help of step5 I calculate the sensor's descriptions and shows at table1 and using step10,

RMSE calculation is done and shown at table2.

Table 5: Description of eight sensors for domestic water Quality and Quantity Monitoring stored from Thingspeak

	Temperature	PH	Turbidity	Flow	TDS	Pressure	Volume	Color
Count	60169.000000	60169.000000	60169.000000	60169.000000	60169.000000	60169.000000	60169.000000	60169.000000
Mean	29.399024	6.974127	127.068228	2.678506	*57.920012*	55.647903	14.378616	*71.255879*
Std	1.074625	1.634548	26.378671	1.525473	44.539004	2.305145	6.803860	*8.*651564
Minimum	17.250000	0.020000	0.000000	0.000000	0.000000	43.660000	0.000000	64.000000
25%	28.810000	7.330000	118.260000	1.000000	0.000000	51.920000	8.410000	64.000000
50%	29.250000	7.470000	118.260000	4.000000	93.160000	57.010000	16.640000	64.000000
75%	30.000000	7.540000	118.260000	4.000000	93.160000	57.010000	16.710000	79.000000
Maximum	32.310000	10.910000	282.290000	6.000000	169.060000	96.750000	84.440000	89.000000

Due to some reason the sensors may not collect accurate data at all, we need to find an individual Root Mean Square Error (RMSE) of each sensor. Sensors may get different RSME values depending on their collected data. table the RMSE values of the individual sensor.

Table 6: RMSE, MSE and Mae values of Eight different sensors installed domestic Water Quantity and Quantity Monitoring calculated using Fb Prophet

Sensor Name	RMSE value	MSE values	Mae Value
Temperature	1.4774932	2.1829906	1.3493952
PH	0.802964	0.64479225	0.7982588
Turbidity	46.42883067	2154.252139	45.861548
Flow	0.734097	*0.5393182*	0.72976575
TDS	11.283986	127.666182	10.36507533
Pressure	2.8979555	8.4130155	2.70498425
Volume	1.5041518	187.502797	13.68939825
Color	5.968331	35.528954	5.552672

Table6 has been shown eight different sensors with their individual RMSE, MSE and MAE errors respectively. Among eight sensors PH and flow sensor are the only two sensors whose errors are less than one (error<1).

Furthermore, if we compare the RMSE error of these eight sensors we can see that some have low RMSE values while others have high, this is due to their nature, max-minimum collecting values. And may be their calibration process. The maximum values the sensor has the more RMSE the sensor has, this means if the sensor doesn’t store the value properly it may cause inaccurate values which may be higher than the normal values. For example, The PH of drinkable water is 6.5 to 8.5, which means the average drinkable water is seven point five (7.5). So here if the average of storage PH data is higher or lesser than the nominal average, then the RMSE value will become high. This may come when the sensor doesn’t collect the data properly or sometimes may collect sometimes may not, because each sensor needs its calibration process and there may be an external effect such as environmental aspects, time of collecting data, the data that the sensor collects the midday may not be the same the data collected early morning. On the other hand, the data collected all day may not be the same as the data collected at night. Due to this, the more the sensor collects different data the more RMSE value the sensor has. The more RSME sensor the worse the sensor is. This means the sensor performance depends on its RMSE. The good RMSE value is the range between (0-1). According to this, the PH sensor is the best performance sensor and its RSME is <1 as shown in table2 while the worst sensor performance is Turbidity its RSME is <10 as shown in table2. To improve sensor performance and reduce the RMSE values it must store more day's data. The more data we store/collect the more sensor performance becomes normal and gets accurate values and it will also be necessary to calibrate the sensor before it is used. The different sensors need different calibration processes.

5.2.3.1 Result comparison with literature (FB Prophet)

Here I will compare the RMSE errors of some existing errors and my work. I will consider those we have at least one common parameter, the first works I will look at those we have the same PH sensor and compare their errors to mine.

Table 7: Comparison between an existing and My PH sensor RMSE errors

RMSE of existing PH sensor	RMSE of my PH sensor
37.8[69] 0.120[70] 0.579[71]	0.802964

The Accuracy of the data set is always dependent on the RMSE error, which means the more RMSE error the data set has the less the accuracy the set data has, on the other hand, the RMSE error value should always be less than one (RMSE<1), the more data error greater than one, the less accurate the data has and the more data set error less than one, the more accurate the data set has. For this scenario, [52] and [54] have much more RMSE error compared to my data as well as [53]. This means my data is more accurate and better than theirs as my RMSE is much less than theirs.

There are some reasons behind getting high error values as listed below:

1. The calibration of the sensor: most of the sensors need to be calibrated before use, this means to get an accurate measurement of the data set the sensor should be calibrated.
2. The number of data sets: collection and storing of more data plays a vital role to reduce the RMSE error. This means the more data set, the fewer the error.
3. The place where data is taken: the source of the data is important, different data sources will give different data values. Suppose the water collected from the domestic is much cleaner than the water collected from the river. This means, the data set from the households are more accurate and less RMSE errors than the data set from the river.

[52] and [54] may have failed to meet at least one of the above reasons, which makes my data better than literature. On the other hand [53] has the lowest RMSE error value which makes their data better than mine. This is because their collection and stored data is more than mine.

5.3 Data analysis using Sarima Arima

Some of the most important challenges in statistics and data science have been related to predicting and time series. When data is sampled on a time-bound property such as days, months, and years, it transforms into a time series since this gives the data an implicit order by its very nature. The process of using historical data to make predictions about future values is called forecasting.

Both ARIMA and SARIMA are forecasting algorithms. ARIMA makes predictions about future values based on past values (autoregressive, moving average). While SARIMA makes use of historical data, it also considers patterns of seasonality. When it comes to complicated data sets that involve cycles, SARIMA is more powerful than ARIMA since it incorporates seasonality. Data observation can flow hourly, daily, or weekly with a minimum of one month of history, depending on time availability the more data stored the more accurate for this reason most scientists prefer a year. Data observation can flow hourly, daily, or weekly with a minimum of one month of history, depending on time availability the more data stored the more accurate for this reason most scientists prefer a year.

ARMA requires stationary time series. Stationarity means a time series is stable. If your time series isn't stable, utilize the Augmented Dickey-Fuller test with differencing.

SARIMA incorporates seasonality into ARIMA. Using seasonality in your time series forecast is crucial.

5.3.1 Data Source/Experiment Details

In this we use the same data source explained in 5.2.1, Here there may be some sensors that are not much good using Fbprophet forecasting so to check their performance again we use Sarima Arima. Then we compare the results from different forecasting methods.

5.3.2 Data analysis

Here we present some important steps with individual step codes; fortunately, the first steps are the same as 5.2.2. so, we will start with step4

Step4: Data Visualization

After we install the necessary libraries, remove the time zone and read the data with its shape. We plot the data to get an idea or examine whether the data is stationary or not. Time series data must be stationary to be modeled. When something is "stationary," it means its statistical qualities are relatively stable across time. The below represent all the first four steps.

```
import pandas as pd
```

```
df=pd.read_csv('PH.csv',parse_dates=True)
df=df.dropna()
print('Shape of data',df.shape)
df.head()


## Cleaning up the data
df.columns=["created_at","PH"]
df.head()
# Convert created_at into Datetime
df['created_at']=pd.to_datetime(df['created_at'])
df.set_index('created_at',inplace=True)
# Data ploting
df.plot(figsize=(15,6))
plt.title("Raw data for PH ",fontsize=12)
plt.ylabel("Different pH values of water",fontsize=10)
plt.xlabel("Date of collected data",fontsize=10)
plt.show()
```

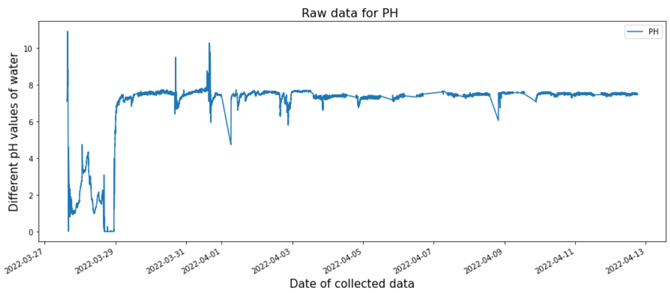

Fig5. 10: Household raw data of PH from Thingspeak using Sarima Arima model

The above Figure shows the PH sensor analyzed using Sarima arima. The X-axis represents the time of collecting data by showing a year, month, and then day, while the Y-axis shows the sensor detected values from different days. The highest detection value is 10 which represents alkaline while the lowest value shown is 0.02 as acidic. From 27. Mar till 29. Mar the sensor wasn't calibrated properly, for this reason the sensor doesn't store proper values in between these days. Fortunately, after I calibrated properly then the sensor collects and stores accurate values. The average showing graph is 7.54 which is neutral.

Step5. Testing for stationarity

```
from statsmodels.tsa.stattools import adfuller
test_result=adfuller(df['PH'])
def adfuller_test(PH):
    result=adfuller(PH)
    labels = ['ADF Test Statistic','p-
value','#Lags Used','Number of Observations Used']
    for value,label in zip(result,labels):
        print(label+' : '+str(value) )
    if result[1] <= 0.05:
        print("strong evidence against the null hypothesis(Ho), reject
the null hypothesis. Data has no unit root and is stationary")
    else:
print("weak evidence against null hypothesis, time series has a unit ro
ot, indicating it is non-stationary ")
adfuller_test(df['PH'])
```

The p-value is all that is needed to evaluate the test results. The following procedure can also be utilized: to make the data stationary $P<0.05$, if $P>0.05$ then the data is Stationary. But that won't be a problem, with help of step6 this will be solved.

Step6. Differencing

```
df['PHFirst Difference'] = df['PH'] - df['PH'].shift(1)
df['PH'].shift(1)
df['Seasonal First Difference']=df['PH']-df['PH'].shift(12)
df.head(14)
## Again test dickey fuller test
```

```
adfuller_test(df['Seasonal First Difference'].dropna())
```

Here I got that the value of p is 0.0, which is less than 0.05, and my data become stationary.

Now with help of below code the data stationary Figure has been shown in Fig5.11.

```
df['Seasonal First Difference'].plot()

plt.title("Seasonal difference for PH data", fontsize=16)

plt.ylabel("Different pH values of water", fontsize=15)

plt.xlabel("Date for Collected data", fontsize=15)
```

Step7. Arima model discussion

```
from pmdarima import auto_arima

stepwise_fit = auto_arima(df['PH'], trace=True,

suppress_warnings=True)

from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
```

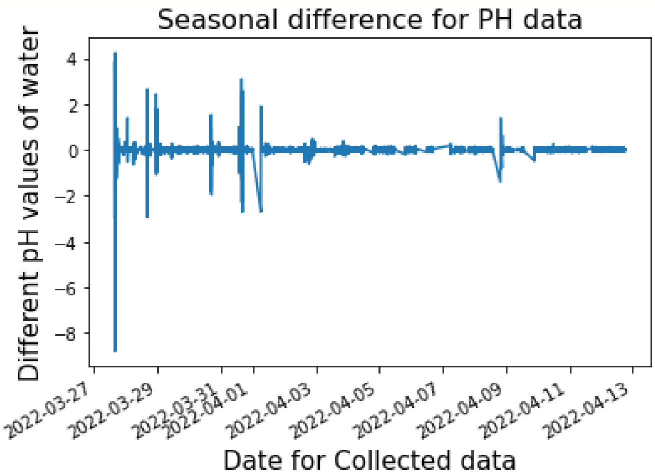

Fig5. 11: Stationary data of PH using Sarima Arima

step8. Split dataset

To begin with, the data must be divided into two sections: a training part and a testing section. As a result, we can achieve this because we begin by training our model on data and then hide the testing phase. Once the model is ready, we ask it to generate predictions based on the test data and measure how well it does.

```
print(df.shape)
```

```
train=df.iloc[:-30]

test=df.iloc[-30:]

print(train.shape,test.shape)
```

step9. ARIMA model creation

```
model=ARIMA(df['PH'],order=(3,1,2))

model_fit=model.fit()

model_fit.summary()
```

step10. Future predict calculation

```
predictions=model.predict(start=57000,end=60100,typ='levels').rename('A
RIMA Predictions')
```

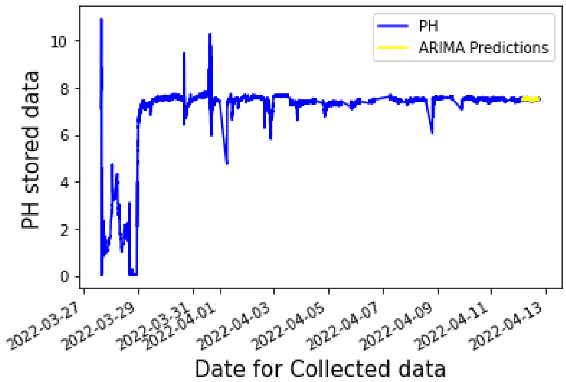

Fig5. 11: Two days Prediction of PH sensor data using Sarima arima

step 11. metrics accuracy

In order to get a real sense of how accurate my model is, I calculate the root mean squared error.

```
from sklearn.metrics import mean_squared_error

from math import sqrt

test['PH'].mean()

rmse=sqrt(mean_squared_error(predictions,test['PH']))

print(rmse)
```

5.3.3 Result and Discussion (Sarima and Arima)

Here using the sarima and arima model I analyze the same data which I used during Fb Prophet analysis. I analyze the prediction of storage data and calculate the RMSE error.

Table 8: ARIMA Model Results

Name of parameter	Coef	std err	Z	P>|z| [0.025 0.975]
Const	6.297e-06	0.000	0.032	0.975 -0.000 0.000
Autoregressive model (random walk of nonseasonal difference of PH in an autoregression)	0.2915	0.025	11.498	0.000 0.242 0.341
Moving Average model *(Random walk of nonseasonal difference of PH in a Moving average)*	-0.4536	0.024	-19.090	0.000-0.500-0.407

Table 9: RMSE, values of eight different sensors installed domestic Water Quantity and Quantity Monitoring calculated using Sarima arima model

Sensor Name	RMSE value
Temperature	0.03626684
PH	0.021022204
Turbidity	0.03668817
Flow	1.494735981
TDS	51.91108645
Pressure	0.010117038
Volume	2.008127154
Color	0.047111798

Here Table8 I analyze the data sensor’s data using the sarima arima model, and Unlike Table5, where I analyzed the sensor’s data using Fb prophet, I got most of the sensor's RMSE value below one. Among eight sensors only three of them have RMSE error which is greater than one. It seems the rest of the sensors have quietly good performance. The performance of the sensor depends on its error values, the more error the sensor has the less the performance the sensor has.

5.4 Result Comparison between Fb Prophet and Sarima Arima

Table 10: RMSE result comparison between Sarima Arima and Fb Prophet

Sensor Name	RSME value using Sarima Arima	RMSE value using Fb Prophet
Temperature	**0.03626684**	1.4774932
PH	**0.021022204**	0.802964
Turbidity	**0.03668817**	46.42883067
Flow	1.494735981	0.734097
TDS	51.91108645	11.283986
Pressure	**0.010117038**	2.8979555
Volume	2.008127154	1.5041518
Color	**0.047111798**	5.968331

From the above Table it seems that the average of my sensors has good performance by sarima arima. Because the performance of the sensor depends on its RMSE error values. For sarima and arima model five sensors have a Rmse error less than one, (Temperature, PH, Turbidity, Pressure and color) where their Rmse are (0.03626684, 0.021022204, 0.03668817, 0.010117038, 0.047111798) respectively. On the other hand, it seems that the volume sensor (2.008127154 and 1.5041518) and TDS sensor (51.91108645 and 11.283986) for sarima arima and Fb Prophet respectively.

have a RMSE value which is more than one for both cases sarima arima and Fb prophet, but it looks they have good performance for Fb Prophet compared to sarima arima (1.5041518, 11.283986) repetitively, but these two doesn’t have good performance for the system as they have RMSE value more than one in both case analysis.

Flow sensor has RMSE value which is less than one (0.734097) in case of Fb Prophet and a RMSE value more than one (01.494735981) for Sarima Arima. This flow sensor has good performance for Fb Prophet.

5.5 Validation of a developed system for multiple system

5.5.1 Validation of quality for a developed system

For quality validation of developed system, I store water parameters of two different sources water from household and water from Badda Lake (behind United International University). to identify the performance and reliability of my developed system first I collect data of water parameters from household and store ThingSpeak server, after that I take some water from Badda lake and take measurement of its parameters using same sensors with previous one (same sensors like household). The difference between these two sources as shown in Table 11. Due to the time limitation the duration of collecting data from lake water is less than the duration of collecting data from household which means the number of stored data is also difference, However, As Table11 shows the water from the lake is not suitable for drinking as it’s not safe for heath. This proves that the developed system can notify and identify the safe and suitability of the drinking water. It can also prove the developed system can measure the water parameters from different sources. TDS and Turbidity from Badda Lake are much more as shown in Table12. Fig12,13 and 14 shows water sample from Badda lake and storage container. Both the Source the TDs and turbidity are high especially for Badda lake its highly out of range, this means my developed system can identity the suitability of drinking water and safety of the drinking water.

Table 11: Validation of developed system for multi system

Sensor name	Average value of water from house hold	Average value of water from Badda lake	Reference Value
PH	7.54	9.514641	6.5-8.5[22]
Temperature	29.39	30.27248366	30[23]
Turbidity	127.06	2261.601	1[25]
TDS	57.92	3568.708	300[26]
Color	74.35	30.98693	100[24]

Fig5. 12;Badda Lake

Fig5. 13: Taking of water sample

Fig5. 14: water Sample Container

5.5.2 Validation of quantity for a developed system

The below Fig15 shows the water level storage from household, the average of this water level is 16liter. The total volume of consuming water in the household where this experiment held on is based on the average flow of water, water level and water pressure. All these sensors were installed by RFL 35 container. Whenever the tap of this container is open the water was following, there it was measure the rate of flow water using flow, volume using ultrasonic sensor and pressure using pressure sensor.

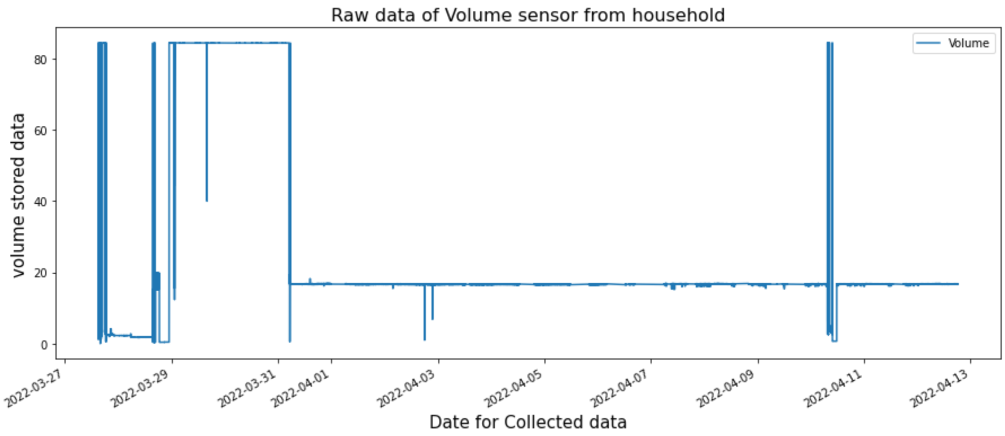

Fig5. 15: Volume storage of water household quantity

Table 12:: Ideal Cost analysis for experimental process

Liters of experimental container	Number days collected the data	Number of liters used per day for normal person	Cost of one liter in BD Tk [72]	Total cost consumed for 14days for one person In Tk	Total number of liters require for one person for a month	Total Consumed for a person for a month In Tk
35 liters	14	2	15	420	60	900

Table 13: Ideal consumed cost of flat of two bedrooms for three people

Number of people staying experimental house	Number of liters of water for a person per day	Number of liters of water for 3 people per day	Number of liters of water for 3 people per month	Cost of one liter in BD[72] (Tk)	Cost of total consumed water for this household customer per day in (Tk)	Total cost of consumed water for a month
3people	2 liters	6 liters	180 liters	15	90	2700

The cost of both systems developed system and ideal system are almost equal, the indicates the developed system is ideally suitable for water quantity validation. The collection process for quantity parameters was not perfect as expect, however, for ideal case the developed system can be consider as quantity validation and it can be used for future quantity parameters.

5.6 Performance Analysis of developed System

Table 14: Weightage Comparison for Water Quality Monitoring parameters

<table>
<tr><th>Contaminant Name</th><th>Parameter’s name</th><th colspan="2">Ideal System weightage</th><th colspan="2">Developed System Weightage</th></tr>
<tr><td>Inorganic contaminants</td><td>Arsenic</td><td colspan="2">13%</td><td colspan="2">0%</td></tr>
<tr><td>Inorganic contaminants</td><td>Fluoride</td><td colspan="2">8%</td><td colspan="2">0%</td></tr>
<tr><td>Inorganic contaminants</td><td>Mercury</td><td colspan="2">2%</td><td colspan="2">0%</td></tr>
<tr><td>Organic contaminants</td><td>Oil</td><td colspan="2">2%</td><td colspan="2">0%</td></tr>
<tr><td>Inorganic Contaminants</td><td>Copper</td><td colspan="2">2%</td><td colspan="2">0%</td></tr>
<tr><td>Organic</td><td>Sodium Chloride</td><td colspan="2">3%</td><td colspan="2">0</td></tr>
<tr><td>Inorganic Contaminants</td><td>Nitrates</td><td colspan="2">4%</td><td colspan="2">0%</td></tr>
<tr><td>Inorganic contaminants</td><td>Lead</td><td colspan="2">5%</td><td colspan="2">0%</td></tr>
<tr><td>Organic</td><td>Pesticides</td><td colspan="2">2%</td><td colspan="2">0%</td></tr>
<tr><td>Inorganic contaminants</td><td>PH (PH unit)</td><td colspan="2">18%</td><td colspan="2">18%</td></tr>
<tr><td>Global Warming</td><td>Temperature</td><td colspan="2">3%</td><td colspan="2">3%</td></tr>
<tr><td>Radiological Contaminants</td><td>Uranium</td><td colspan="2">1%</td><td colspan="2">0</td></tr>
<tr><td>Inorganic Contaminants</td><td>Color</td><td colspan="2">8%</td><td colspan="2">8%</td></tr>
<tr><td>Biological contaminants</td><td>Protozoa</td><td colspan="2">2%</td><td colspan="2">0</td></tr>
<tr><td>Biological contaminants</td><td>Phosphorus</td><td colspan="2">3%</td><td colspan="2">0</td></tr>
<tr><td>Inorganic contaminants</td><td>Turbidity</td><td colspan="2">7%</td><td colspan="2">7%</td></tr>
<tr><td>Biological contaminants</td><td>Bacteria, virus, house dust</td><td colspan="2">8%</td><td colspan="2">2%</td></tr>
<tr><td>Inorganic Contaminants</td><td>TDS</td><td colspan="2">9%</td><td colspan="2">9%</td></tr>
<tr><td></td><td></td><td>Ideal System weightage percentage</td><td>100%</td><td>Developed System Weightage</td><td>47%</td></tr>
</table>

Table14 shows necessary parameters which required to measure the quality of drinking water, for the developed system there are important parameters which have been used during the data collection process, while other important parameters have been missing the developed system, Due to their high commercial price and non-availability of market. The performance of the developed system is around 47%, However, the performance will increase if it's used one of the some of the missing parameters, it may at least reach half performance of ideal system.

5.7 Conclusion

In this chapter, I discuss a variety of machine learning techniques and analyses different methods for collecting, storing, and evaluating parameters relating to household water monitoring. After I have finished computing the RMSE values and descriptions of each sensor.

Utilizing the FbProphet technique and the Sarima Arima method resulted in getting RMSE values that were distinct from one another. This implies that the RMSE value of any sensor obtained through the FbProphet method was distinct from the RMSE values obtained through the Sarima Arima method. From here it seems that the average of my sensors has good performance by sarima arima. Because the performance of the sensor depends on its error values, and among eight sensors I got only three sensors, the greater one for sarima arima while I got the opposite to that using Fb prophet. Here I can say that some of my sensors have good performance by using Fb prophet and others have good performance using sarima arima while others have good performance by both analysis such as PH.

To do this, we compare the outcomes of these two distinct approaches. In that section, we explain the reasons why we get large error numbers, the performance of sensors, and the ways to get low RMSE values.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The findings of this study suggest the implementation of an Internet of Things system that is capable of performing accurate and cost-friendly real-time monitoring of water quality. With the system that has been designed, which consists of Arduino Mega and NodeMCU target boards, it is possible to successfully link several sensors. Real-time algorithm development is performed to keep an eye on the state of the water supply.

ThingSpeak is used to monitor the parameters of the water, such as the pH value, the turbidity, and the total dissolved solids (TDS), as well as the pressure of the water and the level of the water in the tank, as well as the temperature of the water, and the flow of water via the site. Other parameters that can be monitored include the level of the water in the tank, the level of the water in the tank, and the flow of water via the site. In addition to that, the machine and time series analysis techniques that were used to compute and determine the description of these observed parameters, RMSE, MSE, MAE errors values as well as their future prediction.

For Fb Prophet Out of all the water parameters that were measured at that spot, the RMSE value for turbidity was found to have the highest value (46.42883067), while the RMSE value for flow was found to have the lowest value (0.734097). This indicates that the flow sensor has the best performance compared to the other sensor, which indicates that the turbidity sensor has the worst performance. Most sensors have more than one RMSE error value.

For Sarima Arima Out of all the water parameters that were measured at that spot, the RMSE value for TDS was found to have the highest value (51.91108645), while the RMSE value for PH was found to have the lowest value (0.021022204). This indicates that the PH sensor has the best performance compared to the other sensor, which indicates that the TDS sensor has the worst performance.

For these two different time series analyses, compared to their RMSE values of individual sensor to examine the performance of the system, I found that the system has a good performance of 62.5% (five sensors) and 12.5% (one sensor) for sarima Arima and FB Prophet respectively, while 25% (two sensors) of the system doesn't good performance neither sarima Arima nor Fb prophet.

Because we obtain varying outcomes, it is possible that the performance analysis of a single source will not be sufficient. We are going to utilize a different strategy to analyze the data. It leads to producing findings that are different from the source that was used before. because we compare both sets of results obtained using various approaches. The result two sources have different output which means the developed system can monitor both clean water and contaminated water and notify the safety of the drinking water.

This developed system is reliable around 47% of necessary measurable parameter as there is some important sensors for water quality which are missing this system. Additionally, this developed system is suitable for validation of multiple system, here it checked two different source, household water sample, and Badda lake water sample. Average values of parameters from both sources are getting different values which makes the developed system can identify the quality of water for multiple system.

6.2 Future Work

This research needs to be done to undertake an analysis of several other factors found in the water, including its electrical conductivity, free residual chlorine, nitrates, and dissolved oxygen levels.

Because this project will be carried out using the internet, it will be possible for us to create a notification system that will enable clients and vendors to communicate with one another. In addition, we can utilize this project to create an industrial monitoring system. We can create a billing system that will allow the provider, customers, and government to each be able to determine the amount of usable water. The providers may provide a mobile application or dashboard system that allows customers to monitor the status of their home's water supply.

References

[1] J. Hyeon Ryu, "UAS-based Real-Time Water Quality Monitoring, Sampling, and Visualization Platform (UASWQP)," *HardwareX*, vol. 11, p. e00277, 2022, doi: 10.1016/j.ohx.2022.e00277.

[2] P. Li, D. Karunanidhi, T. Subramani, and K. Srinivasamoorthy, "Sources and Consequences of Groundwater Contamination," *Arch. Environ. Contam. Toxicol.*, vol. 80, no. 1, pp. 1–10, 2021, doi: 10.1007/s00244-020-00805-z.

[3] M. F. Hossain, "Arsenic contamination in Bangladesh - An overview," *Agric. Ecosyst. Environ.*, vol. 113, no. 1–4, pp. 1–16, 2006, doi: 10.1016/j.agee.2005.08.034.

[4] A. Borthakur and P. Singh, *Sustainability science—below and above the ground as per the United Nation's sustainable development goals*. LTD, 2020. doi: 10.1016/b978-0-12-818032-7.00017-5.

[5] S. S. Ahluwalia and D. Goyal, "Microbial and plant derived biomass for removal of heavy metals from wastewater," *Bioresour. Technol.*, vol. 98, no. 12, pp. 2243–2257, 2007, doi: 10.1016/j.biortech.2005.12.006.

[6] C. F. Harvey *et al.*, "Groundwater arsenic contamination on the Ganges Delta: Biogeochemistry, hydrology, human perturbations, and human suffering on a large scale," *Comptes Rendus - Geosci.*, vol. 337, no. 1–2, pp. 285–296, 2005, doi: 10.1016/j.crte.2004.10.015.

[7] Meenakshi and R. C. Maheshwari, "Fluoride in drinking water and its removal," *J. Hazard. Mater.*, vol. 137, no. 1, pp. 456–463, 2006, doi: 10.1016/j.jhazmat.2006.02.024.

[8] A. Kass, I. Gavrieli, Y. Yechieli, A. Vengosh, and A. Starinsky, “The impact of freshwater and wastewater irrigation on the chemistry of shallow groundwater: A case study from the Israeli Coastal Aquifer,” *J. Hydrol.*, vol. 300, no. 1–4, pp. 314–331, 2005, doi: 10.1016/j.jhydrol.2004.06.013.

[9] A. F. Al Yaqout, “Assessment and analysis of industrial liquid waste and sludge disposal at unlined landfill sites in arid climate,” *Waste Manag.*, vol. 23, no. 9, pp. 817–824, 2003, doi: 10.1016/S0956-053X(03)00036-9.

[10] J. O. Nriagu, “A Silent Epidemic of Environmental Metal Poisoning? Jerome O. Nriagu,” *Water Res.*, vol. 50, pp. 139–161, 1988, [Online]. Available: file:///C:/Users/Acer/Documents/Mendeley Desktop/RENOMEADO/Unknown - 1988 - A Silent Epidemic of Environmental Metal Poisoning Jerome O. Nriagu.pdf

[11] R. W. Harvey, R. L. Smith, and L. George, “Effect of organic contamination upon microbial distributions and heterotrophic uptake in a cape cod, mass., aquifer,” *Appl. Environ. Microbiol.*, vol. 48, no. 6, pp. 1197–1202, 1984, doi: 10.1128/aem.48.6.1197-1202.1984.

[12] F. D. Daschner, H. Rüden, R. Simon, and J. Clotten, “Microbiological contamination of drinking water in a commercial household water filter system,” *Eur. J. Clin. Microbiol. Infect. Dis.*, vol. 15, no. 3, pp. 233–237, 1996, doi: 10.1007/BF01591360.

[13] A. Binesh, S. Mohammadi, A. Mowlavi, and P. Parvaresh, “Research Article Measurement of Heavy Radioactive Pollution : Radon and Radium in Drinking Water Samples of Mashhad,” *Int. J.*, vol. 10, no. 1, pp. 54–58, 2010.

[14] WHO, “Arsenic 15,” *World Heal. Organ.*, no. February, pp. 1–6, 2018.

[15] American Cancer Society, “Water Fluoridation and Cancer Risk,” *Learn About Cancer*, pp. 1–10, 2015, [Online]. Available:

http://www.cancer.org/cancer/cancercauses/othercarcinogens/athome/water-fluoridation-and-cancer-risk

[16] W. H. O. WHO, “Mercury in Drinking-water, Background document for development of WHO Guidelines for Drinking-water Quality,” *Who*, vol. WHO/SDE/WS, p. WHO/SDE/WSH/05.08/10, 2005, [Online]. Available: http://www.who.int/water_sanitation_health/dwq/chemicals/mercuryfinal.pdf

[17] H. Effects and Y. Family, “Copper in Drinking Water,” *Copp. Drink. Water*, 2000, doi: 10.17226/9782.

[18] I. W. As, M. C. Level, P. H. Effects, T. Methods, R. Osmosis, and C. Exchange, “USEPA, Lead and Copper Monitoring and Reporting Guidance for Public Water Systems, EPA-816-R-02-009, Ground Water and Drinking Water Division, Water Programs, US Environmental Protection Agency, Washington, DC, 2002,” no. Cdc.

[19] P. Hye-Knudsen, “Nitrate in drinking water and methaemoglobinaemia,” *Ugeskr. Laeger*, vol. 147, no. 1, pp. 51–53, 1985.

[20] ATSDR Agency for Toxic Substances and Disease Registry, “Lead Toxicity What Are U . S . Standards for Lead Levels ?,” *Learning*, pp. 1–7, 2007, [Online]. Available: https://www.atsdr.cdc.gov/csem/leadtoxicity/safety_standards.html

[21] D. Quality, “English only Selenium in Drinking-water Background document for development of WHO Guidelines for Drinking-water Quality,” vol. 2, 2003.

[22] World Health Organisation, “pH in drinking-water,” *Guidel. Drink. water Qual.*, vol. 2, no. 2, pp. 1–7, 2007, [Online]. Available: http://www.who.int/water_sanitation_health/dwq/chemicals/ph_revised_2007_clean_version.pdf

[23] World Health Organization, *WHO Housing and health guidelines*. 2018. [Online].

Available: http://www.who.int/phe%0Ahttp://apps.who.int/bookorders.

[24] S. Sharma and A. Bhattacharya, “Drinking water contamination and treatment techniques,” *Appl. Water Sci.*, vol. 7, no. 3, pp. 1043–1067, 2017, doi: 10.1007/s13201-016-0455-7.

[25] WHO, “WATER QUALITY AND HEALTH - REVIEW OF TURBIDITY: Information for regulators and water suppliers,” *Who/Fwc/Wsh/17.01*, p. 10, 2017, [Online]. Available: https://www.who.int/water_sanitation_health/publications/turbidity-information-200217.pdf%0Ahttp://www.who.int/water_sanitation_health/publications/turbidity-information-200217.pdf

[26] A. I. Shaik, “What Is TDS & How Does It Affect Your Drinking Water ? What Is TDS In Water ? Subscribe to Onsitego,” *WHO*, vol. 30, no. 1–5, 2021, [Online]. Available: https://onsitego.com/blog/tds-level-safe-drinking-water-purifier/#:~:text=According to WHO%2C water with,is not recommended for drinking.

[27] M. Syafrudin, R. A. Kristanti, A. Yuniarto, T. Hadibarata, and J. Rhee, “Pesticides in Drinking Water — A Review,” *Int. J. Environ. Res. Public Health*, vol. 18, no. 2, p. 468, 2021.

[28] S. H. Frisbie, E. J. Mitchell, and B. Sarkar, “World Health Organization increases its drinking-water guideline for uranium,” *Environ. Sci. Process. Impacts*, vol. 15, no. 10, pp. 1817–1823, 2013, doi: 10.1039/c3em00381g.

[29] M. Kumar Jha, R. Kumari Sah, M. S. Rashmitha, R. Sinha, B. Sujatha, and K. V. Suma, “Smart Water Monitoring System for Real-Time Water Quality and Usage Monitoring,” *Proc. Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2018*, no. Icirca, pp. 617–621, 2018, doi: 10.1109/ICIRCA.2018.8597179.

[30] A. Jain, A. Malhotra, A. Rohilla, and P. Kaushik, “Water Quality Monitoring and Management System for Residents,” *Int. J. Eng. Adv. Technol.*, vol. 9, no. 2, pp. 567–570, 2019, doi: 10.35940/ijeat.b3521.129219.

[31] V. Saraswathi, A. Rohit, S. Sakthivel, and T. Sandheep, “Water Leakage System Using IoT,” *Int. J. Innov. Res. Eng. Manag.*, vol. 5, no. 2, pp. 67–69, 2018.

[32] A. B. Das, K. S. Srivatsava, and P. Doss, “Smart Water Conservation and Management System Using IOT (Water Utility Monitoring and Control Using IOT Based Approach),” vol. 9Water uti, pp. 9–12, 2018.

[33] K. S S and A. G N, “An IOT based Water Supply Monitoring and Controlling System,” *Int. Res. J. Eng. Technol.*, vol. 05, no. 02, pp. 857–860, 2018, [Online]. Available: https://www.irjet.net/archives/V5/i2/IRJET-V5I2189.pdf

[34] S. Sarkar, S. Sikder, S. I. Ashik, and A. Siddika, “Analysis , Design and Development of an IoT Based Water Management System for Residence,” *Glob. Sci. J.*, vol. 6, no. 10, pp. 398–403, 2018.

[35] C. Rajurkar, S. R. S. Prabaharan, and S. Muthulakshmi, “IoT based water management,” *2017 Int. Conf. Nextgen Electron. Technol. Silicon to Software, ICNETS2 2017*, pp. 255–259, 2017, doi: 10.1109/ICNETS2.2017.8067943.

[36] B. Das and P. C. Jain, “Real-time water quality monitoring system using Internet of Things,” *2017 Int. Conf. Comput. Commun. Electron. COMPTELIX 2017*, pp. 78–82, 2017, doi: 10.1109/COMPTELIX.2017.8003942.

[37] J. Shah, “An Internet of Things Based Model for Smart Water Distribution with Quality Monitoring,” *Int. J. Innov. Res. Sci.*, vol. 6, no. 3, pp. 3446–3451, 2017, doi: 10.15680/IJIRSET.2017.0603074.

[38] R. M. Kingsta, A. S. Saumi, and P. Saranya, “Design and construction of arduino based

pH control system for household waste water reuse,” *Proc. Int. Conf. Trends Electron. Informatics, ICOEI 2019*, no. Icoei, pp. 1037–1041, 2019, doi: 10.1109/ICOEI.2019.8862752.

[39] J. Courbat, Y. B. Kim, D. Briand, and N. F. De Rooij, “Inkjet printing on paper for the realization of humidity and temperature sensors,” *2011 16th Int. Solid-State Sensors, Actuators Microsystems Conf. TRANSDUCERS’11*, pp. 1356–1359, 2011, doi: 10.1109/TRANSDUCERS.2011.5969506.

[40] U. Gowrishetty, K. Walsh, S. Mcnamara, T. Roussel, and J. Aebersold, “SINGLE ELEMENT 3-TERMINAL PRESSURE SENSORS : A NEW APPROACH TO PRESSURE SENSING AND ITS COMPARISON TO THE HALF BRIDGE SENSORS Electrical & Computer Engineering , Louisville , KY , USA Department of Biomedical Engineering , Louisville , KY , USA that the 3,” vol. 1, pp. 1134–1137, 2009.

[41] F. Shaun, H. Regina, F. Marty, T. Bourouina, W. Cesar, and A. Fabrication, “for Water Network Monitoring,” no. c.

[42] C. T. Chiang, S. M. Huang, and C. N. Wu, “Development of a Calibrated Transducer CMOS Circuit for Water Turbidity Monitoring,” *IEEE Sens. J.*, vol. 16, no. 11, pp. 4478–4483, 2016, doi: 10.1109/JSEN.2016.2544800.

[43] C. Network, C. Fundamentals, I. Devices, and I. Platforms, “IoT Project using Ultrasonic Sensor HC- ← Prev SR04 and Arduino to distance calculation Processing App using Hardware Requirements The working principle of Arduino-Bluetooth Module”.

[44] D. Trivedi, A. Khade, K. Jain, and R. Jadhav, “SPI to I2C Protocol Conversion Using Verilog,” *Proc. - 2018 4th Int. Conf. Comput. Commun. Control Autom. ICCUBEA*

2018, pp. 1–4, 2018, doi: 10.1109/ICCUBEA.2018.8697415.

[45] A. Zare and M. T. Iqbal, “Low-Cost ESP32, Raspberry Pi, Node-Red, and MQTT protocol based SCADA system,” *IEMTRONICS 2020 - Int. IOT, Electron. Mechatronics Conf. Proc.*, pp. 0–4, 2020, doi: 10.1109/IEMTRONICS51293.2020.9216412.

[46] N. Rapelli, A. Myakal, V. Kota, and P. R. Rajarapollu, “IOT based smart water management, monitoring and distribution system for an apartment,” *2019 Int. Conf. Intell. Comput. Control Syst. ICCS 2019*, no. Iciccs, pp. 440–443, 2019, doi: 10.1109/ICCS45141.2019.9065369.

[47] S. Kavi Priya, G. Shenbagalakshmi, and T. Revathi, “Architecture of smart sensors for real time drinking water quality and contamination detection in water distributed mains,” *Rom. J. Inf. Sci. Technol.*, vol. 22, no. 3–4, pp. 202–214, 2019.

[48] K. K. Patel, S. M. Patel, and P. G. Scholar, “Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges,” *Int. J. Eng. Sci. Comput.*, vol. 6, no. 5, pp. 1–10, 2016, doi: 10.4010/2016.1482.

[49] K. Sabanci, E. Yigit, D. Üstün, A. Toktaş, and Y. Çelik, “Thingspeak Based Monitoring IoT System for Counting People in A Library,” *2018 Int. Conf. Artif. Intell. Data Process. IDAP 2018*, no. 1, p. 1, 2019, doi: 10.1109/IDAP.2018.8620793.

[50] G. Lakshmi Harika, H. Chowdary, and T. Satya Kiranmai, “Cloud-based Internet of things for smart water consumption monitoring system,” *Proc. 5th Int. Conf. Commun. Electron. Syst. ICCES 2020*, no. Icces, pp. 967–972, 2020, doi: 10.1109/ICCES48766.2020.09138074.

[51] J. Fox, A. Donnellan, and L. Doumen, “The deployment of an IoT network infrastructure, as a localised regional service,” *IEEE 5th World Forum Internet Things,*

WF-IoT 2019 - Conf. Proc., pp. 319–324, 2019, doi: 10.1109/WF-IoT.2019.8767188.

[52] T. O. Ayodele, "Atherosclerotic Cardiovascular Disease," *Atheroscler. Cardiovasc. Dis.*, 2012, doi: 10.5772/711.

[53] S. Ray, "Introduction to Machine Learning and Different types of Machine Learning Algorithms," *Proc. Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. Trends, Prespectives Prospect. Com. 2019*, pp. 35–39, 2019, doi: 10.1109/COMITCon.2019.8862451.

[54] A. Chakure, "Introduction to Machine Learning and Different types of Machine Learning Algorithms", [Online]. Available: https://hackernoon.com/introduction-to-machine-learning-and-different-types-of-machine-learning-algorithms-3z57i242u

[55] J. C. Huang, K. M. Ko, M. H. Shu, and B. M. Hsu, "Application and comparison of several machine learning algorithms and their integration models in regression problems," *Neural Comput. Appl.*, vol. 32, no. 10, pp. 5461–5469, 2020, doi: 10.1007/s00521-019-04644-5.

[56] J. Henke, "Regression Analysis," *Life Course Res. Soc. Policies*, vol. 11, pp. 275–286, 2020, doi: 10.1007/978-3-030-36323-9_26.

[57] N. Satyanarayana, C. Ramalingaswamy, and Y. Ramadevi, "Survey of Classification Techniques in Data Mining," *IJISET -International J. Innov. Sci. Eng. Technol.*, vol. 1, no. 9, pp. 268–278, 2014, [Online]. Available: www.ijiset.com

[58] R. Now, "Linear Regression Algorithm from Scratch What is Regression ? Where is Linear Regression Used ? 1 . Evaluating Trends and Sales Estimates," 2021.

[59] "Logistic Learning Regression in Machine Logistic Function (Sigmoid Function): Assumptions for Logistic Regression : Logistic Regression Equation : Help Others , Please Share," p. 4256464, 2021.

[60] P. Meel, M. Singhal, M. Tanwar, and N. Saini, “using Machine Learned Classifiers,” pp. 71–76, 2020.

[61] N. Darapaneni, P. Jain, R. Khattar, M. Chawla, R. Vaish, and A. R. Paduri, “Analysis and Prediction of COVID-19 Pandemic in India,” *Proc. - IEEE 2020 2nd Int. Conf. Adv. Comput. Commun. Control Networking, ICACCCN 2020*, pp. 291–296, 2020, doi: 10.1109/ICACCCN51052.2020.9362817.

[62] E. Žunić, K. Korjenić, K. Hodžić, and D. Đonko, “Application of Facebook’s Prophet Algorithm for Successful Sales Forecasting Based on Real-world Data,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 12, no. 2, pp. 23–36, 2020, doi: 10.5121/ijcsit.2020.12203.

[63] M. Valipour, “Long-term runoff study using SARIMA and ARIMA models in the United States,” *Meteorol. Appl.*, vol. 22, no. 3, pp. 592–598, 2015, doi: 10.1002/met.1491.

[64] G. Madey, V. Freeh, and R. Tynan, “The open source software development phenomenon: An analysis based on social network theory,” *Am. Conf. Inf. Syst. {(AMCIS2002)}*, pp. 1806–1813, 2002.

[65] X. Zhu and M. Shen, “Based on the ARIMA model with grey theory for short term load forecasting model,” *2012 Int. Conf. Syst. Informatics, ICSAI 2012*, no. Icsai, pp. 564–567, 2012, doi: 10.1109/ICSAI.2012.6223060.

[66] Y. Yang, C. Liu, and F. Guo, “Forecasting method of aero-material consumption rate based on seasonal ARIMA model,” *2017 3rd IEEE Int. Conf. Comput. Commun. ICCC 2017*, vol. 2018-Janua, pp. 2899–2903, 2018, doi: 10.1109/CompComm.2017.8323062.

[67] M. Murat, I. Malinowska, M. Gos, and J. Krzyszczak, “Forecasting daily meteorological time series using ARIMA and regression models,” *Int. Agrophysics*, vol. 32, no. 2, pp. 253–264, 2018, doi: 10.1515/intag-2017-0007.

[68] B. L. Sean J. Taylor, “Prophet : forecasting at scale Where Prophet shines,” *Meta Res.*, pp. 1–5, 2017, [Online]. Available: https://research.facebook.com/blog/2017/02/prophet-forecasting-at-scale/

[69] H. Z. Abyaneh, “Evaluation of multivariate linear regression and artificial neural networks in prediction of water quality parameters,” *J. Environ. Heal. Sci. Eng.*, vol. 12, no. 1, pp. 6–13, 2014, doi: 10.1186/2052-336X-12-40.

[70] Z. Hu *et al.*, “A water quality prediction method based on the deep LSTM network considering correlation in smart mariculture,” *Sensors (Switzerland)*, vol. 19, no. 6, 2019, doi: 10.3390/s19061420.

[71] R. Barzegar, M. T. Aalami, and J. Adamowski, “Short-term water quality variable prediction using a hybrid CNN–LSTM deep learning model,” *Stoch. Environ. Res. Risk Assess.*, vol. 34, no. 2, pp. 415–433, 2020, doi: 10.1007/s00477-020-01776-2.

[72] D. WASA, “Dhaka WASA proposes minimum 20pc hike in water price,” *Financ. Expresse*, pp. 1–2, 2022, [Online]. Available: https://thefinancialexpress.com.bd/national/dhaka-wasa-proposes-minimum-20pc-hike-in-water-price-1644405952

Appendix A

Code used for this project

```
#include <Wire.h>
#include <EEPROM.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include "HX710B.h"
#include <LiquidCrystal_I2C.h>
#include <HTTPClient.h>
#define S0 2
#define S1 15
#define S2 12
#define S3 4
#define sensorOut 39
int frequency = 0;
String volume="";
double area=1.0; //m^2
// set the LCD number of columns and rows
int lcdColumns = 16;
int lcdRows = 2;
// set LCD address, number of columns and rows
// if you don't know your display address, run an I2C scanner sketch
LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);
const char* ssid = "iPhone";
const char* password = "1234567890";
// Domain Name with full URL Path for HTTP POST Request
const char* serverName = "http://api.thingspeak.com/update";
// Service API Key
```

```
String apiKey = "JOVGRYWNC8CYJCYD";
// the following variables are unsigned longs because the time, measured in
// milliseconds, will quickly become a bigger number than can be stored in an int.
unsigned long lastTime = 0;
// Timer set to 10 minutes (600000)
//unsigned long timerDelay = 600000;
// Set timer to 5 seconds (5000)
unsigned long timerDelay = 5000;
//pressure pin
const int DOUT = 19;   //sensor data pin
const int SCLK  = 18;   //sensor clock pin
String prpascal="";
HX710B pressure_sensor;
#define TdsSensorPin 35
#define PhSensorPin 32
#define VREF 3.3      // analog reference voltage(Volt) of the ADC
#define SCOUNT  30           // sum of sample point
int analogBuffer[SCOUNT];    // store the analog value in the array, read from ADC
int analogBufferTemp[SCOUNT];
int analogBufferIndex = 0;
int copyIndex = 0;
float averageVoltage = 0;
float tdsValue = 0;
float temperature = 25
int turbidityPin = 34;
float volt=0;
float ntu=0;
float pressure=0;
float tdsraw=0;
String ph="0.0";
String red="",green="",blue="";
```

```
#define temp_SENSOR_PIN  13 // ESP32 pin GIOP13 connected to DS18B20 sensor's DQ pin
OneWire oneWire(temp_SENSOR_PIN);
DallasTemperature DS18B20(&oneWire);
float tempC=0; // temperature in Celsius
float tempF=0; // temperature in Fahrenheit
const int trigPin = 17;
const int echoPin = 16;
//define sound speed in cm/uS
#define SOUND_SPEED 0.034
#define CM_TO_INCH 0.393701
long duration;
float distanceCm;
float distanceInch;
//flo part
#define flowSENSOR 33
long currentMillis = 0;
long previousMillis = 0;
int interval = 1000;
float calibrationFactor = 4.5;
volatile byte pulseCount;
byte pulse1Sec = 0;
float flowRate;
unsigned int flowMilliLitres;
unsigned long totalMilliLitres;
void IRAM_ATTR pulseCounter()
{
 pulseCount++;
}
//flo end
void setup() {
   Serial.begin(115200);
```

```
    // initialize LCD
  lcd.init();
  // turn on LCD backlight
  lcd.backlight();
   pinMode(flowSENSOR, INPUT_PULLUP);
   pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input

  pinMode(S0, OUTPUT);
pinMode(S1, OUTPUT);
pinMode(S2, OUTPUT);
pinMode(S3, OUTPUT);
pinMode(sensorOut, INPUT);
// Setting frequency-scaling to 20%
digitalWrite(S0,HIGH);
digitalWrite(S1,HIGH);
   //wifi setup
    WiFi.begin(ssid, password);
  Serial.println("Connecting");
  while(WiFi.status() != WL_CONNECTED) {
   delay(500);
   Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to WiFi network with IP Address: ");
  Serial.println(WiFi.localIP());
//wifi done

    DS18B20.begin();    // initialize the DS18B20 sensor
    pressure_sensor.begin(DOUT, SCLK);
//flow
pulseCount = 0;
```

```
 flowRate = 0.0;
 flowMilliLitres = 0;
 totalMilliLitres = 0;
 previousMillis = 0;
 attachInterrupt(digitalPinToInterrupt(flowSENSOR), pulseCounter, FALLING);
// Init and get the time
 // configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
 // printLocalTime();
}
void loop() {
 // digitalWrite(15,HIGH);
  turbidityread();
  phread();
  tempread();
  tdsread();
  pressureread();
  flow();
  sonar();
  color();
  lcddip();
  senddata();
}
void phread(){
  float phvalue=0.0;
  float avgph;
  float Value;
  for(int i=0;i<100;i++){
    Value= analogRead(32);
    float voltage=Value*(3.3/4095.0);
    phvalue=(3.3*voltage);
    avgph=avgph+phvalue;
  }
```

```
  phvalue=avgph/100.0;
  Serial.println("****************PH Sensor***************************");
   Serial.print(Value);
   Serial.print(" | ph = ");
   ph=String(phvalue,2);
   Serial.println(ph);
   delay(500);
  }
void pressureread(){
   if (pressure_sensor.is_ready()) {
   pressure=pressure_sensor.pascal();
   prpascal=String(pressure_sensor.pascal());
   Serial.println("****************Pressure Sensor***************************");
   Serial.print("Pascal: ");
   Serial.println(prpascal);
   Serial.print("ATM: ");
   Serial.println(pressure_sensor.atm());
   Serial.print("mmHg: ");
   Serial.println(pressure_sensor.mmHg());
   Serial.print("PSI: ");
   Serial.println(pressure_sensor.psi());
   Serial.println("*******************************************");
  } else {
   Serial.println("Pressure sensor not found.");
  }
  delay(1000);
  }
void tempread(){
  DS18B20.requestTemperatures();       // send the command to get temperatures
  tempC = DS18B20.getTempCByIndex(0);  // read temperature in °C
  tempF = tempC * 9 / 5 + 32; // convert °C to °F
  Serial.println("****************Temp Sensor***************************");
```

```
  Serial.print("Temperature: ");
  Serial.print(tempC);    // print the temperature in °C
  Serial.print("°C");
  Serial.print("  ~  ");  // separator between °C and °F
  Serial.print(tempF);    // print the temperature in °F
  Serial.println("°F");
  temperature = tempC;
  delay(500);
  }
void tdsread(){
  Serial.println("****************Tds Sensor**************************");
   float temperature = tempC;

  static unsigned long analogSampleTimepoint = millis();
  if (millis() - analogSampleTimepoint > 40U)  //every 40 milliseconds,read the analog value
from the ADC
  {
    analogSampleTimepoint = millis();
    analogBuffer[analogBufferIndex] = analogRead(TdsSensorPin);     //read the analog value
and store into the buffer
    Serial.print("Raw tds = ");
    Serial.println(analogBuffer[analogBufferIndex]);
    analogBufferIndex++;
    if (analogBufferIndex == SCOUNT)
      analogBufferIndex = 0;
  }
  static unsigned long printTimepoint = millis();
  if (millis() - printTimepoint > 800U)
  {
    printTimepoint = millis();
    for (copyIndex = 0; copyIndex < SCOUNT; copyIndex++)
      analogBufferTemp[copyIndex] = analogBuffer[copyIndex];
```

```
    averageVoltage = getMedianNum(analogBufferTemp, SCOUNT) * (float)VREF / 1024.0; // read the analog value more stable by the median filtering algorithm, and convert to voltage value
    float compensationCoefficient = 1.0 + 0.02 * (temperature - 25.0); //temperature compensation formula: fFinalResult(25^C) = fFinalResult(current)/(1.0+0.02*(fTP-25.0));
    float compensationVolatge = averageVoltage / compensationCoefficient; //temperature compensation
    tdsValue = (133.42 * compensationVolatge * compensationVolatge * compensationVolatge - 255.86 * compensationVolatge * compensationVolatge + 857.39 * compensationVolatge) * 0.5; //convert voltage value to tds value
    Serial.print("TDS Value:");
    Serial.print(tdsValue, 0);
    Serial.println("ppm");
    Serial.print("Temperature:");
    Serial.print(temperature);
    Serial.println("ºC");
  }
}
int getMedianNum(int bArray[], int iFilterLen)
{
  int bTab[iFilterLen];
  for (byte i = 0; i < iFilterLen; i++)
    bTab[i] = bArray[i];
  int i, j, bTemp;
  for (j = 0; j < iFilterLen - 1; j++)
  {
    for (i = 0; i < iFilterLen - j - 1; i++)
    {
      if (bTab[i] > bTab[i + 1])
      {
        bTemp = bTab[i];
        bTab[i] = bTab[i + 1];
        bTab[i + 1] = bTemp;
      }
```

```
    }
  }
  if ((iFilterLen & 1) > 0)
    bTemp = bTab[(iFilterLen - 1) / 2];
  else
    bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2;
  return bTemp;
}
void turbidityread(){
   volt = 0;
    for(int i=0; i<800; i++)
    {
        volt += ((float)analogRead(turbidityPin)/4095)*3.3;
    }
    volt = volt/800;
    volt = round_to_dp(volt,2);
    if(volt < 1.65){  //2.5for 5v, 1.65for 3.3v
      ntu = 3000;
    }
    else if(volt >= 2.772){
      ntu = 0;
    }else{
     // ntu = -1120.4*sq(volt)+5742.3*volt-4352.9; //5v equation
     ntu = -2572.2*sq(volt) + 8700.5*volt - 4352.9; //3.3v equation
    }
    Serial.println("****************Turbidity Sensor**************************");
    Serial.print("Voltage = ");
    Serial.println(volt);
    Serial.print("NTU = ");
    Serial.println(ntu);
  delay(1000);
  }
```

```
float round_to_dp( float in_value, int decimal_place )
{
  float multiplier = powf( 10.0f, decimal_place );
  in_value = roundf( in_value * multiplier ) / multiplier;
  return in_value;
}

void senddata(){
  //Send an HTTP POST request every 10 seconds
  if ((millis() - lastTime) > timerDelay) {
    //Check WiFi connection status
    if(WiFi.status()== WL_CONNECTED){
      WiFiClient client;
      HTTPClient http;

      // Your Domain name with URL path or IP address with path
      http.begin(client, serverName);

      // Specify content-type header
      http.addHeader("Content-Type", "application/x-www-form-urlencoded");
      // Data to send with HTTP POST
      String httpRequestData = "api_key=" + apiKey + "&field1=" + String(tempC)+ "&field2=" + String(ph)+ "&field3=" + String(ntu)+ "&field4=" + String(int(flowRate))+ "&field5=" + String(tdsValue)+ "&field6=" + String(prpascal)+ "&field7=" + String(volume)+ "&field8=" + String(frequency);
      // Send HTTP POST request
      int httpResponseCode = http.POST(httpRequestData);
      /*
      // If you need an HTTP request with a content type: application/json, use the following:
      http.addHeader("Content-Type", "application/json");
      // JSON data to send with HTTP POST
      String httpRequestData = "{\"api_key\":\"" + apiKey + "\",\"field1\":\"" + String(random(40)) + "\"}";
```

```
      // Send HTTP POST request
      int httpResponseCode = http.POST(httpRequestData);*/


      Serial.print("HTTP Response code: ");
      Serial.println(httpResponseCode);
      // Free resources
      http.end();
    }
    else {
      Serial.println("WiFi Disconnected");
    }
    lastTime = millis();
  }
}
void flow()
{
 currentMillis = millis();
 if (currentMillis - previousMillis > interval) {
 pulse1Sec = pulseCount;
 pulseCount = 0;
 flowRate = ((1000.0 / (millis() - previousMillis)) * pulse1Sec)/ calibrationFactor;
 previousMillis = millis();
 flowMilliLitres = (flowRate / 60) * 1000;
 // Add the millilitres passed in this second to the cumulative total
 totalMilliLitres += flowMilliLitres;
 // Print the flow rate for this second in litres / minute
  Serial.println("****************Flow Sensor**************************");
 Serial.print("Flow rate: ");
 Serial.print(int(flowRate)); // Print the integer part of the variable
 Serial.print("L/min");
 Serial.print("\t"); // Print tab space
 // Print the cumulative total of litres flowed since starting
```

```
 Serial.print("Output Liquid Quantity: ");
 Serial.print(totalMilliLitres);
 Serial.print("mL / ");
 Serial.print(totalMilliLitres / 1000);
 Serial.println("L");
 }
}
void sonar() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculate the distance
  distanceCm = duration * SOUND_SPEED/2;
  double distancem = distanceCm /100;
  volume=String(distancem *area );
  // Convert to inches
  distanceInch = distanceCm * CM_TO_INCH;
  Serial.println("****************Sonar Sensor*************************");
  // Prints the distance in the Serial Monitor
  Serial.print("Distance (cm): ");
  Serial.println(distanceCm);
  Serial.print("Distance (inch): ");
  Serial.println(distanceInch);
  Serial.print("Volume (m^3): ");
  Serial.println(volume);
```

```
  delay(1000);
}
void color(){
  // Setting red filtered photodiodes to be read
digitalWrite(S2,LOW);
digitalWrite(S3,LOW);
// Reading the output frequency
frequency = pulseIn(sensorOut, LOW);
red=String(frequency);
// Printing the value on the serial monitor
Serial.println("****************Color Sensor***************************");
Serial.print("R= ");//printing name
Serial.print(frequency);//printing RED color frequency
Serial.print(" ");
delay(100);
// Setting Green filtered photodiodes to be read
digitalWrite(S2,HIGH);
digitalWrite(S3,HIGH);
// Reading the output frequency
frequency = pulseIn(sensorOut, LOW);
green=String(frequency);
// Printing the value on the serial monitor
Serial.print("G= ");//printing name
Serial.print(frequency);//printing RED color frequency
Serial.print(" ");
delay(100);
// Setting Blue filtered photodiodes to be read
digitalWrite(S2,LOW);
digitalWrite(S3,HIGH);
// Reading the output frequency
frequency = pulseIn(sensorOut, LOW);
blue=String(frequency);
```

```
// Printing the value on the serial monitor
Serial.print("B= ");//printing name
Serial.print(frequency);//printing RED color frequency
Serial.println(" ");
delay(100);
  }

void lcddip(){
  // set cursor to first column, first row
  lcd.clear();
  lcd.setCursor(0, 0);
  // print message
  lcd.print("PH=");
  lcd.setCursor(3, 0);
  lcd.print(ph);
  lcd.setCursor(6, 0);
  // print message
  lcd.print("Temp=");
  lcd.setCursor(12, 0);
  lcd.print(String(tempC));
  //delay(1000);
  // clears the display to print new message
  //lcd.clear();
  // set cursor to first column, second row
  lcd.setCursor(0,1);
  lcd.print("Distance=");
  lcd.setCursor(9,1);
  lcd.print(distanceCm);
  delay(2000);
  lcd.clear();
  lcd.setCursor(0, 0);
  // print message
```

```
lcd.print("TDs=");
lcd.setCursor(4, 0);
lcd.print(tdsValue);
lcd.setCursor(6, 0);
delay(2000);
lcd.clear();
// print message
lcd.setCursor(0, 0);
lcd.print("Turb=");
lcd.setCursor(5, 0);
lcd.print(String(ntu));
//delay(1000);
// clears the display to print new message
//lcd.clear();
// set cursor to first column, second row
lcd.setCursor(0,1);
lcd.print("Flow=");
lcd.setCursor(5,1);
lcd.print(flowRate);
delay(2000);
lcd.clear();
lcd.setCursor(0, 0);
// print message
lcd.print("Pr=");
lcd.setCursor(3, 0);
lcd.print(prpascal);
lcd.setCursor(9, 0);
lcd.print("Pa");
// set cursor to first column, second row
lcd.setCursor(0,1);
lcd.print("R=");
lcd.setCursor(2,1);
```

```
  lcd.print(red);
  lcd.setCursor(5,1);
  lcd.print("G=");
  lcd.setCursor(7,1);
  lcd.print(green);
  lcd.setCursor(10,1);
  lcd.print("B=");
  lcd.setCursor(13,1);
  lcd.print(blue);
}
```