A SMART ENVIRONMENTAL MONITORING SYSTEM FOR DATA CENTRES USING IoT AND MACHINE LEARNING

Wayne Steven Okello

A Project Report Submitted in Partial Fulfilment of the Requirements for the Degree of Master of Science in Embedded and Mobile Systems of the Nelson Mandela African Institution of Science and Technology

Arusha, Tanzania

August, 2023

ABSTRACT

Data centres are a crucial part of many organizations in the world today consisting of expensive assets that store and process critical business data as well as applications responsible for their daily operations. Unconducive environmental conditions can lead to decline in performance, sporadic failures and total damage of equipment in the data centers which can consequently lead to data loss as well as disruption of the continuity of business operations. The objective of this project was to develop an environmental monitoring system that employs Internet of Things (IoT) and machine learning to monitor and predict important environmental parameters within a data centre setting. The system comprises of a Wireless Sensor Network (WSN) of four (4) sensor nodes and a sink node. The sensor nodes measure environmental parameters of temperature, humidity, smoke, water, voltage and current. The readings captured from the sensor nodes are sent wirelessly to a database on a Raspberry Pi 4 for local storage as well as the ThingSpeak platform for cloud data logging and real-time visualization. An audio alarm is triggered, and email, Short Message Service (SMS), as well as WhatsApp alert notifications are sent to the data centre administrators in case any undesirable environmental condition is detected. Time series forecasting machine learning models were developed to predict future temperature and humidity trends. The models were trained using Facebook Prophet, Auto-Regressive Integrated Moving Average (ARIMA) and Exponential Smoothing (ES) algorithms. Facebook Prophet manifested the best performance with a Mean Absolute Percentage Error (MAPE) of 5.77% and 8.98% for the temperature and humidity models respectively. In conclusion, the developed environmental monitoring system for data centers surpasses existing alternatives by integrating forecasting capabilities, monitoring several critical parameters, and offering scalability for improved efficiency and reliability. The study recommendations include exploring a Web of Things (WoT) approach and incorporating instant corrective measures for improved performance.

DECLARATION

I, Wayne Steven Okello, do hereby declare to the Senate of the Nelson Mandela African Institution of Science and Technology that this project report is my original work and that it has neither been submitted nor being concurrently submitted for a degree award in any other institution.

Wayne Steven Okello		18.8.2023
Name of Candidate	**Signature**	**Date**

The above declaration is confirmed by:

Dr. Silas Mirau		18.8.2023
Name of Supervisor 1	**Signature**	**Date**

Prof. Kisangiri Michael		01/09/2023
Name of Supervisor 2	**Signature**	**Date**

COPYRIGHT

This project report is copyright material protected under the Berne Convention, the Copyright Act of 1999, and other international and national enactments, on behalf, of intellectual property. It must not be produced by any means, in full or in part, except for shorts extracts in fair dealing, for researcher private study, critical scholarly review, or discourse with an acknowledgment, without the written permission of the office of Deputy Vice-Chancellor for Academic, Research, and Innovation on behalf of the author and the Nelson Mandela African Institution of Science and Technology.

CERTIFICATION

The undersigned certify that they have read and hereby recommend for acceptance by the Nelson Mandela African Institution of Science and Technology, a project report titled ***"A Smart Environmental Monitoring System for Data Centres Using IoT and Machine Learning"*** in partial fulfillment of the requirements for the degree of Master of Science in Embedded and Mobile Systems of the Nelson Mandela African Institution of Science and Technology.

Dr. Silas Mirau	[signature]	18.8.2023
Name of Supervisor 1	**Signature**	**Date**

Prof. Kisangiri Michael	[signature]	01/09/2022
Name of Supervisor 2	**Signature**	**Date**

ACKNOWLEDGEMENTS

First and foremost, I would like to thank the Almighty God for enabling me to successfully implement this project.

I would like to express my deepest thanks to my supervisors at the Nelson Mandela African Institution of Science and Technology, Dr. Silas Mirau and Prof. Michael Kisangiri for their numerous contributions throughout the project lifetime.

I would like to express my indebtedness appreciation to the Center of Excellence for East Africa (CENIT@EA) and DAAD GIZ's scholarship program for the financial support rendered towards my two-year studies at the Nelson Mandela African Institution of Science and Technology (NM-AIST).

Great thanks to the management of netLabs!UG for providing me with an industrial placement as well as a conducive research environment that facilitated the successful implementation of the research project.

It is my radiant sentiment to place on record my best regards, deepest sense of gratitude to my industrial supervisors, Dr. Andrew Katumba and Dr. Edwin Mugume for their technical guidance and support towards the implementation of this project as well as the report writing.

Last but not least, I would like to thank my dear parents, siblings and friends that always provided unending support during the course of the entire project.

DEDICATION

I dedicate this work to my dear parents, siblings and friends.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

3D	Three Dimensional
AC	Alternating Current
ADF	Augmented Dickey - Fuller
AI	Artificial Intelligence
API	Application Programming Interface
ARIMA	Auto-Regressive Integrated Moving Average
CPU	Central Processing Unit
DC	Direct Current
ES	Exponential Smoothing
GPIO	General Purpose Input/Output
HVAC	Heating, Ventilation, and Air Conditioning
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IFTTT	IF This Then That
IoT	Internet of Things
KPSS	Kwiatkowski-Phillips-Schmidt-Shin
LCD	Liquid Crystal Display
LoRa	Long Range Radio
LoRaWAN	Long Range Radio Wide Area Network
LPG	Liquefied Petroleum Gas
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MCU	Microcontroller Unit
MySQL	My Structured Query Language
NAT	Network Address Translation

PCB	Printed Circuit Board
PEOU	Perceived Ease of Use
PIC	Peripheral Interface Controller
PPM	Parts Per Million
PU	Perceived Usefulness
RMSE	Root Mean Squared Error
SD	Secure Digital
SMS	Short Message Service
SSH	Secure Shell
TAM	Technology Acceptance Model
UAT	User Acceptance Testing
UPSes	Uninterruptible Power Supplies
VNC	Virtual Network Computing
WoT	Web of Things
WSN	Wireless Sensor Network
XAMPP	Cross-platform, Apache server, MySQL, PHP, Perl
XP	Extreme Programming

CHAPTER ONE

INTRODUCTION

1.1 Background of the Problem

Monitoring is always paramount in any organization irrespective of its structure and nature of its business (Rasool *et al.*, 2014). Monitoring enables early identification of prospective problems so that they can be promptly dealt with before causing severe damage and thereby ensuring reliable execution of company operations.

A data centre refers to a facility that houses computing resources, networking and storage infrastructure as well as associated utilities such as power supplies and Heating, Ventilation, and Air Conditioning (HVAC) systems used by business entities to assemble, process, store, share and access vast quantities of data. Data centres vary in size from as small as an exclusive space within a building to as large as a group of buildings depending on various factors that include but are not limited to company size and core business of the organization.

Data centres were initially owned by mostly top technology firms but due to the increasing digitization, the evolution of new technologies such as Cloud Computing, Artificial Intelligence (AI), Internet of Things (IoT), Big Data, Blockchain, etc, in addition to the significantly growing technology acceptance across the world today, business operations are more data-driven than ever before and data centres have subsequently become a very crucial component in the business models of many modern enterprises (Marzuki & Newell, 2019).

There are a number of sources of unexpected data centre outages that include but are not limited to human errors, power outages, environmental factors, network failures, and cyberattacks, with environmental issues contributing to 25% of downtime according to Onibonoje *et al.* (2019). Environmental conditions such as extreme temperatures, presence of water, power spikes and surges as well as fires in the data centre can lead to equipment failure, damage and loss of huge volumes of critical data. Low humidity leads to static discharge as a result of electrostatic energy accumulation which damages equipment and causes loss of data, while high humidity results into condensation as well as corrosion of electrical components.

The developed system effectively addresses the environmental challenges by leveraging IoT for monitoring critical environmental parameters of temperature, humidity, water, smoke, current, and voltage. It promptly notifies data centre administrators when any unfavorable condition is detected, enabling them to take timely action. Furthermore, machine learning techniques are applied to forecast future temperature and humidity patterns, allowing administrators to proactively anticipate and mitigate potential issues.

The use of IoT facilitates the collection of real-time data and offers scalability and flexibility, as sensor nodes can be easily added or relocated to adapt to changing requirements. Meanwhile, machine learning provides valuable data-driven insights by uncovering hidden patterns and trends that may not be immediately evident. By combining IoT and machine learning, the system enables administrators to stay ahead of environmental challenges, implement preventive measures, and minimize the impact of disruptions in data centres.

1.2 Statement of the Problem

Data centres comprise of a number of core equipment such as servers, storage systems, networking switches, routers, firewalls as well as support infrastructure that include power subsystems, HVAC systems, Uninterruptable Power Supplies (UPSes), etc. These components are expensive organization assets and are responsible for storing as well as managing crucial business data and applications that are essential to the continuity of a firm's daily operations.

Unconducive environmental conditions such as extreme temperatures, humidity, fire, power spikes and surges, and water leaks can lead to a decline in performance, sporadic failures and total damage of equipment in the data centers which can consequently lead to data loss as well as disruption of the continuity of business operations.

Therefore, there was need to develop a smart real-time system that enables organizations to monitor and analyze significant environmental parameters i.e., temperature, humidity, water, smoke, current and voltage within their data centre premises. This system provides alerts of potential environmental hazards and also predicts future temperature and humidity patterns. This enables organizations to promptly take the necessary measures to alleviate their impact. Furthermore, unlike other environmental parameters that require immediate detection and response, the prediction of temperature and humidity patterns provides valuable insights for long-term planning, proactive maintenance and risk mitigation in data centres.

1.3 Rationale of the Study

A smart environmental monitoring system for data centres using IoT and machine learning ensures that firms are aware of the environmental conditions of their data centres at all times and are notified of events where pre-defined environmental parameter thresholds are exceeded so as to promptly take proactive measures. This in turn reduces unexpected downtime, increases reliability, minimizes risk of equipment damage and reduces the operating costs in the long term.

1.4 Objectives of the Study

1.4.1 General Objective

The main objective of the project was to develop a smart environmental monitoring system for data centres using IoT and machine learning.

1.4.2 Specific Objectives

The study aimed to achieve the following specific objectives:

(i) To determine and gather the requirements for a smart environmental monitoring system for data centres using IoT and machine learning.

(ii) To develop an IoT environmental monitoring system for data centres.

(iii) To develop machine learning models for predicting temperature and humidity patterns.

(iv) To validate the developed IoT system.

1.5 Research Questions

The study intended to answer the following questions:

(i) What are the essential requirements for developing a smart environmental monitoring system for data centres using IoT and machine learning?

(ii) How can an IoT environmental monitoring system for data centres be developed?

(iii) How can machine learning models for predicting temperature and humidity patterns be developed?

(iv) How can the developed IoT system be validated?

1.6 Significance of the Study

Real-time monitoring of environmental parameters in data centres enables operators to respond to unsuitable environmental conditions in a timely manner. In addition, analysis and prediction of some of these parameters can guide organizations in devising appropriate predictive measures and planning as well as optimization of resources. This consequently reduces the risk of damage to equipment, data loss, and operating costs as well as increases uptime thereby enhancing business continuity.

This project serves as a compelling demonstration of the practical application of IoT within a data center setting. It showcases the development and effective utilization of time series forecasting models for accurate and efficient environmental monitoring in data centers. By successfully implementing an IoT system and leveraging forecasting models, the project significantly enhances environmental monitoring practices in data centers. This technological advancement brings about improved reliability and proactive management of environmental conditions, fostering a more robust and resilient data center infrastructure.

1.7 Delineation of the Study

The project was delineated to monitoring of environmental parameters of temperature, humidity, smoke, water, current and voltage within a data centre setting. The scope of the project was limited to development of a WSN based module for capturing, storing and processing real-time environmental parameters, development of a web-based dashboard for monitoring and analysis of the environmental conditions, development of machine learning models to predict temperature and humidity trends as well as development of alert mechanisms upon detection of any unsuitable events.

CHAPTER TWO

LITERATURE REVIEW

2.1 Related Works

Kurniawan *et al.* (2019) developed an IoT based system that utilizes Wemos DHT Shield wireless sensor to capture the temperature and humidity readings. A Raspberry Pi processes the readings stored in a MySQL database and triggers the web server that enables the website with real-time graph visualizations to be accessible through the internet or a private wireless network. Furthermore, the system uses the python yowsup library to request data from the web server on the Raspberry Pi, processes it and finally returns the required information in form of WhatsApp notifications.

Roy *et al.* (2016) presented a server room cooling system that uses the single bus wire protocol to connect a DS18B20 sensor to a Raspberry pi MCU with the former as a slave and the latter as the master. The temperature is captured by the DS18B20 sensor which sends the sensed value to the Raspberry Pi, which compares it to the database threshold value and dispatches a warning email notification message to the data centre operator in case the temperature goes below the threshold.

Onibonoje *et al.* (2019), in their work propose a real-time conditioning and control system for server rooms based on IoT that monitors environmental factors namely heat, smoke, water leakage, power outage, light out, flame and correlates the heat index with the area air temperature and humidity. It further undertakes timely corrective measures such as turning off the system upon fire or smoke detection, turning on air conditioning system when heat threshold is surpassed, cutting off perimeter water supply in case of water leak. In addition, the system sends out notifications to the administrator in case of any unexpected event.

Kaliyamurthie *et al.* (2019) described a wireless temperature and humidity measurement as well as monitoring system for data centres that utilizes LoRaWAN communication link for data transmission. The LoRaWAN gateway sends the sensed data to a cloud application for further processing and visualization, allowing more optimization of the air conditioning and ventilation systems.

Yamanoue (2020) presented an IoT based system comprising of Raspberry Pi, sensors, Wiki pages and bots that monitors servers and server rooms by utilizing a Network Address Translation (NAT) router to configure as well as manage sensors using a Wiki page. The sampling rate of the data as well as sensor settings can be adjusted using the Wiki page commands and the bots are used to obtain data from the sensors while a series of commands are utilized to process it.

Yumang *et al.* (2017) described a ZigBee-based system that employs thermal imaging to monitor server room temperature and humidity in real-time. A microcontroller processes the data gathered by the infrared and humidity sensors which is sent to the computer via ZigBee. MATLAB software is then used to apply the color mapping and Bicubic algorithms and a buzzer alarm will be triggered in case the threshold value is reached.

Rasool *et al.* (2014) presented an RF based centralized System that monitors temperature, humidity, battery life and status, water leakage, smoke in server rooms. The daily data from the deployed sensors is transmitted using the Peripheral Interface Controller (PIC) to the Graphical User Interface (GUI) where it is stored and displayed to users in form of tables as well as visualizations.

Khan *et al.* (2018) proposed a physical and environmental monitoring system based on IoT that comprises of Raspberry Pis as smart objects and sensors located at various locations in the data centres to monitor temperature, humidity and atmospheric pressure. The proposed system uses LoRa gateways to link the smart objects to the server and activates alarms upon detection of any undesirable situation and this enables the data centre administrators to timely respond before additional damages transpire.

Nasution *et al.* (2019) highlighted the design and development of a remote temperature and humidity monitoring system that uses a small super computer in form of Lattepanda to which a DHT11 sensor is connected to read temperature and humidity values in the server room. The captured readings are sent to the cloud based ThingSpeak platform every 30 seconds for storage and visualization.

2.2 Technical Research Gap

From the related works, there are number of environmental monitoring systems for data centres or server rooms. However, they have no prediction/forecasting capabilities, most of them only monitor temperature and humidity, offer no redundancy and are limited in scalability. Therefore, there was a gap for developing scalable data centre environmental monitoring systems with prediction capabilities in addition to keeping track of other environmental parameters other than temperature and humidity that can aid in the planning and optimization of data centre resources.

2.3 Proposed System

The system was developed based on requirements gathered from various stake holders as illustrated in Fig. 1. It consists of a WSN for sensing, processing and storage of environmental parameters of temperature, humidity, smoke, water, voltage and current, machine learning models for forecasting temperature and humidity patterns and a web dashboard for remote monitoring and analysis of the sensed parameters. In addition, the system provides alerts to data centre operators or administrators in case any undesirable environmental conditions are detected. The developed system improves reliability of data centres as a result of decreased unplanned downtime, reduced risk of equipment damage and data loss.

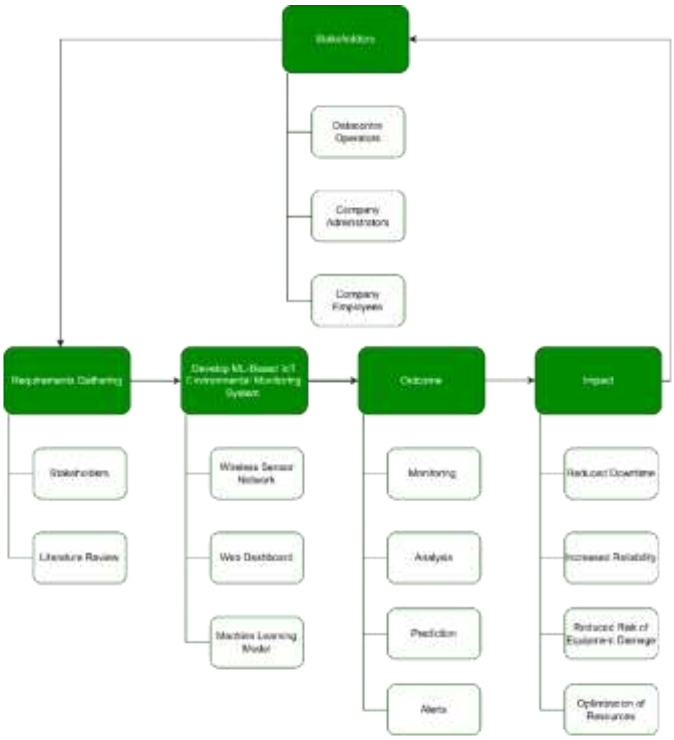


Figure 1: System conceptual framework

CHAPTER THREE

MATERIALS AND METHODS

3.1 Project Case Study

The area of the study was netlabs!UG, a research Centre of Excellence in the College of Engineering, Design, Art and Technology, Makerere University, Kampala, Uganda. netLabs!UG was selected due to the fact that it possesses a modern data centre facility that is open for use to researchers.

3.2 Research Methods

The project utilized a mixed method approach in the collection of both primary and secondary data relevant towards the implementation of the project study. This was done to better comprehend the environmental monitoring systems currently used in data centers and to discover extra inspirations for system design.

3.3 Target Population

This project targeted organizations with data centres or server rooms. The netlabs!UG data centre was used as the case study.

3.4 Sampling Techniques

The sampling technique utilized in this study was the simple random sampling where each person in the target population had an equal probability of being selected.

The sample size was calculated using the formula:

$$\text{Sample Size} = (Z^2 * P * (1 - P)) / E^2$$

where:

(a) Z is the z-score corresponding to the desired level of confidence

(b) P is the estimated proportion of data centers

(c) E is the desired margin of error.

3.5 System Requirements

The following are the techniques and tools that were used to determine the system requirements.

3.5.1 Data Collection Methods

Primary data was obtained through questionnaires, in-person interviews as well as observation while secondary data was acquired from journal papers, conference papers, research reports, textbooks and reputable websites.

3.5.2 Data Analysis

The Google form platform was used to create online questionnaires that were distributed to the target stakeholders. The Google form platform analyzed the submitted data by providing an overall overview of all respondents' responses, a breakdown of each respondent's individual responses, and an analysis of each question's response.

3.6 System Development Approach

The Extreme Programming (XP) agile methodology illustrated in Fig. 2 was employed in the development of the WSN and web dashboard. This was mainly due to its emphasis on time and cost savings through elimination of unproductive activities in addition to its flexibility as it accommodates continuous integration of changes (Qureshi, 2015).

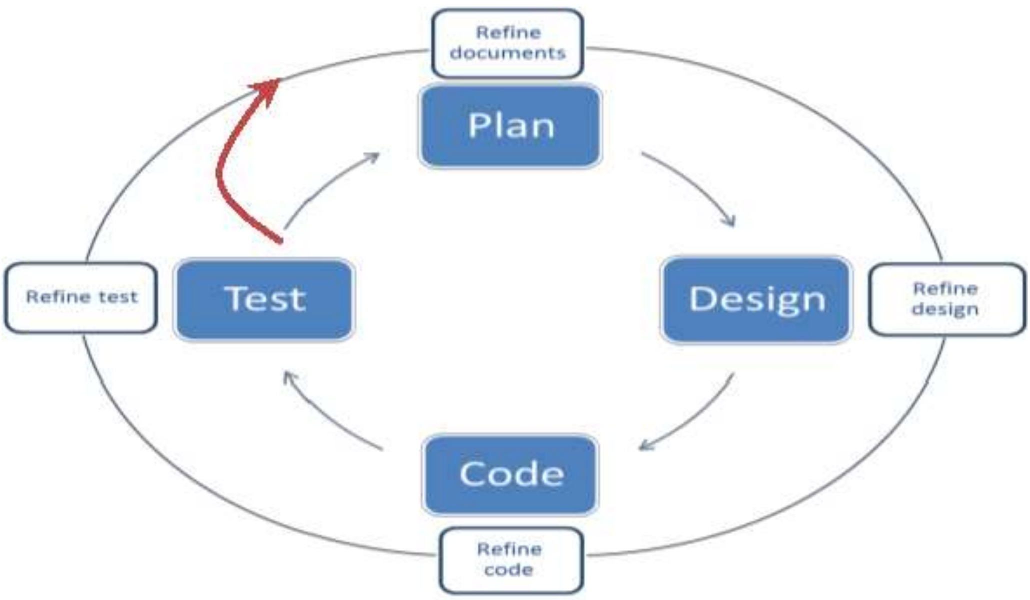


Figure 2: Extreme Programming (XP) agile methodology

During the development of the machine learning models, an iterative and incremental methodology was employed, dividing the task into smaller manageable components, such as data preprocessing, exploratory data analysis, modeling, and evaluation. This iterative approach allowed for continuous fine-tuning and refinement of the machine learning models, ensuring optimal performance and adaptation to changing requirements.

3.7 Theoretical Framework

The Technology Acceptance Model (TAM), introduced by Fred Davis in the late 1980s, is a well-established theoretical framework that helps explain and predict how users accept and adopt new technologies. Over time, TAM has been refined and expanded upon by various researchers, further cementing its significance in the field. At its core, TAM suggests that users' intention to adopt a technology is influenced by their perceptions of its usefulness (PU) and ease of use (PEOU) (Olushola & Abiola, 2017). In essence, users are more likely to accept and embrace a technology if they believe it will be beneficial to them and if they perceive it as user-friendly and easy to use.

In this project, the TAM was employed as a framework to evaluate user acceptance of the developed system, as depicted in Fig. 3. TAM was chosen for its effectiveness in describing system behavior and its ability to analyze user behavior during system implementation. It specifically focuses on factors such as ease of use and usefulness, which are crucial in determining user acceptance. Moreover, TAM is renowned for its track record of generating reliable and trustworthy results.

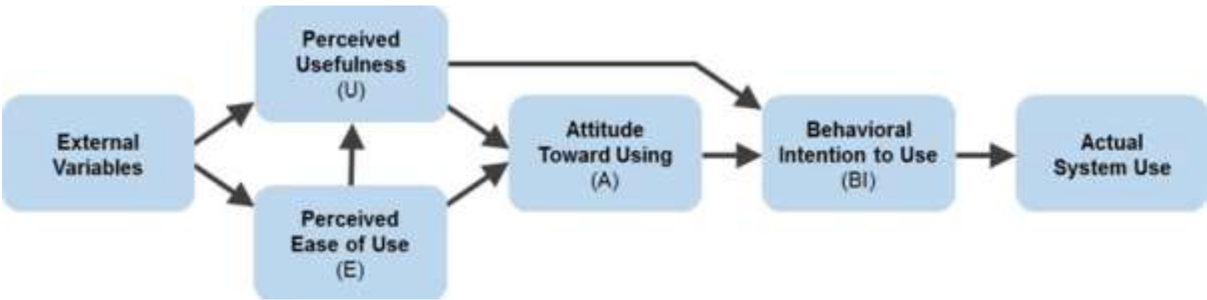


Figure 3: Technology acceptance model

3.8 System Development Materials and Tools

3.8.1 Hardware Materials Requirements

The major components necessary for the development of a smart environmental monitoring system for data centres using IoT and machine learning are as outlined in Table 1.

Table 1: System hardware components

S/N	Hardware Component	Specification
1.	Microcontroller Boards	ESP32
2.	Raspberry Pi	4B
3.	Temperature and Humidity Sensors	DHT 22
4.	Smoke Sensors	MQ2
5.	Water Level Sensor	Solu SL067
6.	Current Sensor	ACS721
7.	Voltage Sensor	ZMPT101B
8.	LCD Display Modules	20x4
9.	Active Piezo Buzzers	HDC 3-24V High Decibel
10.	I2C Serial Interface LCD Adapter	N/A
11.	Digital Multimeter	ZOYI ZT-S1
12.	3D Printer	Ender-3

(i) Raspberry Pi 4B

The Raspberry Pi 4B as shown in Fig. 4 was used in this project as the sink node where the data collected by the different sensor nodes was forwarded for local storage as well as processing.

The Raspberry Pi is a cheap, single-board, high performance Linux-based computer that allows controlling of physical computing electronic components and IoT devices through its set of General-Purpose Input Output (GPIO) pins.

The Raspberry Pi 4B in particular provides desktop performance with increased CPU clock speed, memory, network connectivity and multimedia performance in comparison with its predecessor the Raspberry Pi 3B+.

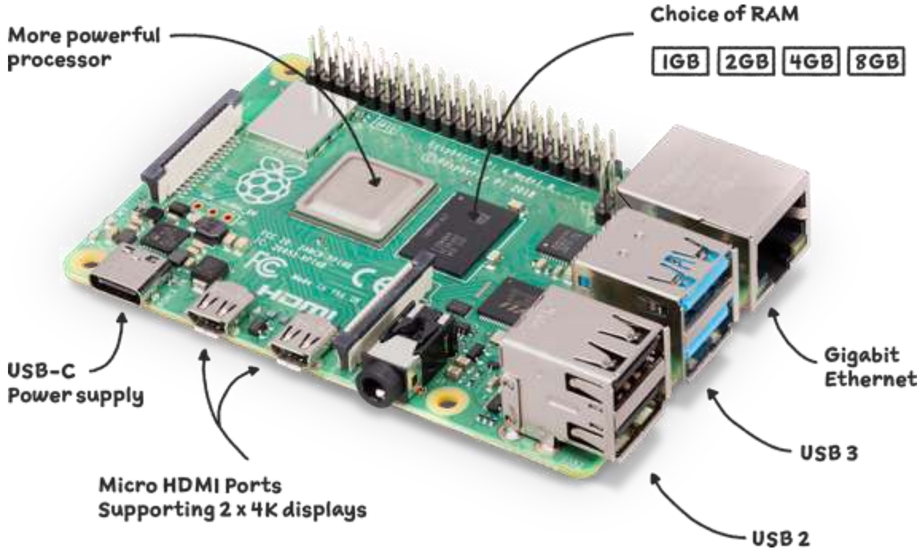


Figure 4: Raspberry pi 4B

(ii) ESP32 Microcontroller Board

The ESP32 was used as the microcontroller board in each of the sensor nodes where it was interfaced with various sensors capturing different environmental parameters (Fig. 5).

Figure 5: ESP32 microcontroller board

ESP32 is a 32-bit low-power, low-cost dual core microcontroller with inbuilt Wi-Fi and Bluetooth connectivity as well as a capacity of at least 30 usable GPIO pins (Fig. 6).

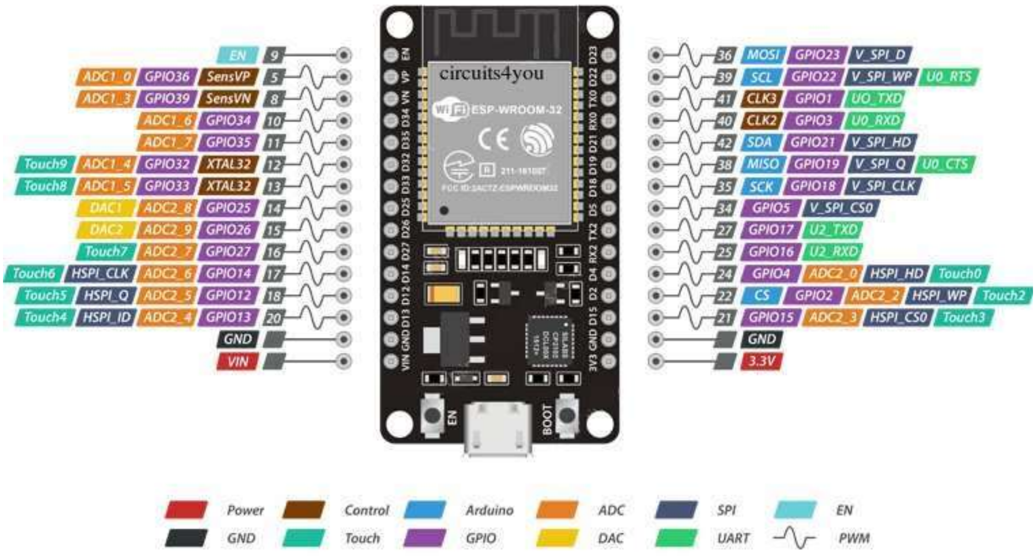


Figure 6: The ESP32 microcontroller pinout

(iii) Temperature and Humidity Sensor

The DHT22 sensors as shown in Fig. 7 were used to measure the ambient as well as rack temperature and humidity. It utilizes a thermistor and a capacitive humidity sensor to capture readings.

Two DHT sensors were utilized i.e., one for measuring ambient temperature and humidity while the other for measuring the rack temperature and humidity.

Figure 7: The DHT22 sensor

(iv) Smoke Sensor

The MQ2 sensor as shown in Fig. 8 was used to detect smoke in the data centre. It is a metal oxide semiconductor that utilizes a voltage divider network to detect concentration of gases such as smoke, LPG, propane, hydrogen, alcohol, carbon monoxide and methane with the range of 200 to 10000 parts per million (ppm).

Figure 8: MQ2 sensor

(v) Voltage Sensor

The ZMPT101B Voltage Sensor (Fig. 9) was used to measure the incoming voltage of the data centre power supply. The sensor utilizes the resistive divider design principle in measurement of voltage.

Figure 9: The ZMPT101B sensor

(vi) Current Sensor

The ACS712 sensor module as shown in Fig. 10 was used to measure the incoming current of the data centre power supply. It can measure AC as well as DC and is easy to interface with any microcontroller.

Figure 10: The ACS712 sensor

(vii) Water Level Sensor

The Solu SL067 sensor as shown in Fig. 11 was used to detect water leakage in the data centre. When exposed to water, the power and sensor traces combine to form a variable resistor similar to a potentiometer whose resistance varies accordingly.

Figure 11: Solu SL067 sensor

(viii) Liquid Crystal Display

The liquid crystal display (LCD) 20x4 display module (Fig. 12) was used to display the sensor values being measured on each sensor node and therefore provided a means of onsite monitoring of the different environmental parameters.

Figure 12: The LCD 20x4 display module

(ix) The I2C Serial Interface LCD Adapter

This is a device with a PCF8574 microcontroller chip that uses a two-wire communication protocol to interface with other micro-controller chips (Fig. 13). It enables the 20x4 LCD to be controlled by only two wires i.e., SCL and SDA. It conserves numerous microcontroller pins in addition to allowing LCD contrast control through its inbuilt potentiometer.

Figure 13: The I2C Serial Interface LCD Adapter

(x) Buzzer

A buzzer as shown in Fig. 14 was used to produce a sound alarm in case a predefined threshold is exceeded. A buzzer is an audio signaling device that converts audio signals to sound signals commonly used in timers, alarms, and computers.

Figure 14: Buzzer

(xi) Digital Multimeter

A digital multimeter is a test instrument used to measure electrical quantities, primarily voltage, current and resistance (Fig. 15). The ZOYI ZT-S1 digital multimeter was used in the calibration of the ZMPT101B voltage sensor and ACS712 module.

Figure 15: ZOYI ZT-S1 digital multimeter

(xii) The 3D Printer

A 3D printer is a device that makes it possible to produce physical objects from three-dimensional digital models, usually by applying successively numerous thin layers of a material (Fig. 16). The Ender-3 was used to make custom 3D casings to house the sensor nodes as well as the sink node.

Figure 16: Ender-3 3D printer

3.8.2 Software Tools Requirements

(i) Thonny IDE

It is a pre-built user-friendly Python Integrated Development Environment on Raspberry Pi that allows various ways of stepping through code as well as step by step evaluation of expressions and was used for programming the Raspberry Pi.

(ii) Arduino IDE

It is an open-source user-friendly development environment that comprises of a text editor, text console, message area and tool bar used for uploading of software programs to Arduino hardware as well as other microcontrollers like ESP32. It was used for writing, compiling and uploading C++ programs to the ESP32 microcontrollers.

(iii) MySQL

MySQL is a free widely used open-source relational database management system based on the standard Structured Query Language (SQL). It was used to store the readings or measurements from the different sensor nodes.

(iv) XAMPP

It is a free open-source solution stack for web development that consists of a database server (MySQL/MariaDB), web server (Apache) and scripting engines (PHP and Perl). The XAMPP was used to host the MySQL database on the sink node.

(v) Plotly Dash

This is a low-code python-based framework for rapidly building data driven interactive web dashboards. It is built on top of Plotly.js, React and React making it suitable for developing customized user interfaces. This was used to develop the web dashboard for displaying the different sensor readings.

(vi) Visual Studio Code

It is a free general purpose source code editor that supports numerous modern-day features such as intelligent code auto completion, debugging, Git version control, syntax highlighting and code refactoring. It was used for writing PHP scripts that were responsible for sending data from the sensor nodes to the sink node via HTTP as well as plotly dash code for developing the web dashboard.

(vii) JupyterLab

It is a flexible web based interactive development environment optimized for data science and machine learning. This was used in the development of the machine learning models for forecasting temperature and humidity patterns.

(viii) Google Colab

Google Colab is an online machine learning and data science tool that facilitates integration of python code, HTML, LaTex, charts and images into a sole document. This was also utilized in the development of the machine learning models for forecasting temperature and humidity trends.

(ix) Fusion 360

Fusion 360 is a software platform created by Autodesk that is used for Printed Circuit Board (PCB) design, 3D modelling, Computer Aided Design (CAD), Computer Aided Manufacturing (CAM) and Computer Aided Engineering (CAE). It was used for designing the PCBs of the sensor nodes as well as their respective 3D casings.

(x) IFTTT

The IFTTT stands for If This Then That. It is a web service that facilitates the connection of events using simple conditional statements. The IFTTT was used in the integration of the email and SMS alerts.

(xi) Callmebot WhatsApp Bot

It is an easy-to-use API that sends WhatsApp text messages from IoT devices, scripts, programs. It was used in the integration of the WhatsApp alerts.

3.9 System Testing

The system underwent various testing during the course of its development as follows:

(i) Unit testing was carried out whereby individual modules were independently scrutinized to ensure that they operated properly.

(ii) Integration testing was performed after unit testing. This involved combination of different modules and testing them as a whole.

(iii) After integration testing, system testing was carried out to evaluate if the system met the specified requirements.

3.10 System Validation

User Acceptance Testing (UAT) was executed by the end users to determine if the developed system met the user needs and ensured that it was viable in the real production environment.

Selected end users i.e., data centre operators, company administrators and employees were allowed to interact with the developed system and each was requested to fill in a user experience evaluation form whose feedback was used to validate the system.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Requirements Gathering

4.1.1 Overview

An online questionnaire was designed and distributed to Eight (8) institutions with data centres across Uganda. The questionnaire included a number of both multi-choice as well as open ended questions aimed at gathering relevant primary data that would improve the research study as illustrated in Appendix 1. Nineteen (19) personnel that frequently interact with data centres in the selected institutions took part in the survey as illustrated in Table 2.

Table 2: Target institutions and participant distribution

S/N	Name of institutions	Number of participants
1.	netLabs!UG	5
2.	Marconi Research and Innovation Lab	5
3.	CSquared	2
4.	Soliton Telmec	3
5.	Uganda Internet eXchange Point (UIXP)	1
6.	Directorate for ICT Support (DICTS) – Makerere University	1
7.	Inter-University Council of East Africa (IUCEA)	1
8.	Lira University	1
	Total	**19**

4.1.2 Questionnaire Findings

Figures 17 – 23 depict some of the questions in the online questionnaire and their corresponding analyzed results from the 19 participants.

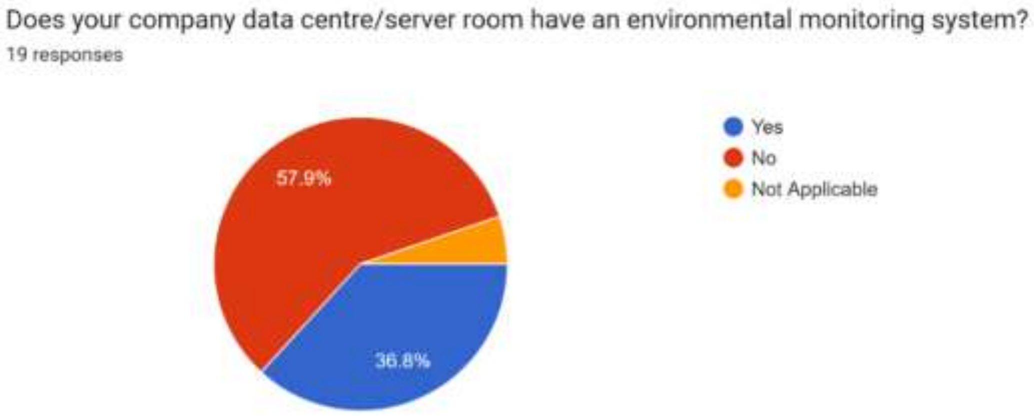


Figure 17: Response to whether their data centres had an environmental monitoring system

The majority, accounting for 57.9% of the participants, indicated that their company data centers did not have environmental monitoring systems. On the other hand, 36.8% of the participants confirmed that their company data centers were equipped with environmental monitoring systems. A smaller portion, comprising 5.3% of the participants, found the question not applicable to their situation. This highlights varying levels of adoption and awareness of environmental monitoring systems in company data centers, indicating room for improvement in implementation to ensure optimal conditions and mitigate risks.

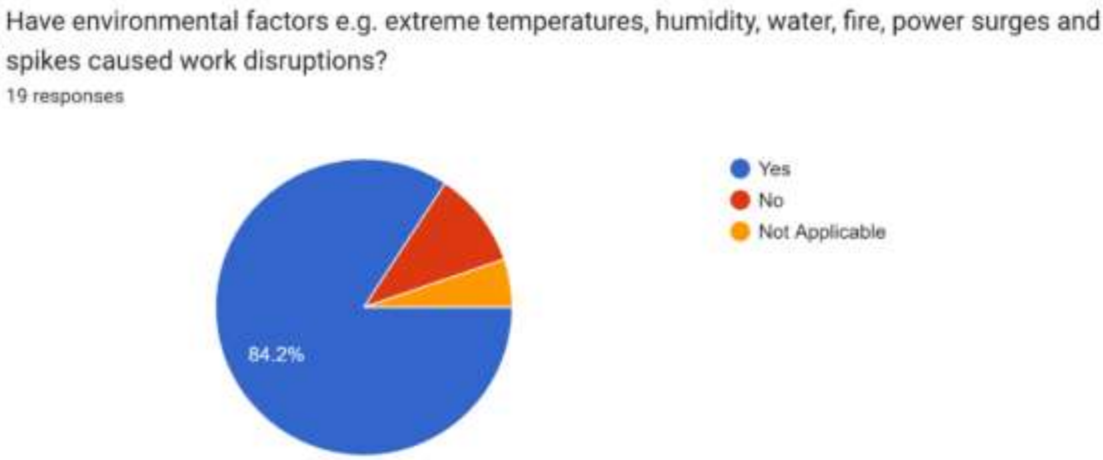


Figure 18: Response to whether environmental factors have ever caused work disruptions

A significant majority of participants, specifically 84.2%, recognized that environmental factors have caused work disruptions at some point. This suggests that environmental conditions have had an impact on their work environment, emphasizing the need to address and mitigate these issues for a more stable work environment.

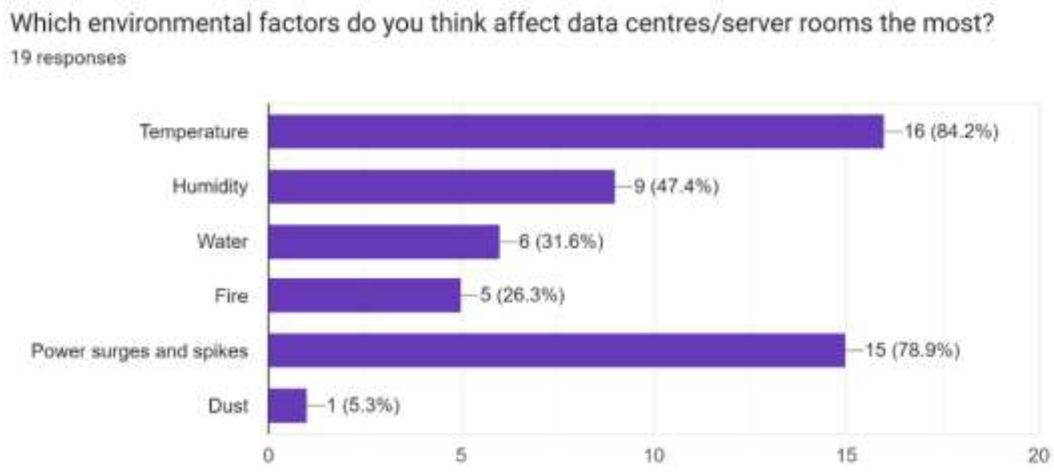


Figure 19: Response to the environmental factors thought to affect data centres the most

Temperature was identified by the majority of participants (84.2%) as the most influential factor affecting the data center, followed by power surges and spikes (78.9%) and humidity (47.4%). This emphasizes the need for effective monitoring and management of these factors.

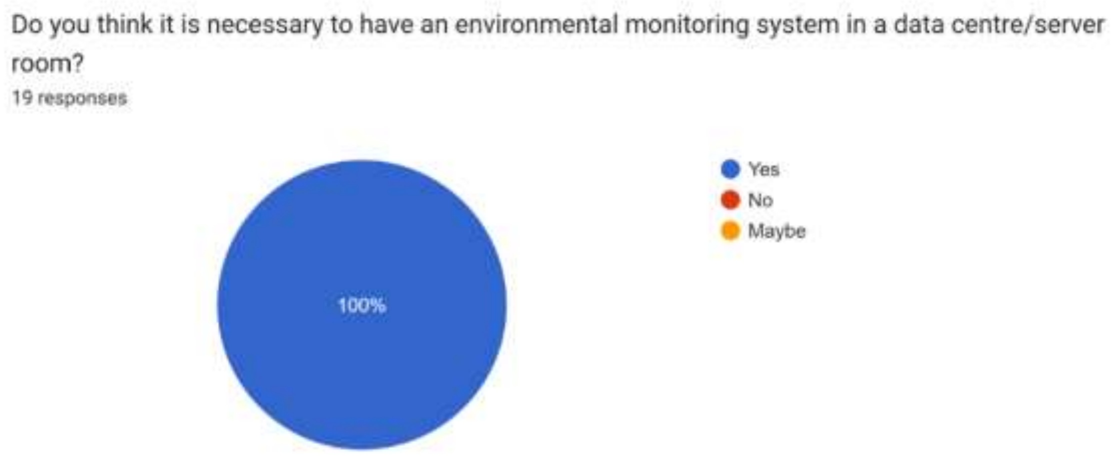


Figure 20: Response to whether it is necessary to have an environmental monitoring system in a data centre

Every participant recognized the importance of having an environmental monitoring system in their data centers. This highlights the need to prioritize their implementation for optimal performance, reliability, and safety.

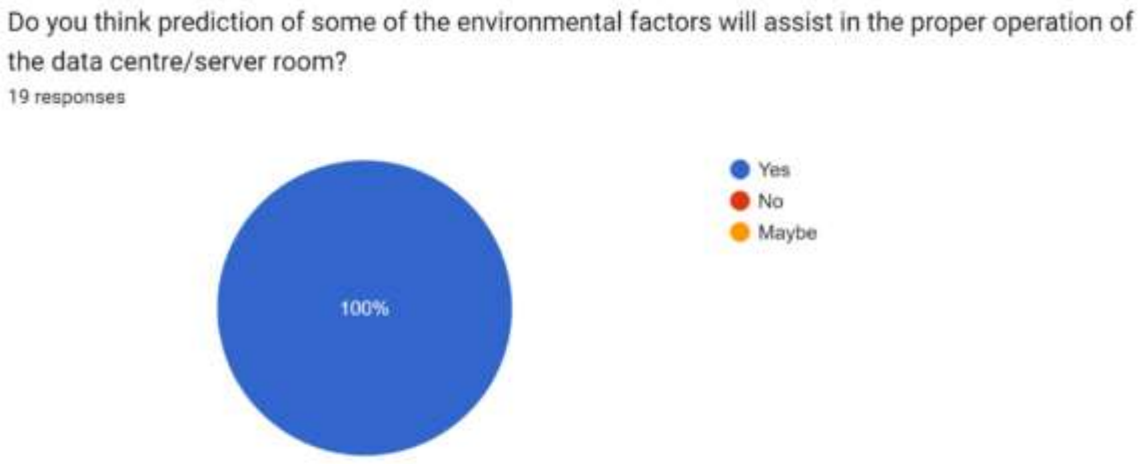


Figure 21: Response to whether prediction of some of the environmental factors would assist in the proper operation of the data centre

All participants acknowledged that prediction of some of the environmental factors would enhance the operation of data centres. This highlights the significance of leveraging predictive capabilities to anticipate and proactively address environmental conditions that may impact the performance, stability, and reliability of data center operations.

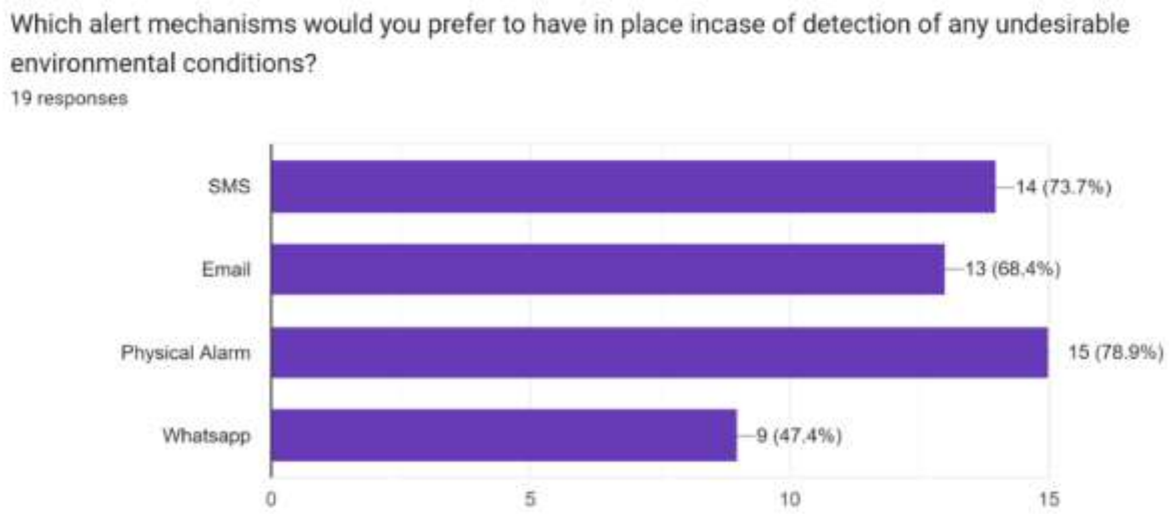


Figure 22: Response to the preferred alert mechanisms

The majority of participants (78.9%) favored a physical alarm as their preferred alert mechanism, followed by 73.7% who preferred SMS, 68.4% who favored email, and 47.4% who preferred WhatsApp. This suggests that there is still a preference for traditional notification methods that provide immediate and attention-grabbing alerts. However, it is also worth noting that a considerable portion of participants expressed a preference for digital communication channels such as SMS, email, and WhatsApp. This highlights the need for a versatile alert system that can cater to different preferences and ensure effective and timely notifications in data center environments.

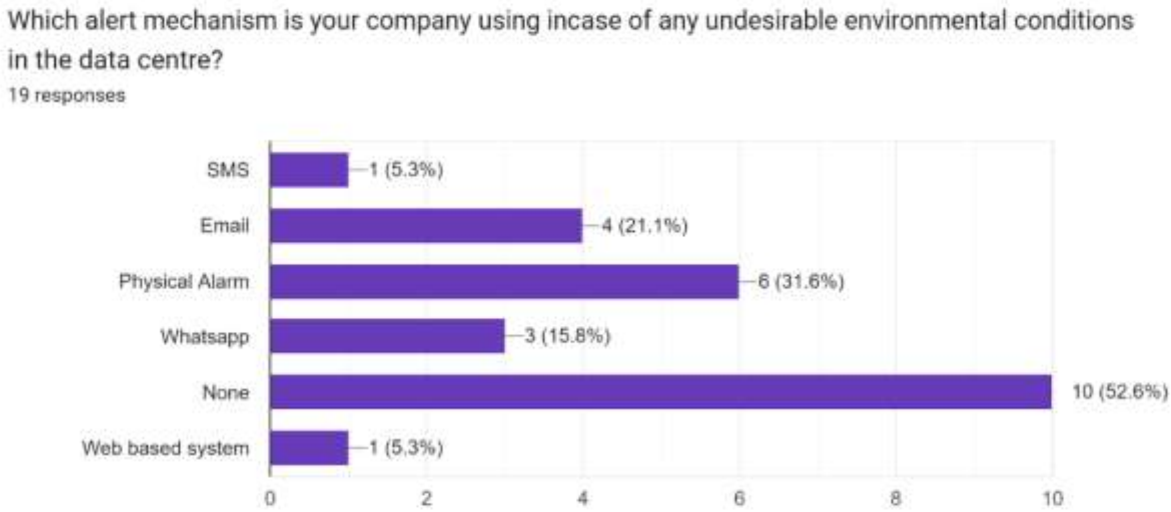


Figure 23: Response to the alert mechanisms their institutions were currently using

Among the participants, 52.6% reported the absence of an alert mechanism in their data centers for detecting undesirable environmental conditions. On the other hand, 31.6% utilize physical alarms, 21.1% rely on email, 15.8% utilize WhatsApp, and 5.3% employ SMS and a web-based system for alert notifications. This highlights a potential gap in the infrastructure of these data centers, as the absence of an alert mechanism can lead to delayed response and potential risks. The varying preferences for different alert mechanisms suggest the need for flexibility and customization in implementing alert systems that align with the specific requirements and preferences of data center operators.

4.1.3 Identified Requirements

The system requirements were gathered using an online questionnaire as described in the above sections and illustrated in Appendix 1. Both functional and non-functional requirements were identified as described in Table 3 and Table 4.

Table 3: Identified functional requirements

S/N	Functional Requirement	Description
1.	Measure temperature and humidity	The system measures ambient as well as rack temperature and humidity within a data centre
2.	Detect smoke	The system detects presence of smoke with the data centre premises
3.	Detect water leakage	The system detects water leakage within the data centre
4.	Measure voltage and current	The system measures the voltage and current of the data centre power supply
5.	Store data	The system stores data both locally and to the cloud
6.	Provide alerts	The system provides alerts to stakeholders upon detection of any undesirable conditions in the data centre
7.	Forecast temperature and humidity patterns	The system predicts future temperature and humidity trends

Table 4: Identified non-functional requirements

S/N	Non-Functional Requirement	Description
1.	Reliability and availability	The system is readily available and operates with similar efficiency after extensive use
2.	Performance	The system is relatively efficient and effective
3.	Security	The system data is stored in a secure database
4.	Scalability	The system allows integration of more sensor nodes while maintaining similar performance
5.	Usability	The system is usable by end users with minimum instructions

4.2 System Design

4.2.1 System Architecture

The system comprises of a heterogeneous WSN with four (4) sensor nodes and a sink node as illustrated in the system architecture in Fig. 24.

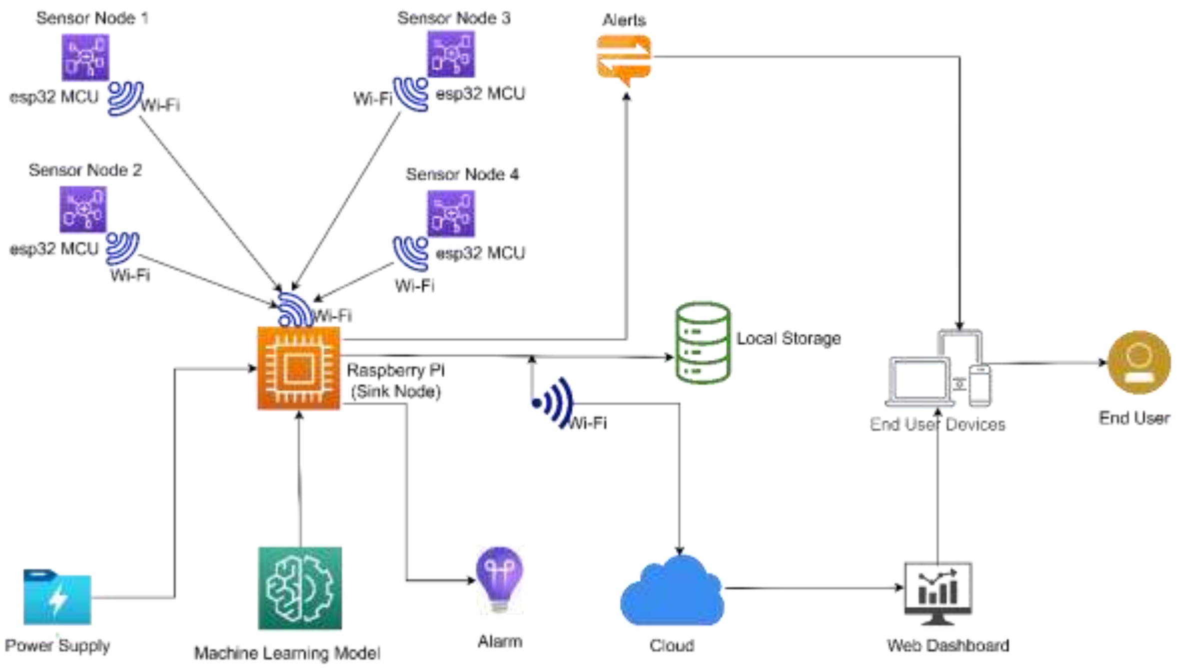


Figure 24: System architecture describing conceptual interactions of the sensor nodes, sink node and end user

The sensor nodes measure various environmental parameters, including temperature, humidity, smoke, water, current, and voltage. The sensed readings are wirelessly transmitted to the sink node for local storage in a database. Machine learning models deployed on the sink node are utilized to predict future temperature and humidity patterns. In case any undesirable condition is detected, alerts are sent to data center administrators. Additionally, the stored readings are sent to the cloud for logging purposes and real-time visualizations of the sensed environmental parameters as well as forecasts are provided through a web dashboard.

4.2.2 System Block Diagram

The system was further decomposed into five (5) components or units as demonstrated in Fig. 25.

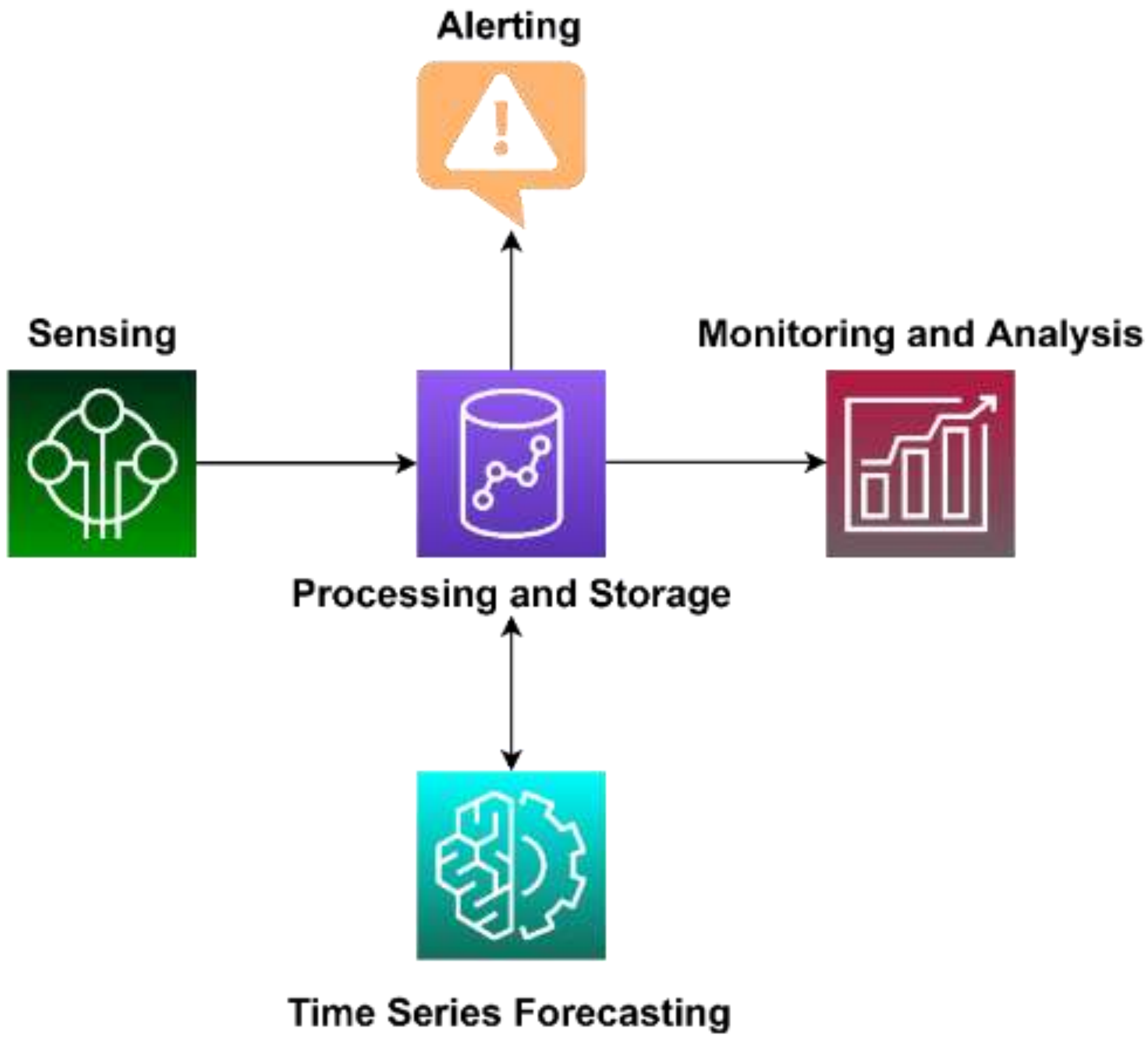


Figure 25: Block diagram defining interrelations among system units

The sensing unit comprises of sensor nodes, responsible for measuring environmental parameters of temperature, humidity, smoke, water, current, and voltage. The processing and storage unit is responsible for both local and cloud storage of the readings obtained from the sensing unit. It is also responsible for the deployment of the machine learning models from the time series forecasting unit. The monitoring and analysis unit encompasses a web dashboard that offers real-time visualizations of the sensed parameters, showcasing their various trends and patterns. The alerting unit is designed to trigger a sound alarm and send alert notifications to data center administrators whenever any unfavorable environmental condition is detected. Lastly, the time series forecasting unit incorporates machine learning models dedicated to predicting future temperature and humidity patterns.

4.2.3 Sensor Nodes Circuit Diagrams

The circuit diagrams of the sensor nodes were designed using fritzing software and kapwing online design platform and are illustrated in Figs. 26, 27 and 28.

(i) Sensor Node 1 and 2

The ESP32 microcontroller board is connected to a DHT22 sensor, MQ2 sensor, 20x4 LCD, and buzzer, using a breadboard. The VCC and GND pins of the DHT22 sensor and MQ2 sensor are connected to the respective 3.3 V and GND pins of the ESP32. The DATA pin of the DHT22 sensor is connected to GPIO pin 2, while the AOUT pin of the MQ2 sensor is connected to GPIO pin 4. An I2C LCD adapter is used to simplify the connection to the 20x4 LCD, where the VCC pin is connected to the 5V pin, the GND pin to the GND pin, and the SDA and SCL pins to the corresponding pins on the ESP32. Lastly, the negative terminal of the buzzer is connected to the GND pin, and the positive terminal is connected to GPIO pin 18.

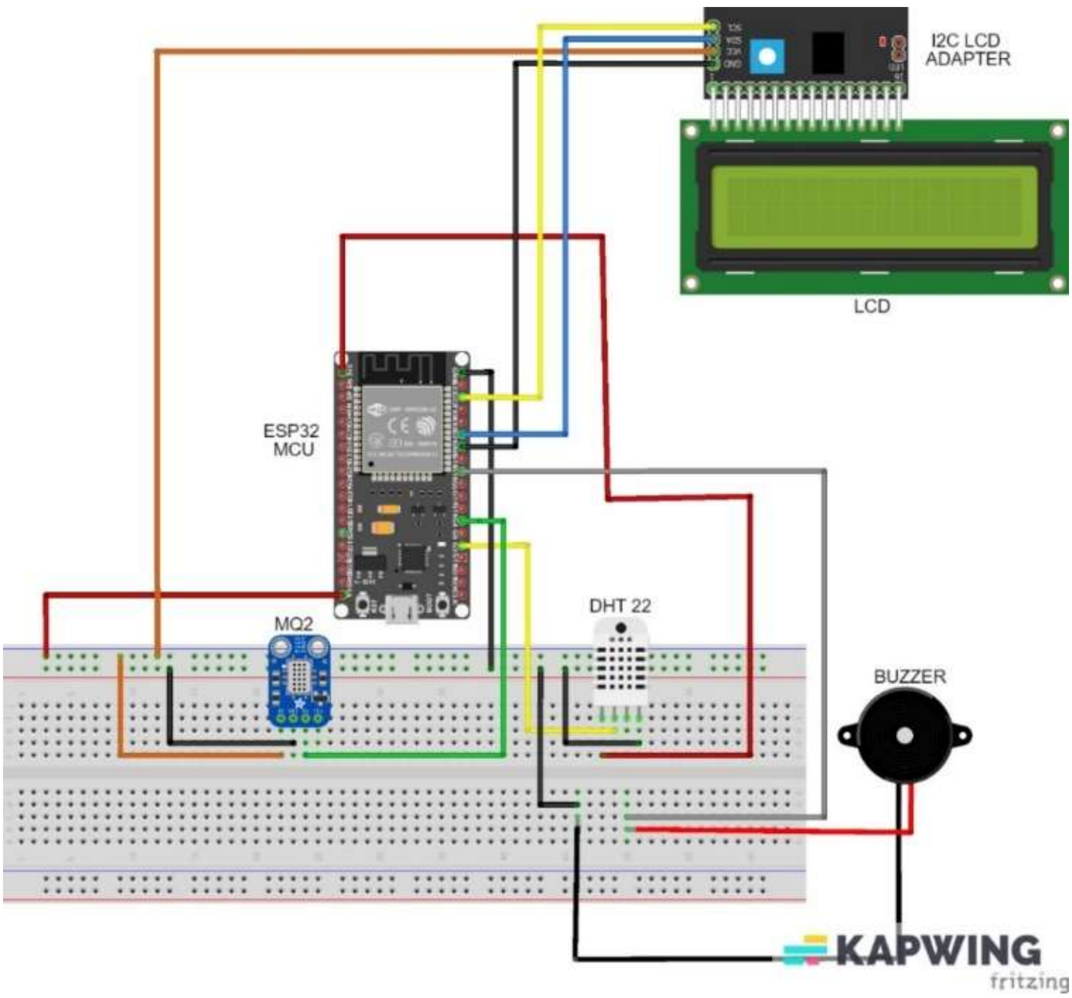


Figure 26: Sensor node 1 and sensor node 2 circuit diagram

(ii) Sensor Node 3

The ESP32 microcontroller board is connected to a Solu SL067 water level sensor, 20x4 LCD, and buzzer, using a breadboard. The power pin of the water level sensor is connected to GPIO pin 17 of the ESP32. The GND pin of the water level sensor is connected to the respective GND pin of the ESP32. The signal pin of the water level sensor is connected to GPIO pin 35. An I2C LCD adapter is used to simplify the connection to the 20x4 LCD, where the VCC pin is connected to the 5 V pin, the GND pin to the GND pin, and the SDA and SCL pins to the corresponding pins on the ESP32. Finally, the negative terminal of the buzzer is connected to the GND pin, and the positive terminal is connected to GPIO pin 18.

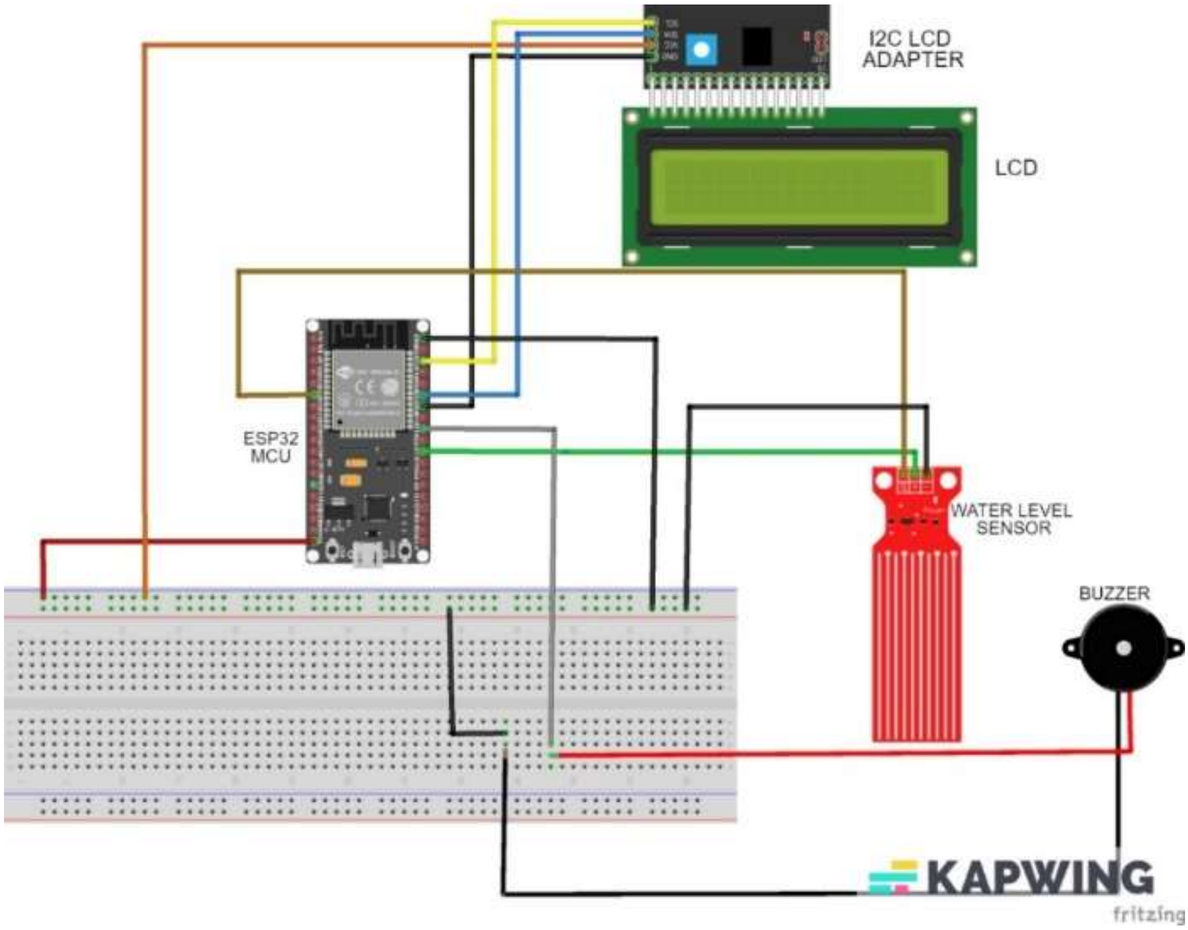


Figure 27: Sensor node 3 circuit diagram

(iii) Sensor Node 4

The ESP32 microcontroller board is connected to a ZMPT101B sensor, ACS721 sensor, 20x4 LCD, and buzzer, using a breadboard. The VCC and GND pins of the ZMPT101B sensor and ACS721 sensor are connected to the respective 3.3V and GND pins of the ESP32. The VOUT pin of the ZMPT101B sensor is connected to GPIO pin 35, while the OUT pin of the ACS721 sensor is connected to GPIO pin 34. An I2C LCD adapter is used to simplify the connection to the 20x4 LCD, where the VCC pin is connected to the 5 V pin, the GND pin to the GND pin, and the SDA and SCL pins to the corresponding pins on the ESP32. Lastly, the negative terminal of the buzzer is connected to the GND pin, and the positive terminal is connected to GPIO pin 18.

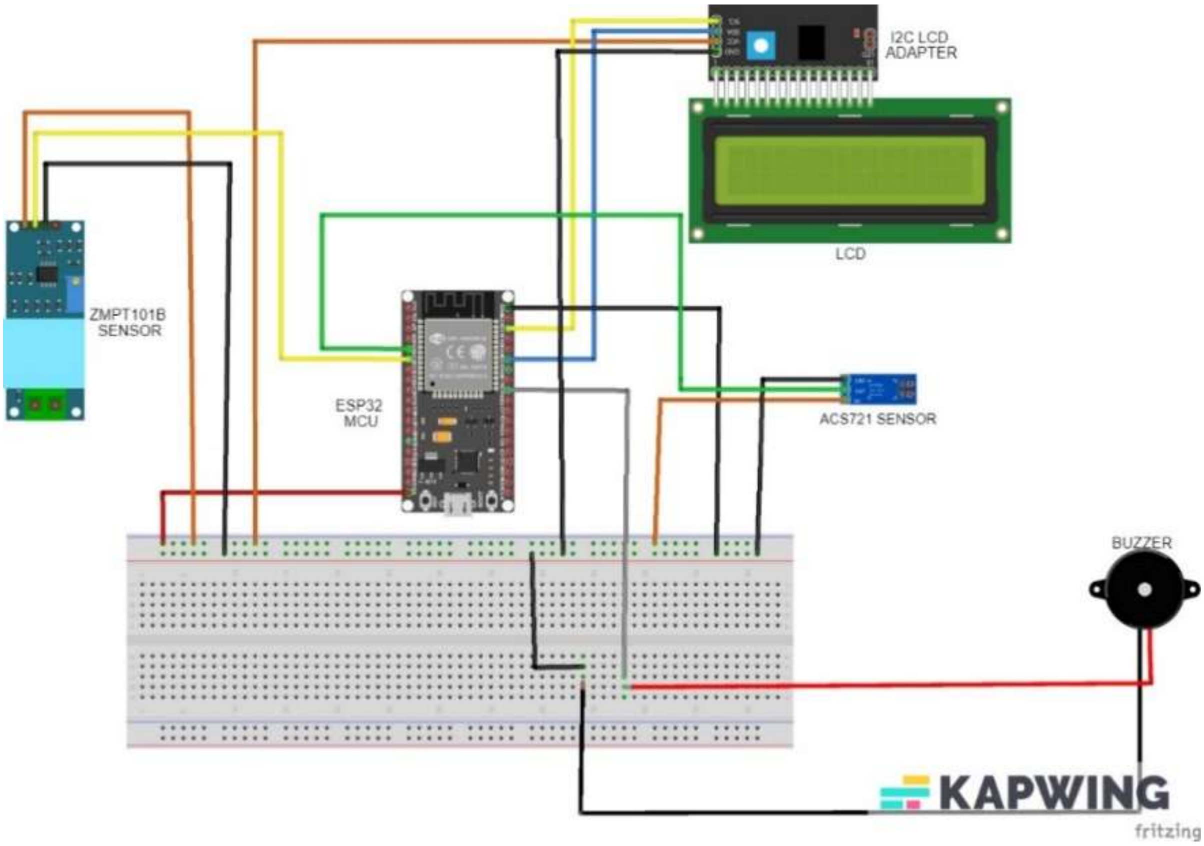


Figure 28: Sensor node 4 circuit diagram

4.2.4 PCB Schematic Design

The schematic and layout designs of the sensor nodes' PCBs were designed using Fusion 360 software as illustrated from Fig. 29 to Fig. 34. The connections depicted in the PCB schematic and layout for the different sensor nodes correspond precisely to those described in the circuit diagrams for the respective nodes. This was to ensure a consistent representation of the circuit connections between the circuit diagrams and the actual PCB design.

(i) Sensor Node 1 and 2

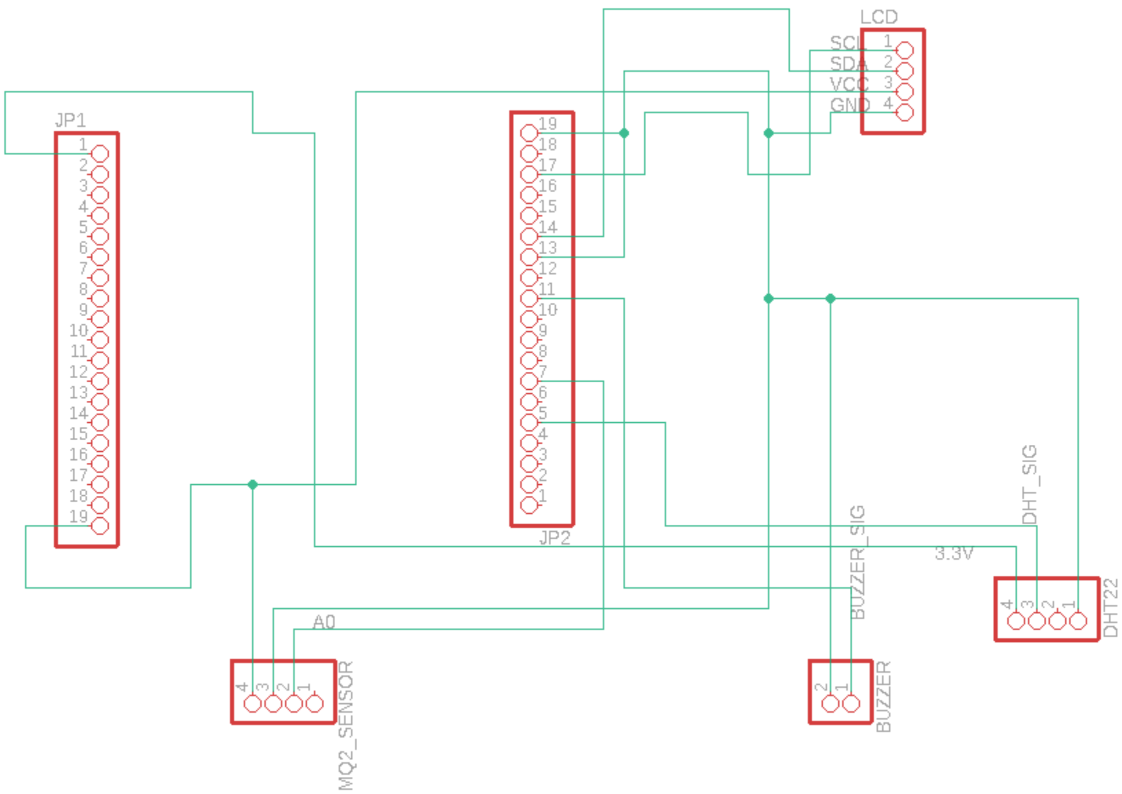


Figure 29: Sensor node 1 and sensor node 2 PCB schematic diagram

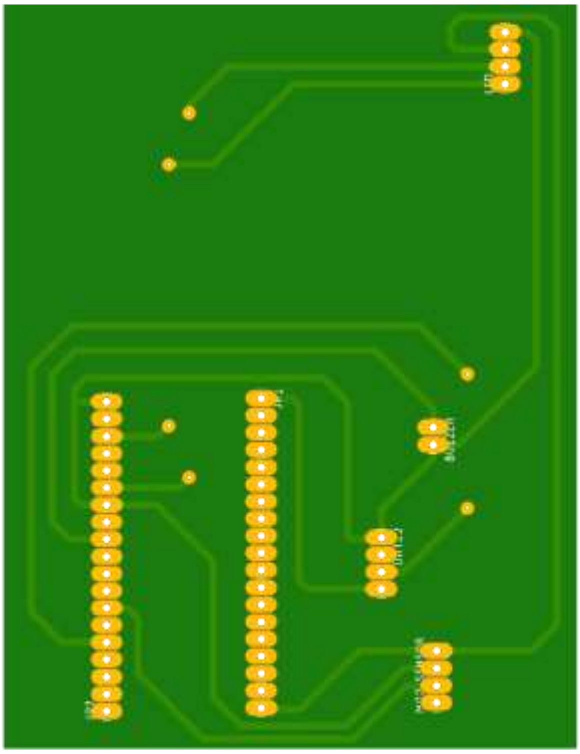


Figure 30: Sensor node 1 and sensor node 2 PCB layout

(ii) Sensor Node 3

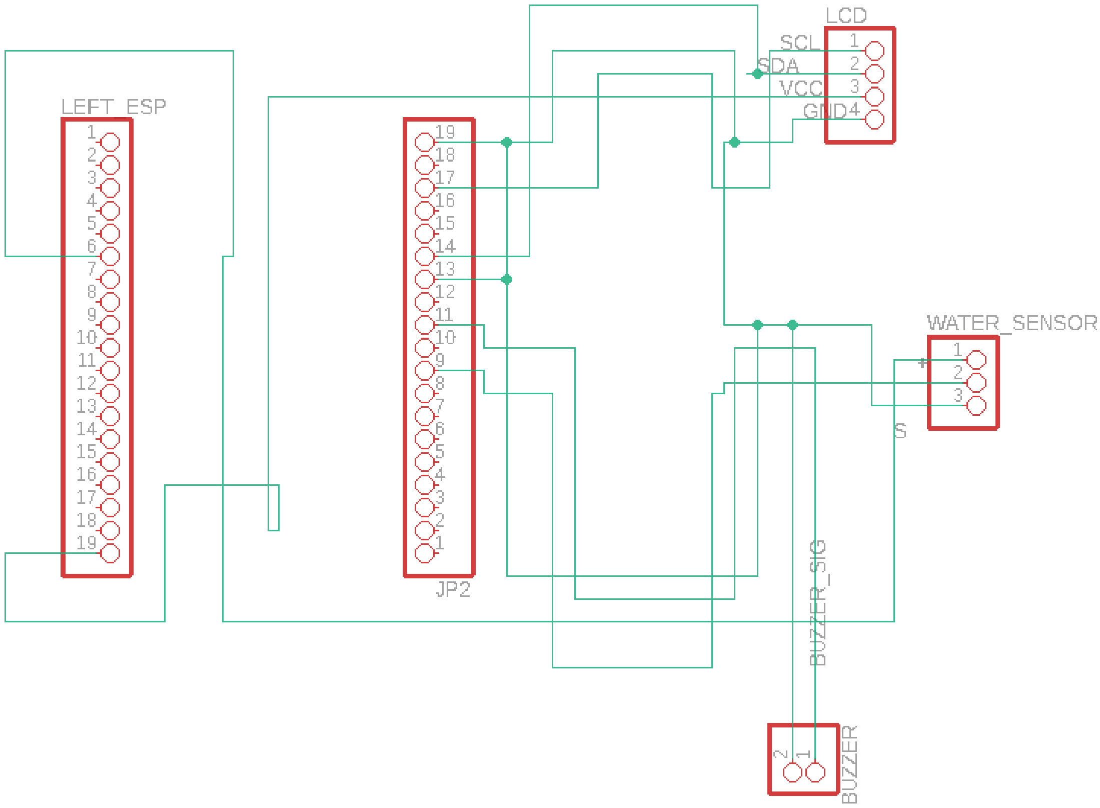


Figure 31: Sensor node 3 PCB schematic diagram

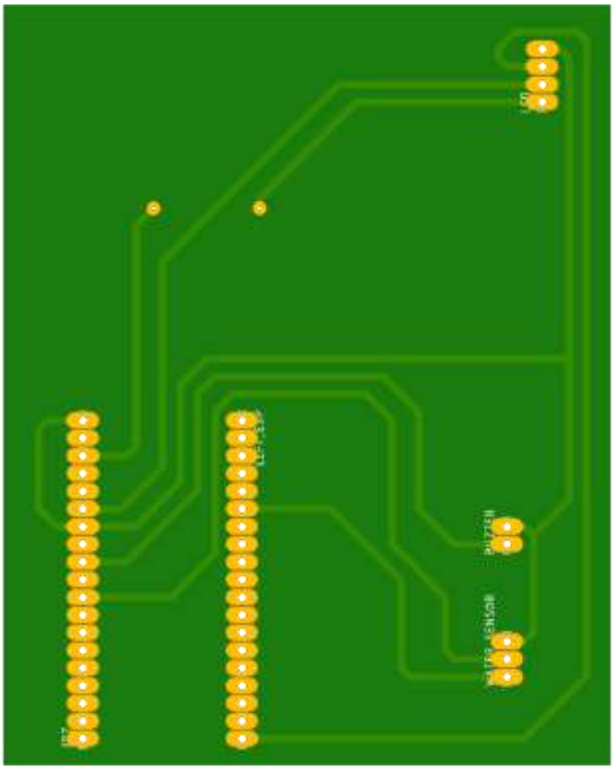


Figure 32: Sensor node 3 PCB layout

(iii) Sensor Node 4

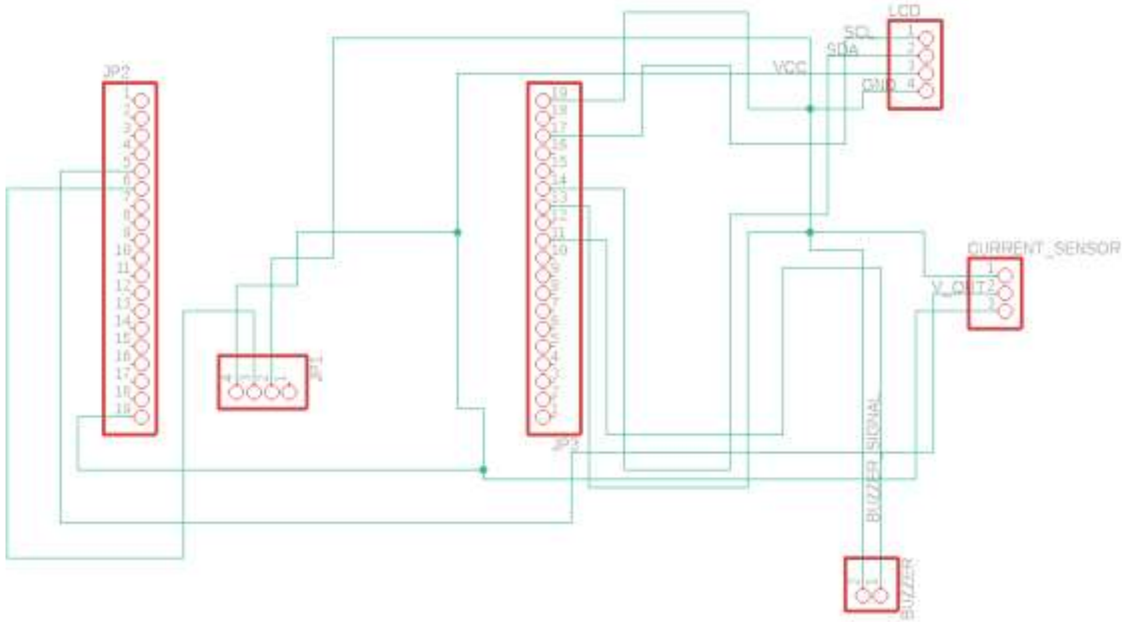


Figure 33: Sensor node 4 PCB schematic diagram

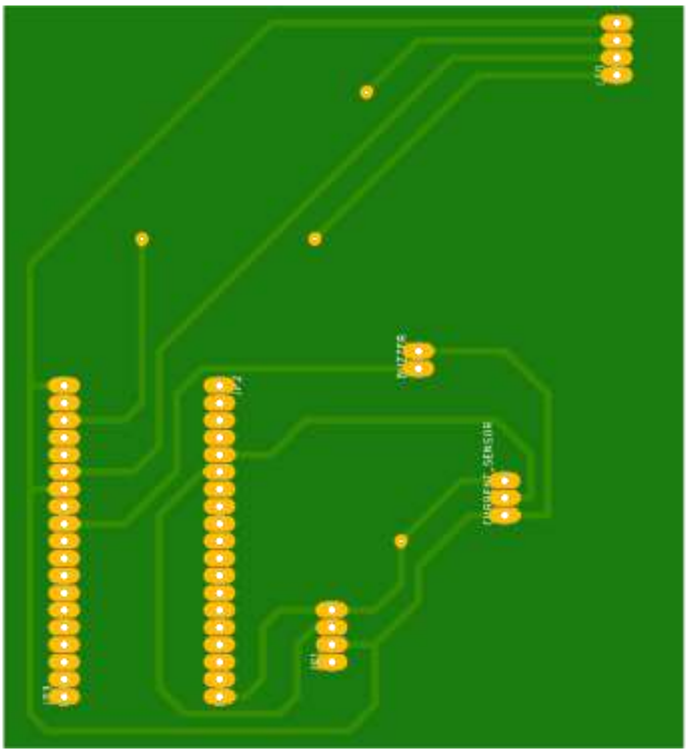


Figure 34: Sensor node 4 PCB layout

4.2.5 3D Model Design

The 3D casings of the sensor nodes and sink node were designed using Fusion 360 software as illustrated from Figs. 35, 36 and 37.

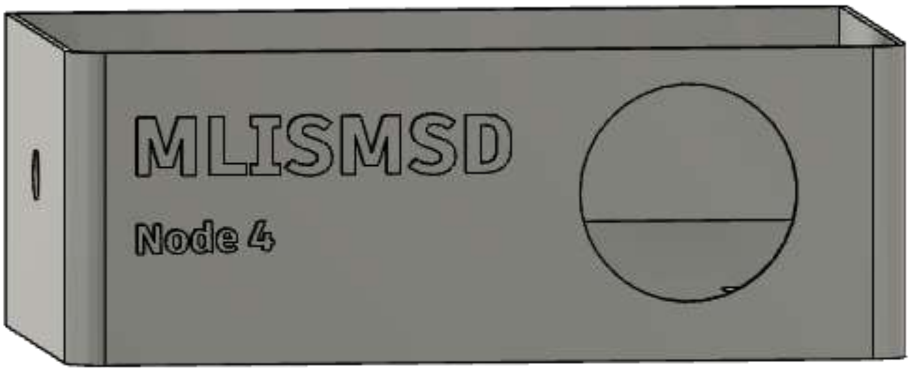


Figure 35: Sensor node 4 3D model layout

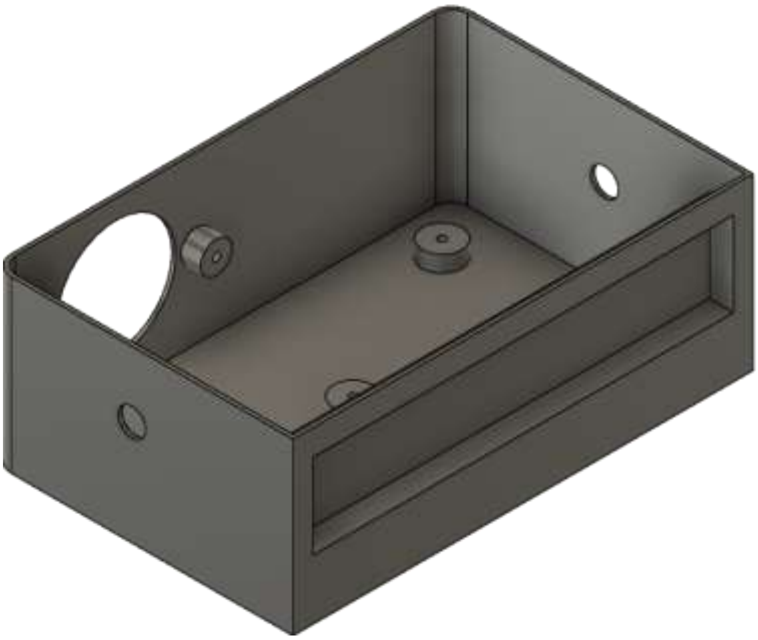


Figure 36: Sensor node 3 3D model layout

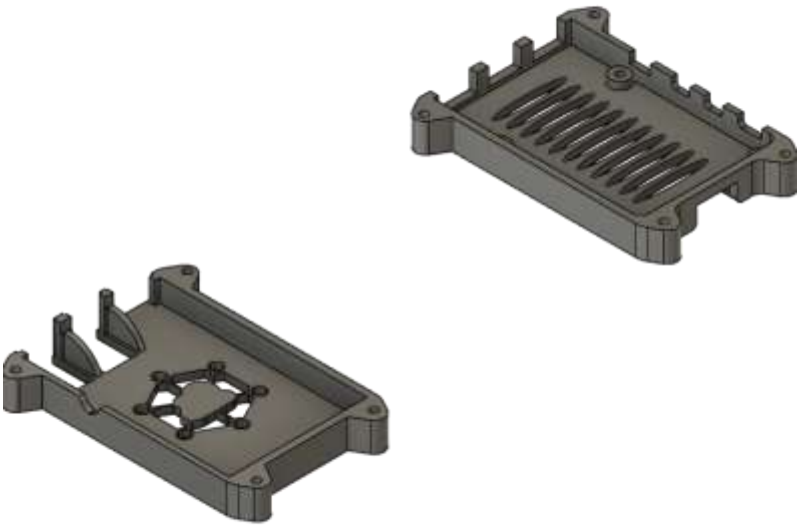


Figure 37: Sink node 3D model layout

4.3 Development of an IoT Environmental Monitoring System

4.3.1 Overview

The IoT environmental monitoring system consists of a sensing unit, processing and storage unit, monitoring and analysis unit as well as an alerting unit.

4.3.2 Sensing Unit

The sensing unit consists of four sensor nodes that were set up and configured as described below. The sensor nodes were initially developed using breadboards but were later fabricated on PCBs and housed in customized 3D casings.

(i) Sensor Node 1 and Sensor Node 2

Sensor Node 1 and Sensor Node 2 as illustrated in Figs. 38 and 39 are identical where the former is used for rack monitoring while the latter is used for ambient monitoring. They are composed of an ESP32 microcontroller, a DHT22 sensor, an MQ2 sensor, a 20x4 LCD display module and a buzzer.

The DHT22 measures temperature and humidity readings, the MQ2 sensor detects presence of smoke, the 20x4 LCD display module displays sensor readings and the buzzer produces an audio alarm in case (a) either temperature or humidity is outside the optimal data centre range (temperature below 15°C or above 32°C and humidity below 20% or above 80%) (Standards and Guidelines, 2019), or (b) if smoke is detected.

Figure 38: Sensor node 1 upper view

Figure 39: Sensor node 1 PCB configuration

(ii) Sensor Node 3

This sensor node consists of an ESP32 microcontroller, a water level sensor for detecting the presence of water, a 20x4 LCD display module for displaying sensor readings and a buzzer to produce an audio alarm in case water is detected as illustrated in Figs. 40 and 41.

Figure 40: Sensor node 3 upper view

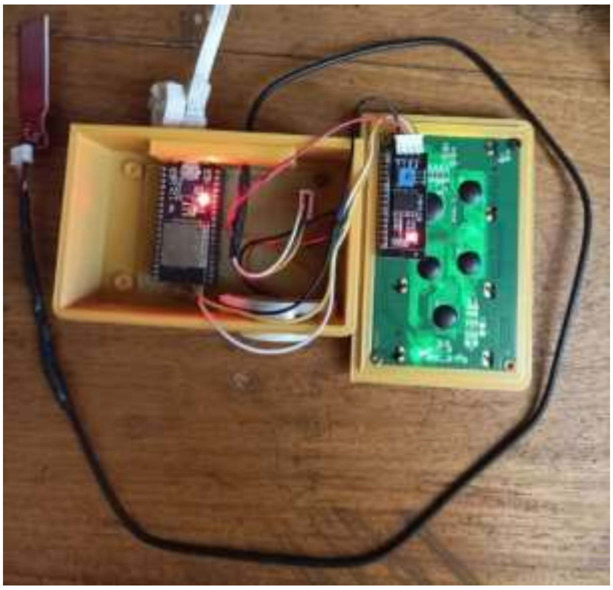


Figure 41: Sensor node 3 PCB configuration

(iii) Sensor Node 4

It is composed of an ESP32 microcontroller, a ZMPT101B voltage sensor, an ACS721 current sensor module, a 20x4 LCD display module and a buzzer as illustrated in Figs. 43 and 44.

The ZMPT101B voltage sensor measures input AC voltage, the ACS721 current sensor module measures input AC current, the 20x4 LCD display module displays sensor readings and the buzzer produces an audio alarm in case of voltage spikes (voltage exceeds 245 V).

A digital multimeter was used in calibration of the ZMPT101B voltage sensor whose potentiometer was adjusted until sensor readings matched those of the multimeter as illustrate d in Fig. 42.

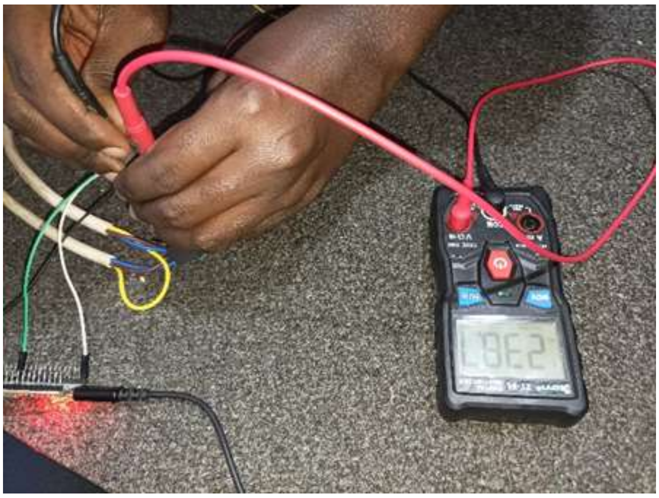


Figure 42: Calibrating voltage sensor using a digital multimeter

Figure 43: Sensor node 4 upper view

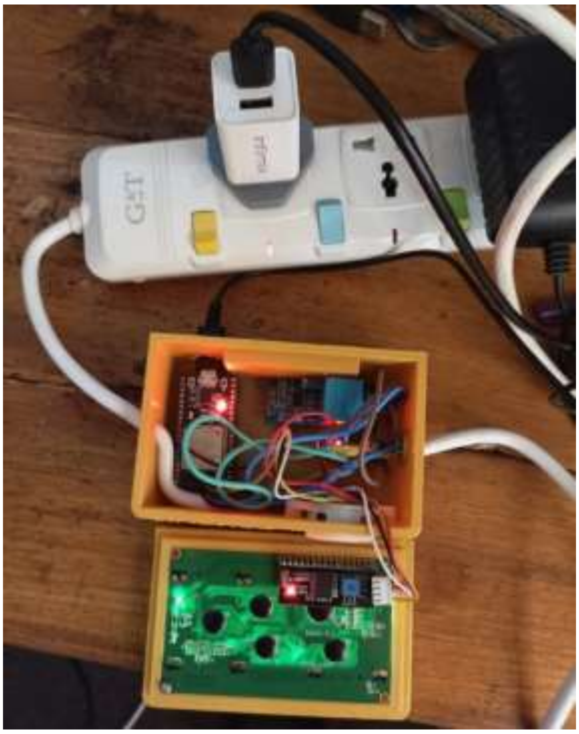


Figure 44: Sensor node 4 PCB configuration

4.3.3 Processing and Storage Unit

A Raspberry Pi 4 illustrated in Fig. 45 was set up as the sink node. The sensor nodes send or publish their readings to a local MySQL database via Wi-Fi by making an HTTP POST request to the Raspberry Pi LAMP server. Furthermore, the machine learning models responsible for forecasting temperature and humidity trends are deployed on the sink node.

Figure 45: Raspberry pi sink node in customized 3D casing upper view

The sensor readings are inserted into the database every five (5) minutes as shown in Fig. 46 and the database is backed up to the local SD card of the Raspberry Pi as well as Google Drive after every 24 hours using the SqlBak service as illustrated in Fig. 47.

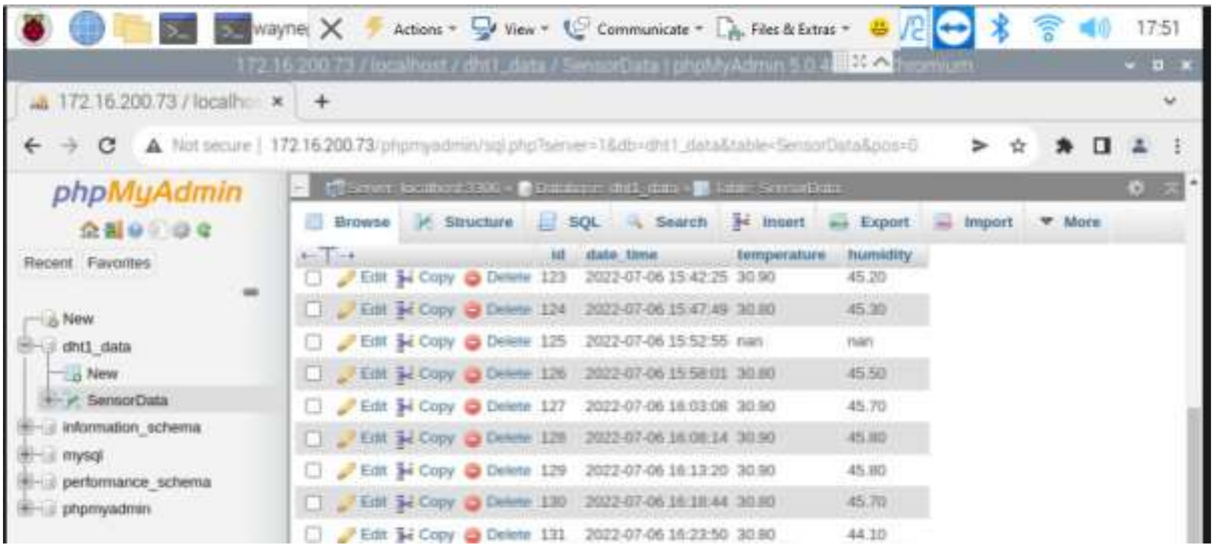


Figure 46: Sample data stored in the MySQL database

Figure 47: Confirmation message of successful database backup using SqlBak

A simple web page illustrated in Fig. 48 was designed to display the readings stored in the database and is updated once a new record is stored. The web page can be accessed across the local network and enables visualization of the stored sensor readings without necessarily logging into the database.

ID	Date_Time	Temperature(°C)	Humidity
149	2022-07-06 17:56:22	30.50	42.50
148	2022-07-06 17:51:15	30.50	42.50
147	2022-07-06 17:46:06	30.50	42.60
146	2022-07-06 17:40:56	25.50	25.50
145	2022-07-06 17:35:51	30.50	42.10
144	2022-07-06 17:30:40	30.50	43.50
143	2022-07-06 17:25:26	30.50	43.90
142	2022-07-06 17:20:18	30.60	43.90
141	2022-07-06 17:15:11	30.50	43.90
140	2022-07-06 17:10:05	30.60	42.00
139	2022-07-06 17:04:55	30.60	43.80
138	2022-07-06 16:59:48	30.60	43.70
137	2022-07-06 16:54:42	30.60	43.30
136	2022-07-06 16:49:36	30.70	43.10
135	2022-07-06 16:44:26	30.70	43.00
134	2022-07-06 16:39:16	30.70	44.90
133	2022-07-06 16:34:09	30.70	44.60

Figure 48: Web page displaying sensor readings stored in the database

The Raspberry Pi can be remotely accessed through Secure Shell (SSH), Virtual Network Computing (VNC) and TeamViewer. SSH and VNC can be used to access it while it is on the same network with the connecting device while TeamViewer can be used to access it from anywhere across the globe as long as both the Raspberry Pi and the connecting device have an Internet connection.

Figure 49: Raspberry pi TeamViewer interface

4.3.4 Alerting Unit

Each sensor node consists of a buzzer that produces an audio alarm in case an undesired event is detected. In addition, sensor nodes are integrated with email, SMS and WhatsApp alerts that are sent to data centre administrators in the event of unfavorable environmental conditions in the data centre as illustrated from Fig. 50 to Fig. 64.

The IFTTT web service was used in the email and SMS alert integration while Callmebot WhatsApp bot API was used for the WhatsApp alert integration.

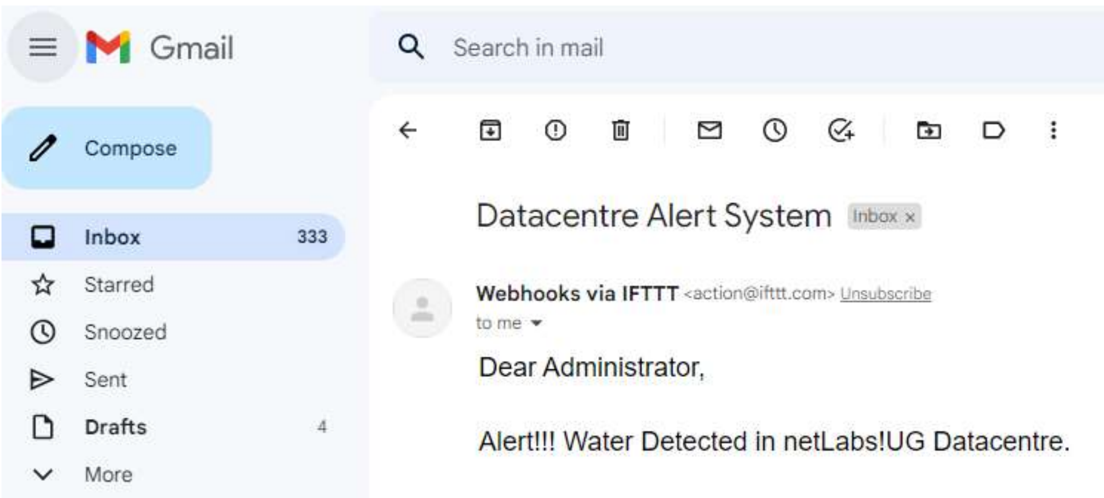


Figure 50: Water alert email notification

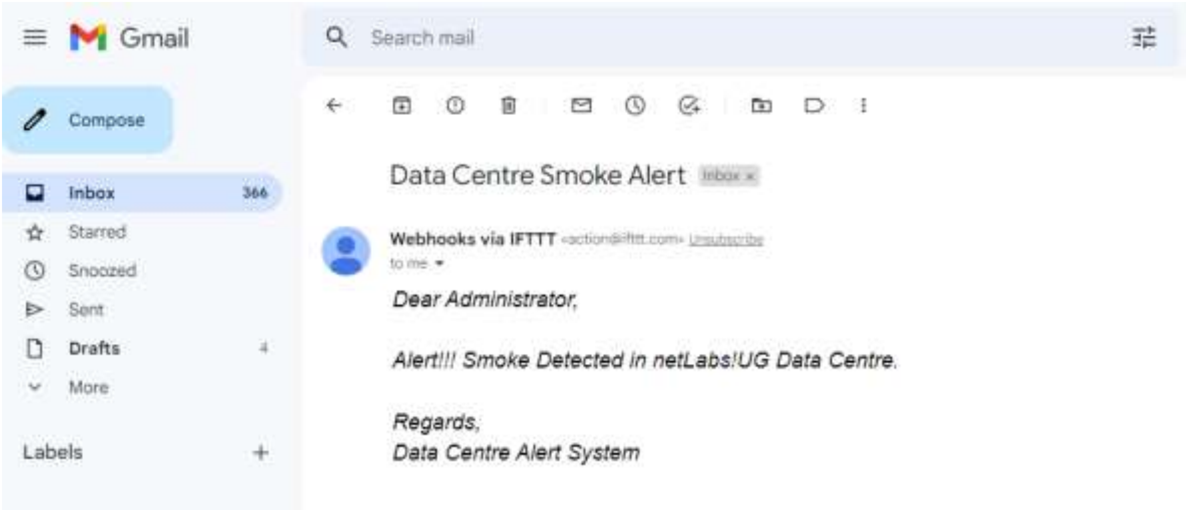


Figure 51: Smoke alert email notification

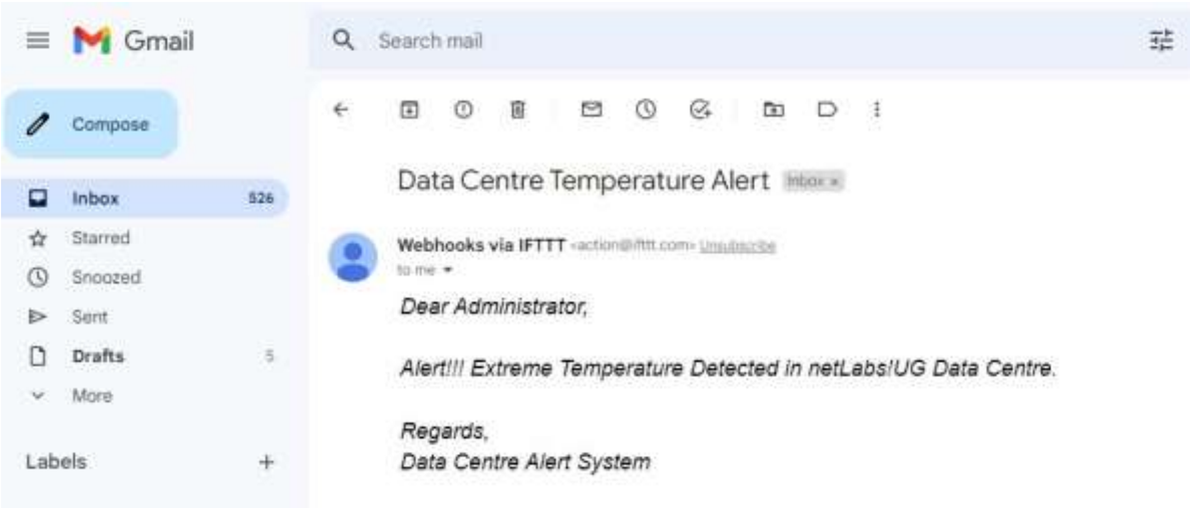


Figure 52: Temperature alert email notification

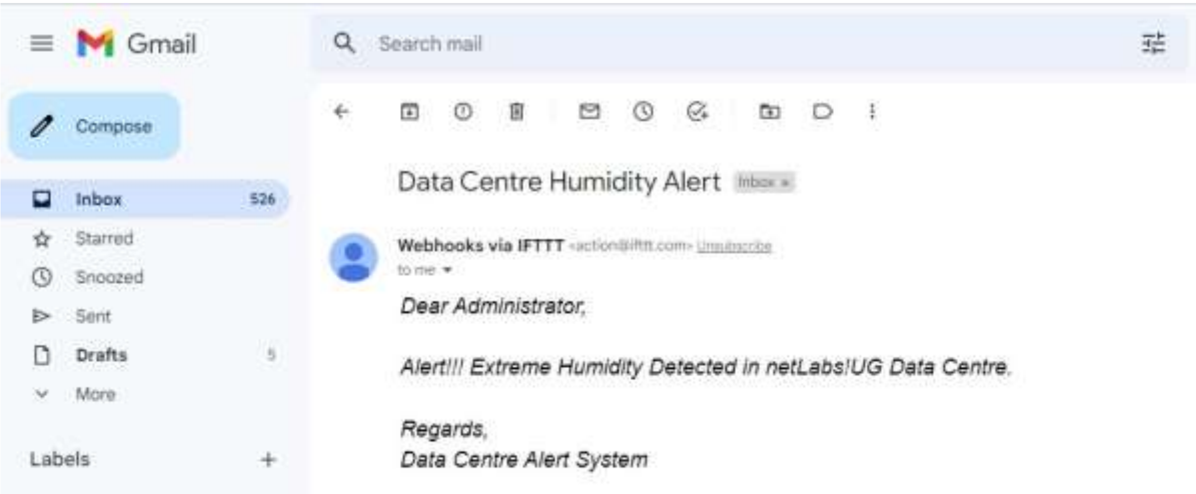


Figure 53: Humidity alert email notification

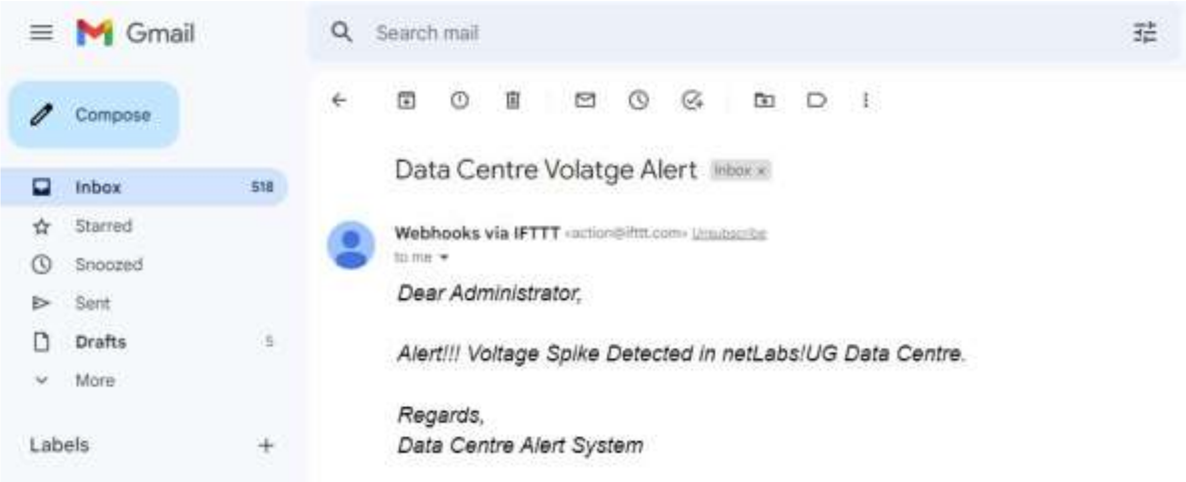


Figure 54: Voltage alert email notification

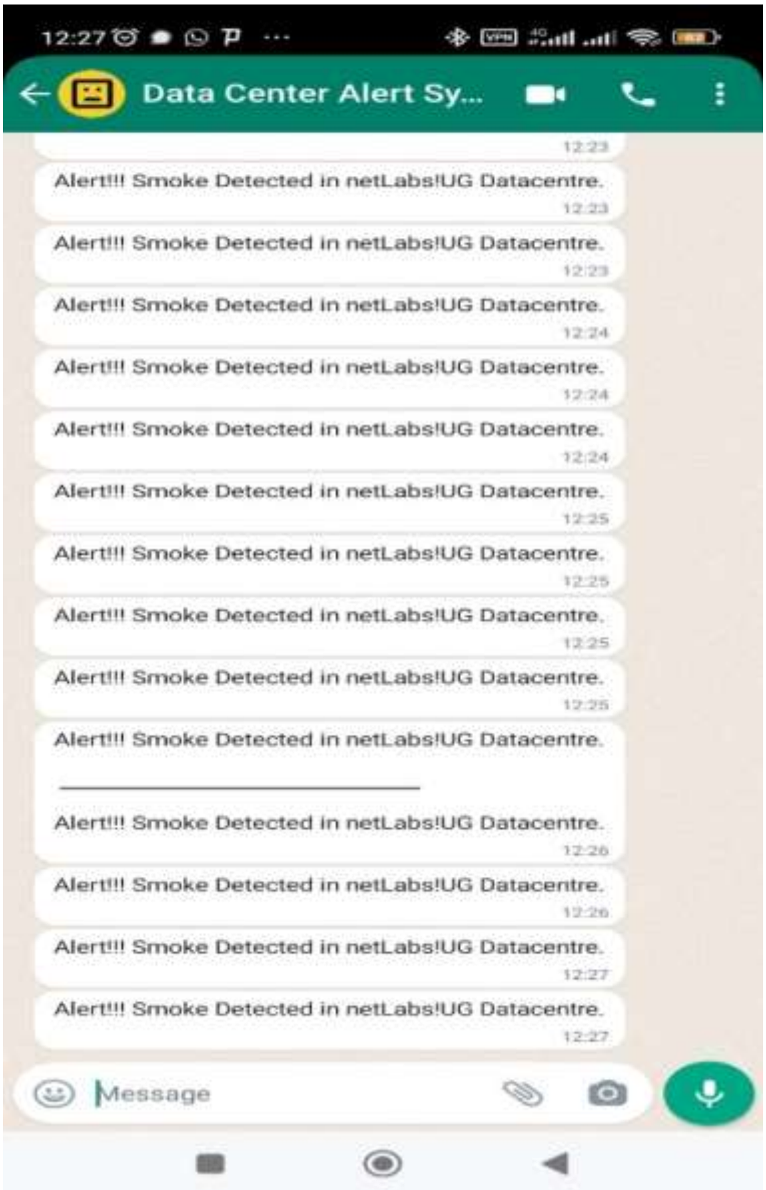


Figure 55: Smoke alert WhatsApp notification

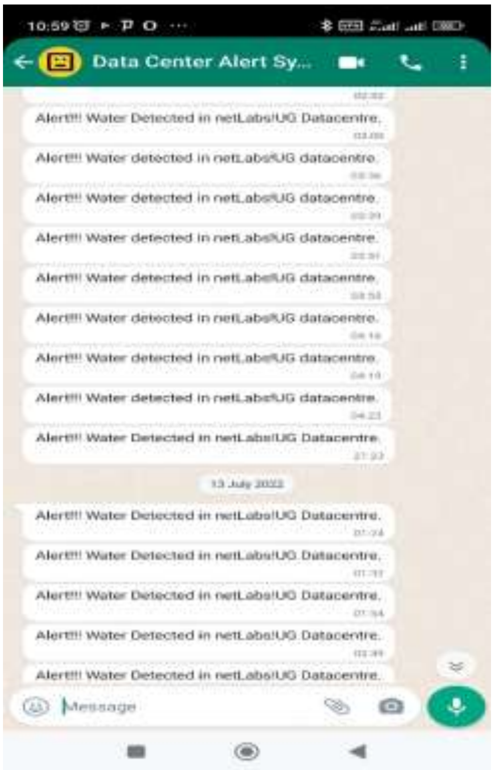


Figure 56: Water alert WhatsApp notification

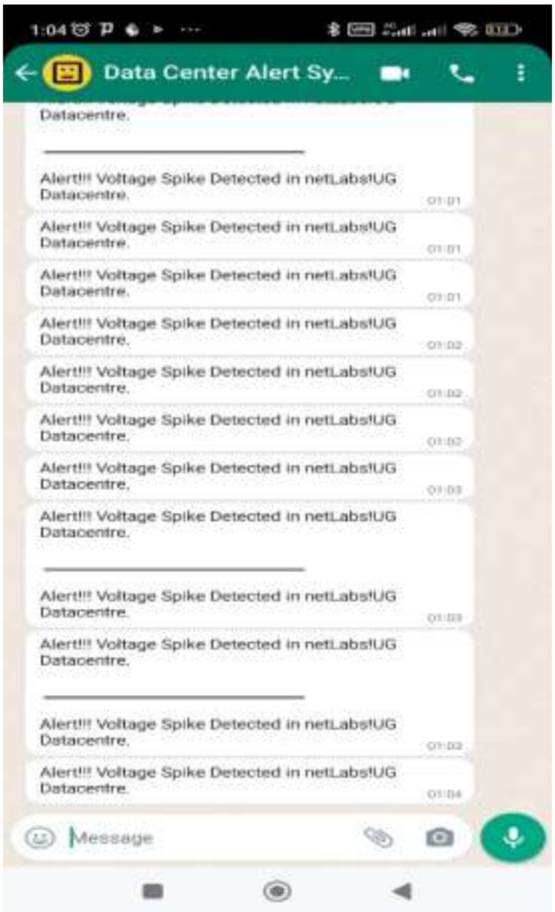


Figure 57: Voltage alert WhatsApp notification

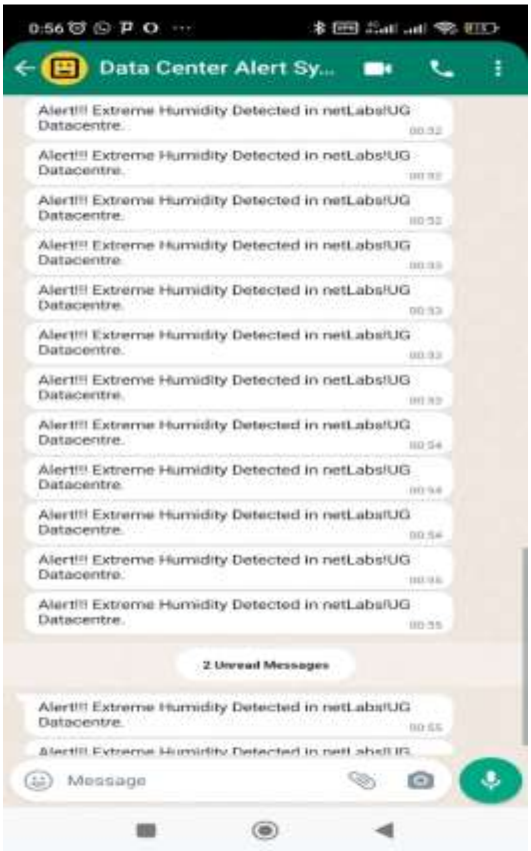


Figure 58: Humidity alert WhatsApp notification

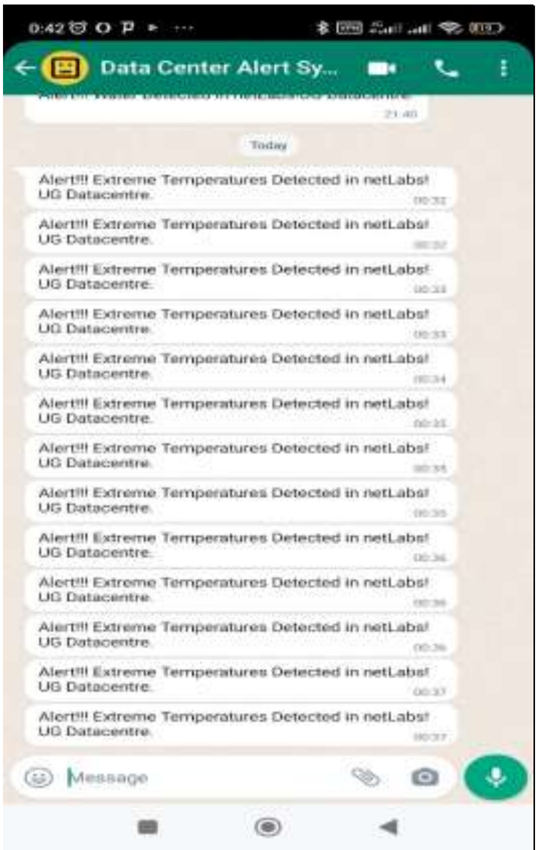


Figure 59: Temperature alert WhatsApp notification

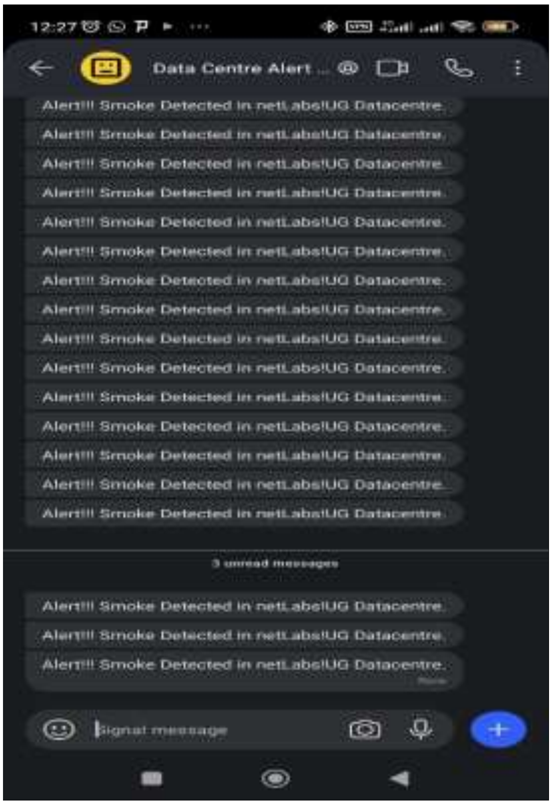


Figure 60: Smoke alert SMS notification

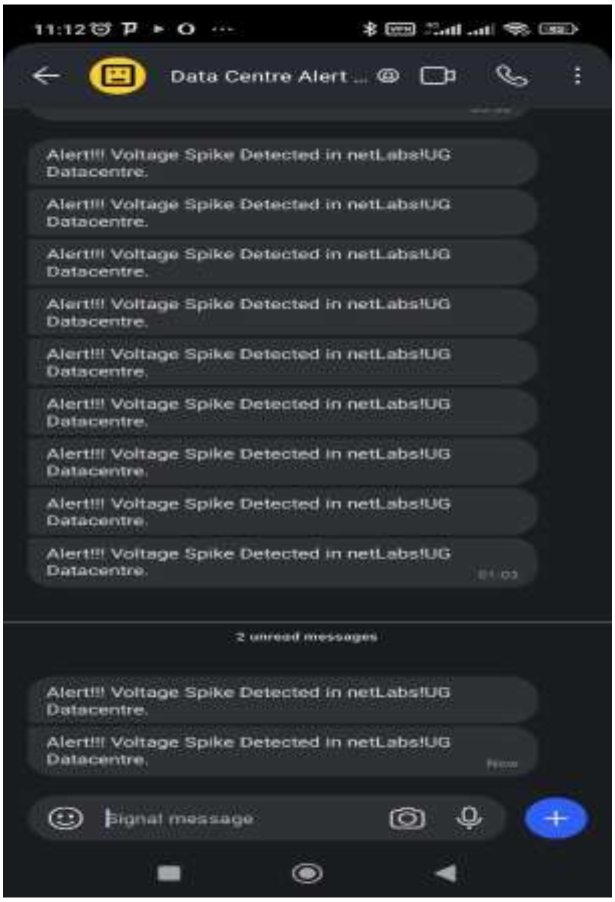


Figure 61: Voltage alert SMS notification

Figure 62: Water alert SMS notification

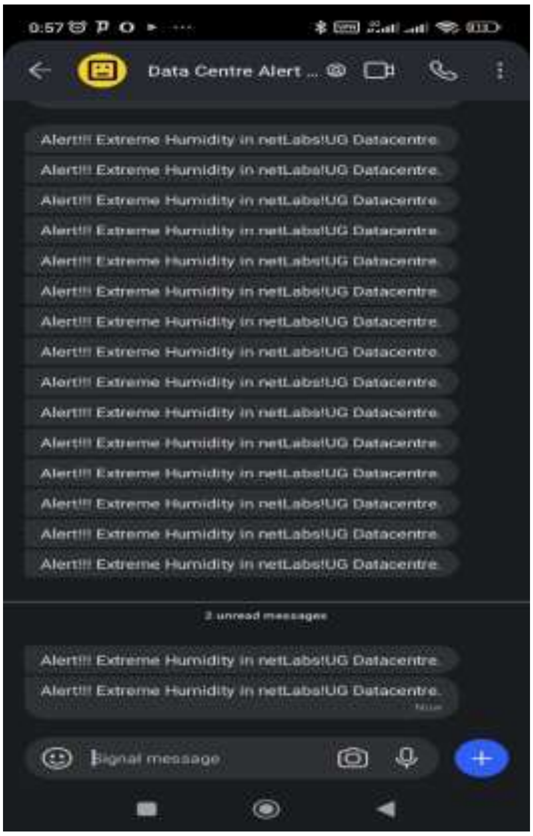


Figure 63: Humidity alert SMS notification

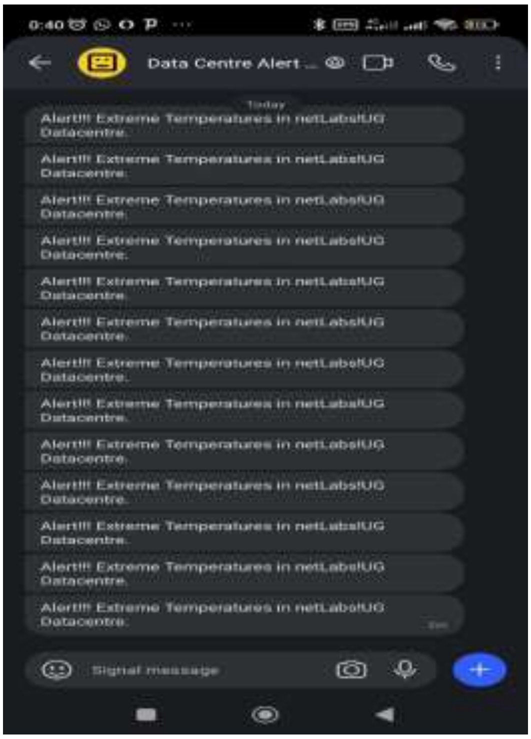


Figure 64: Temperature alert SMS notification

4.3.5 Monitoring and Analysis

Sensor readings are sent to the ThingSpeak platform using the ThingSpeak API for cloud data logging and are updated every five (5) minutes as illustrated in Figs. 65, 66 and 67.

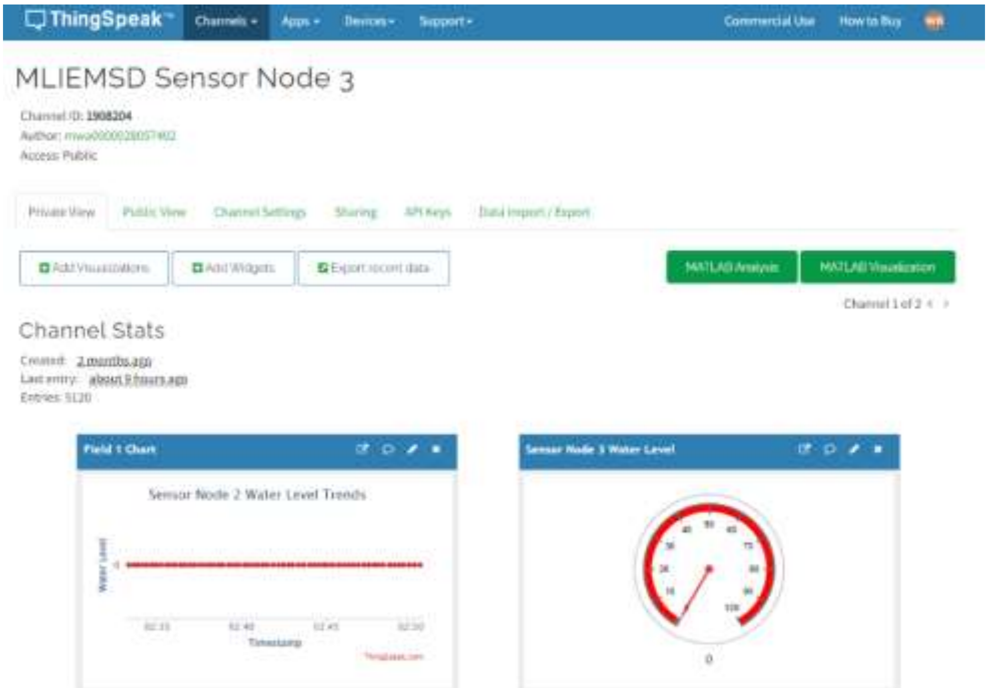


Figure 65: Sensor node 3 ThingSpeak channel

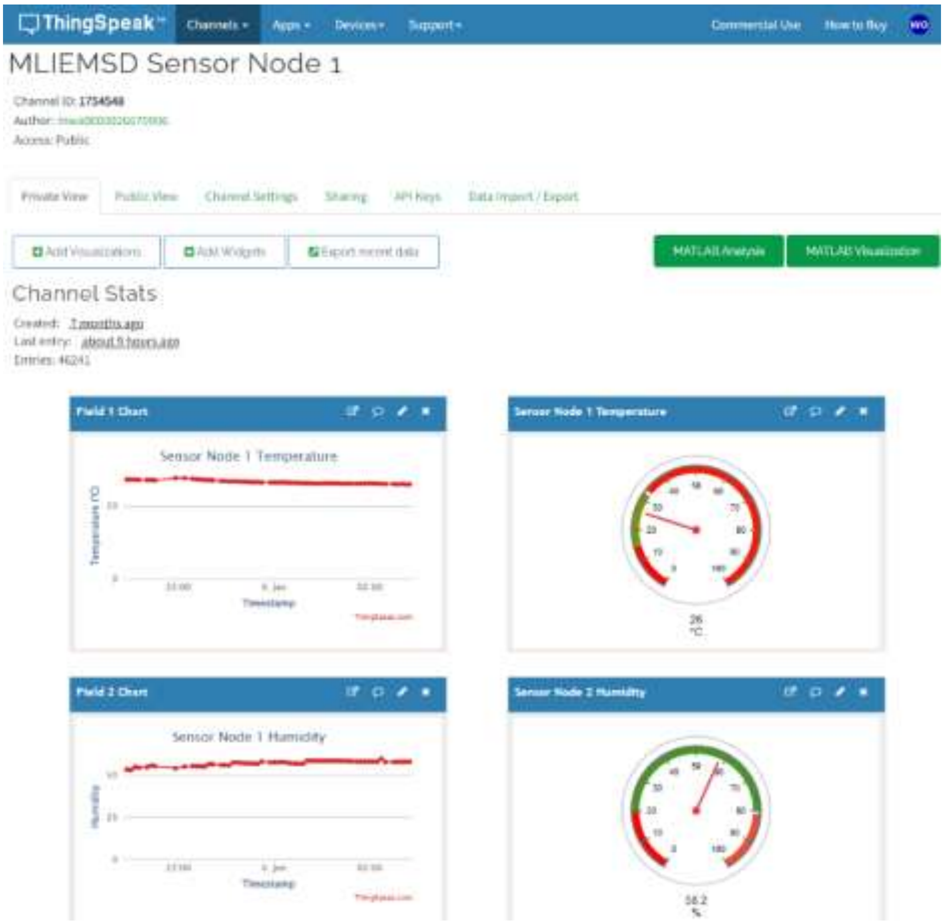


Figure 66: Sensor node 1 ThingSpeak channel

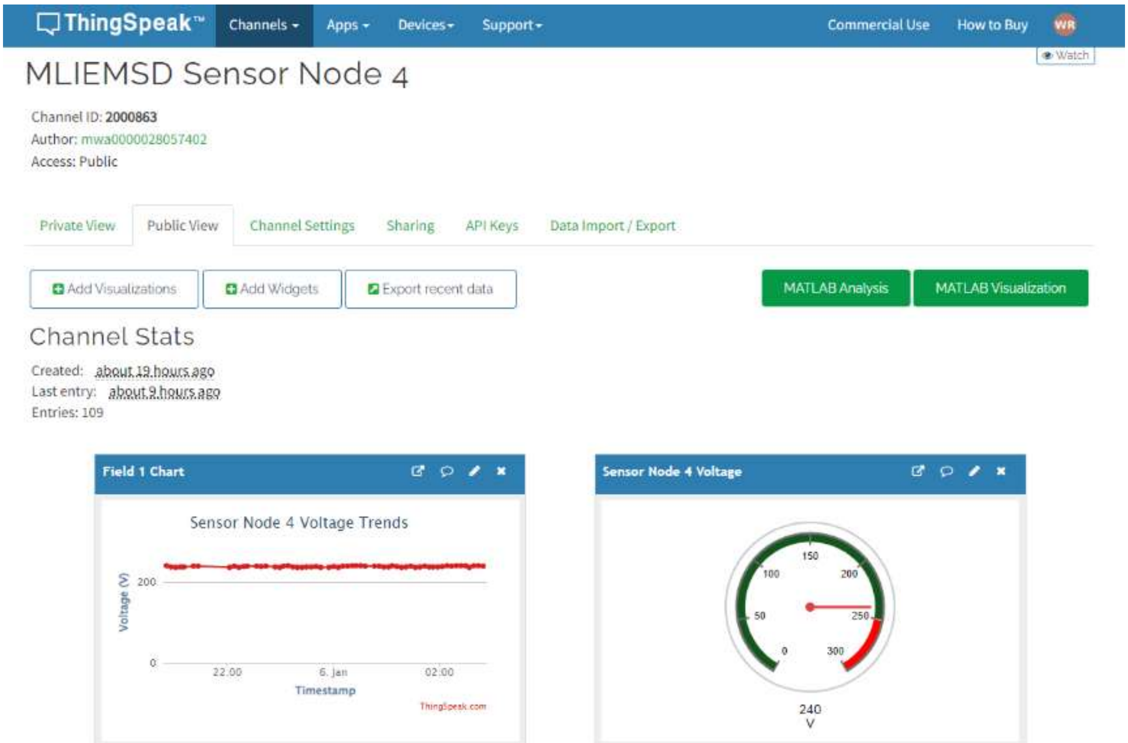


Figure 67: Sensor node 4 ThingSpeak channel

A customized web dashboard was developed using the dash framework, and it utilizes the ThingSpeak platform iframes to display real-time trends of the environmental parameters measured by the different sensors as well as forecasts from the time series forecasting machine learning models. The dashboard consists of four (4) pages i.e., sensor node 1, sensor node 2, sensor node 3, sensor node 4 pages displaying trends of the various measured parameters and forecasts page displaying the predicted temperature and humidity patterns as illustrated from Fig. 68 to Fig. 71.

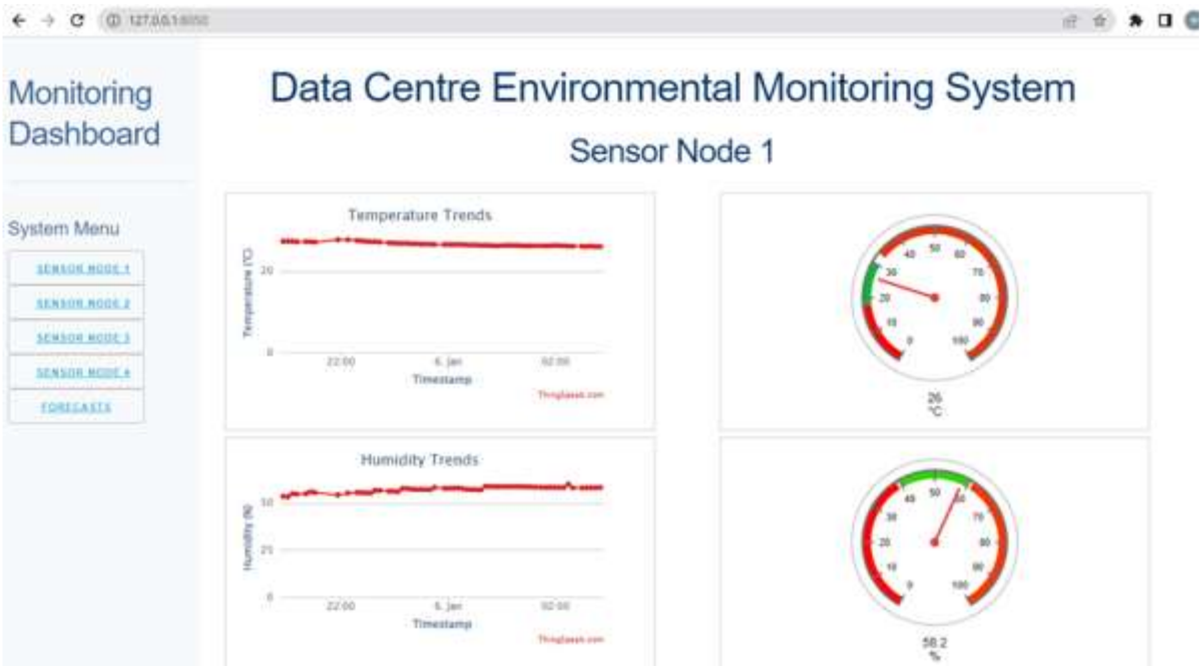


Figure 68: Sensor node 1 page displaying temperature and humidity trends

Figure 69: Sensor node 3 page displaying water level trends

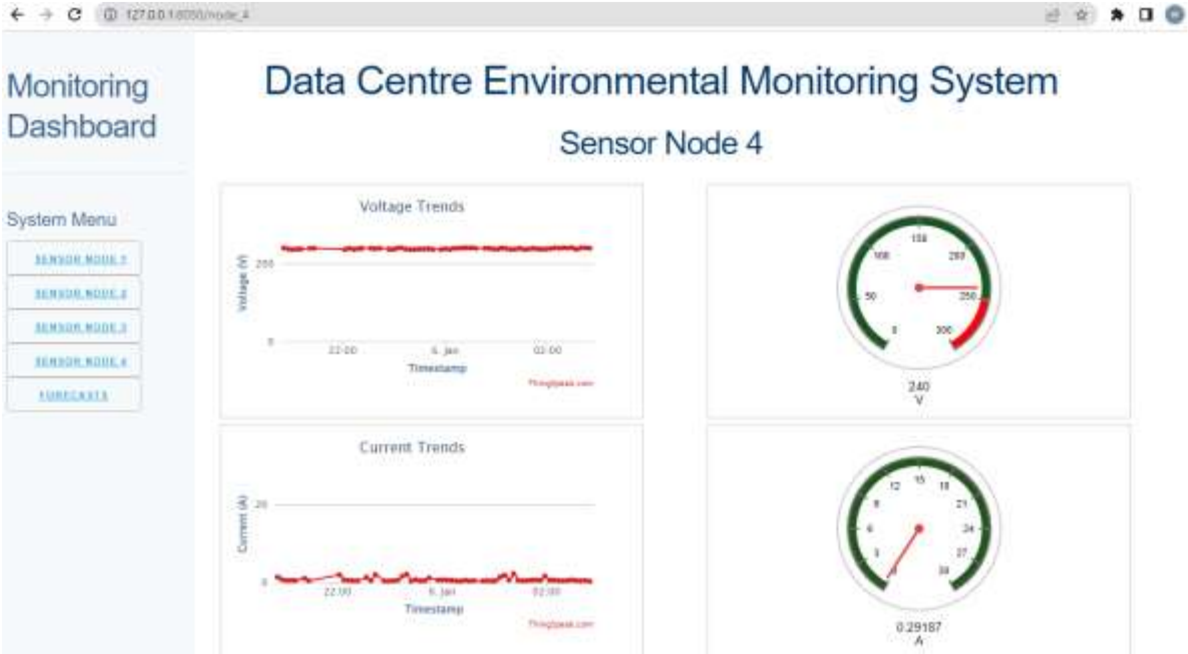


Figure 70: Sensor node 4 page displaying voltage, current and power trends

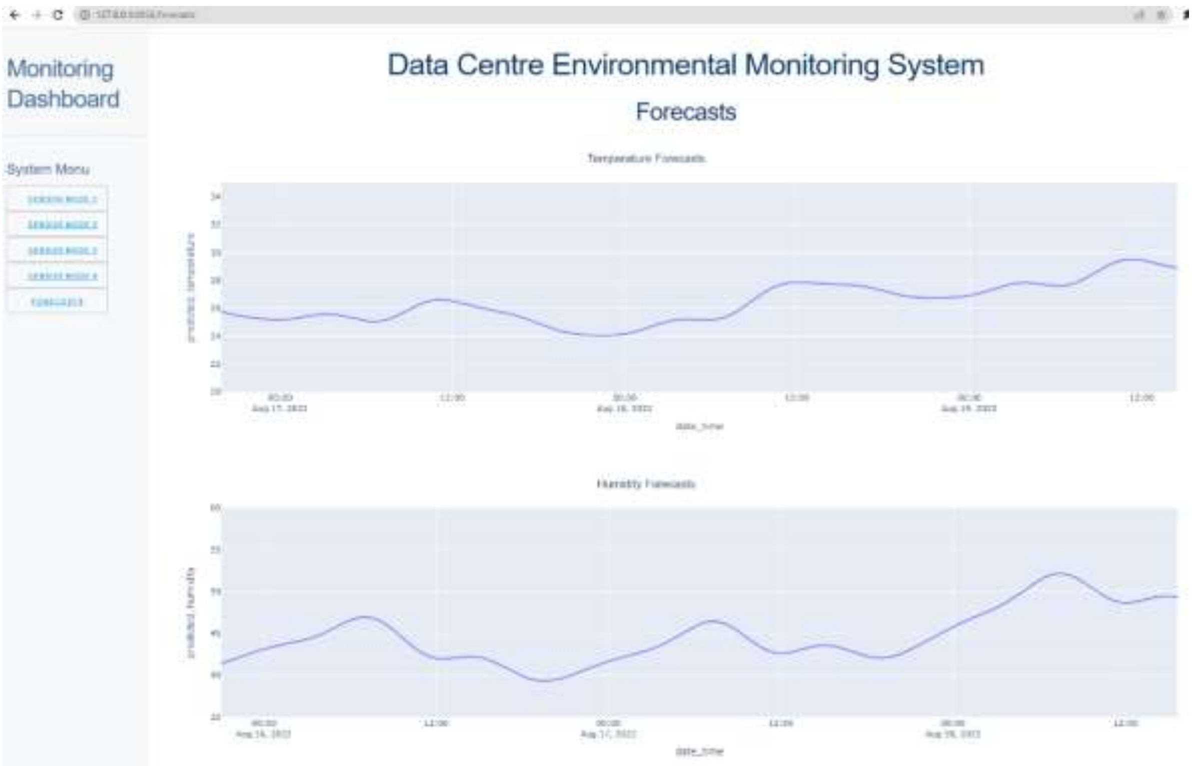


Figure 71: Forecasts page displaying predicted temperature and humidity trends

4.4 Development of Time Series Forecasting Models for Temperature and Humidity

4.4.1 Overview

The dataset was collected from sensor node 1 deployed in the netLabs!UG data centre, Kampala, Uganda where sensor readings were logged every five (5) minutes for a period of 44 days (from 11 July 2022 to 25 August 2022).

Table 5: Dataset description

S/N	Feature	No of observations
1.	id	8010
2.	date_time	8010
3.	temperature	7998
4.	humidity	7998

This phase involved data preprocessing, exploratory data analysis, modeling and evaluation as elaborated in the sections below.

4.4.2 Data Preprocessing

This involved inspection of the panel data captured from the sensor nodes, data cleaning (checking and handling of missing values), data type validation, time series validation as well as data range validation. The aim was to get a better understanding of the data and to prepare it for training the machine learning models.

The dataset was checked for missing values and it was discovered that there were twelve (12) missing values found. The missing values were due to sensor malfunctions during the data collection. Linear interpolation was utilized to handle the missing values due to its ability to maintain temporal continuity.

The data types of the panel data were explored and it was observed that 1 feature was of integer data type, 1 feature was of object data type while 2 features were of float data type.

The panel data was also observed to be in the expected time series ranges i.e., between 11 July 2022 and 25 August 2022.

All the observations were found to be in the expected standard range with no unacceptable outliers observed.

4.4.3 Exploratory Data Analysis

This involved disclosing of the different data constraints, identifying of significant features as well as suitable methods to be employed in the modelling phase.

Matplotlib and seaborn libraries were used to create visualizations that provided more insights about the panel data that in turn guided in the building of the machine learning models.

The stationarity of the time series was checked using the Augmented Dickey–Fuller (ADF) test as well as Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test where both tests confirmed that the series were stationary. The results of the tests are illustrated in Figs. 72 and 73 as well as Table 6.

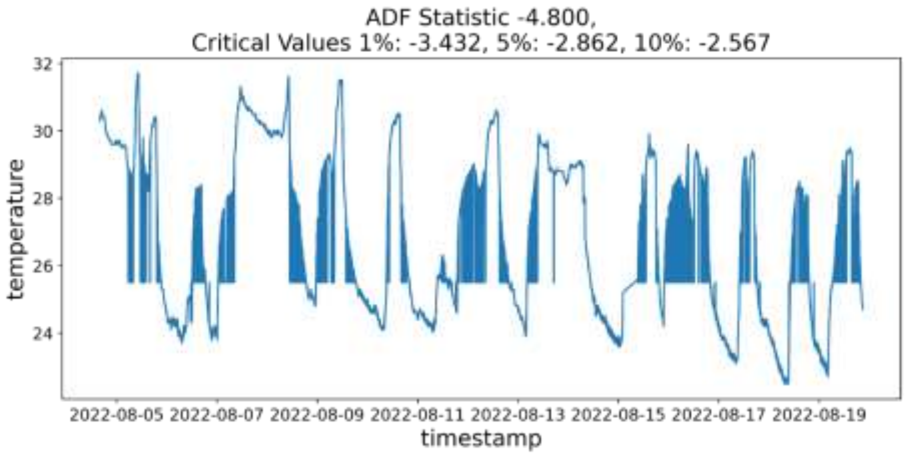


Figure 72: Temperature distribution with ADF statistic and critical values

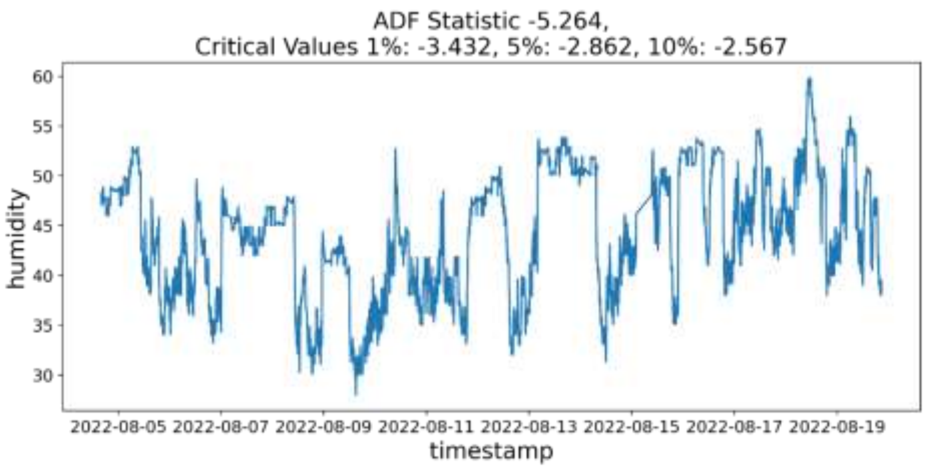


Figure 73: Humidity distribution with ADF statistic and critical values

Table 6: p-values from the ADF and KPSS stationarity tests

Parameter	ADF p-value	KPSS p-value
Temperature	5.44e-05	0.055
Humidity	7.50e-08	0.1

Different time series features were created and the temperature and humidity trends of each were explored as illustrated in Figures 74 - 77;

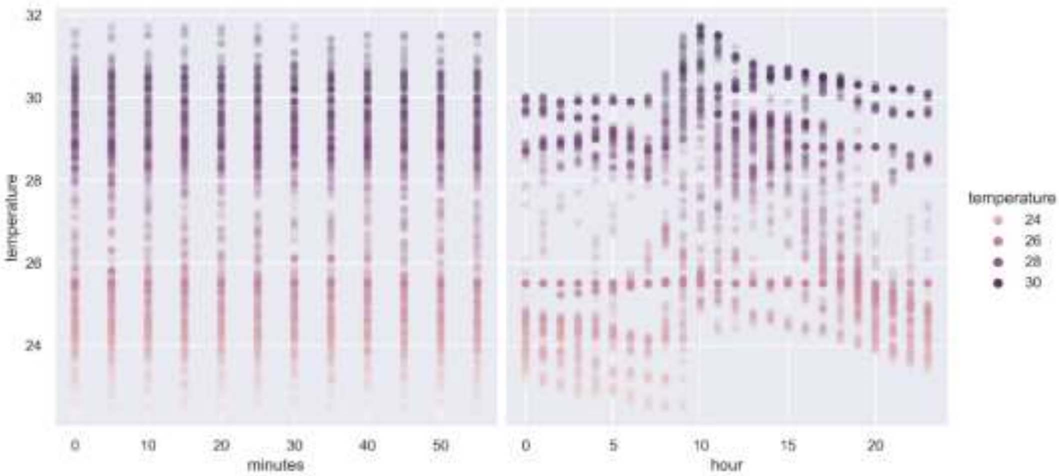


Figure 74: Temperature distribution by minutes and hour

The distribution of temperature by minutes exhibits a consistent pattern, with slight variations in the peaks starting from the 35th minute. In contrast, when examining the temperature distribution by hours, similar peak values are observed for the first eight hours. However, starting from the 9th hour and continuing until the 11th hour, a gradual increase in the peak values is noticed. After the 11th hour, the peak values gradually decrease, indicating a reduction in temperature towards the end of the observed time range.

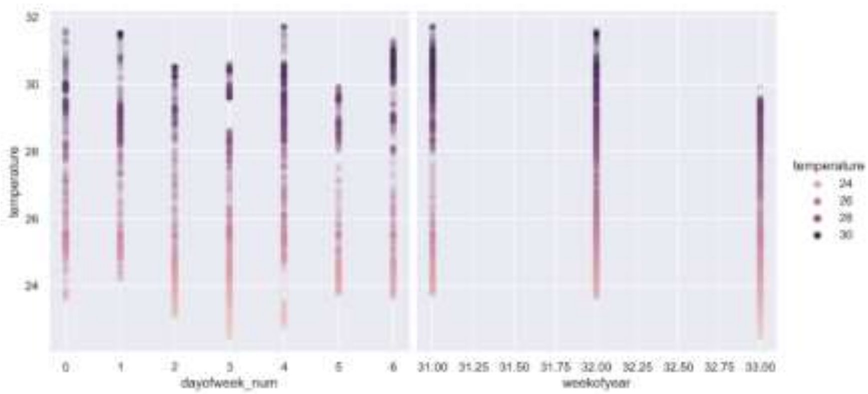


Figure 75: Temperature distribution by day and week

The temperature distribution by day of the week reveals different peaks for each day, suggesting unique temperature patterns throughout the seven-day cycle. Conversely, the temperature distribution by week of the year indicates similar peak values for the first two weeks, followed by a decrease in peak values during the last week.

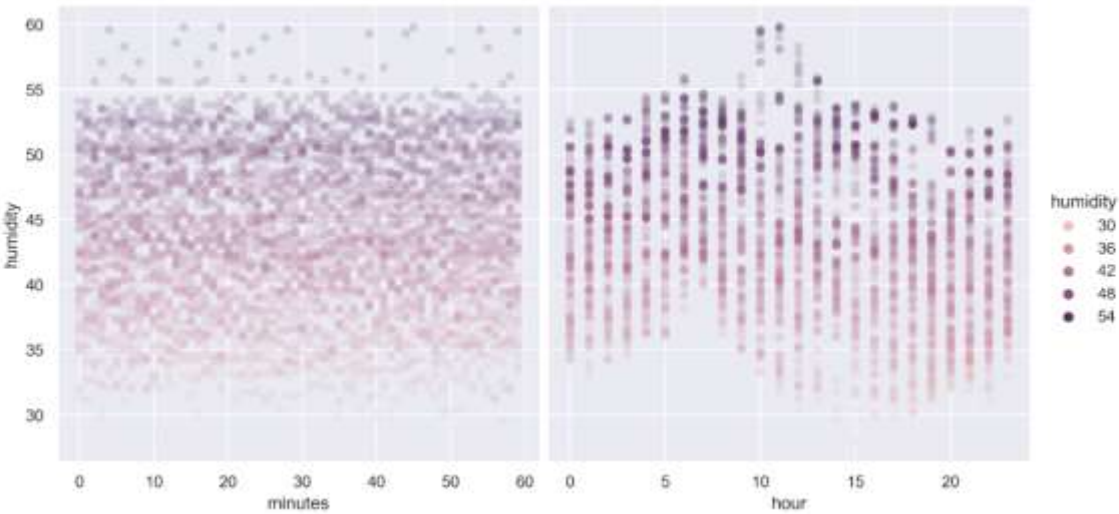


Figure 76: Humidity distribution by minutes and hour

The humidity distribution by minutes demonstrates a fairly consistent pattern throughout the 60 minutes, with little variation observed. On the other hand, the humidity distribution by hour shows similarities in peak values for the first 4 hours, followed by slight increases and decreases in peaks during specific hour ranges. The 10th to 12th hour exhibits a gradual increase in peak values, while the 13th to 16th hour shows a gradual decrease. Subsequently, the peak values become fairly similar for the remaining hours.

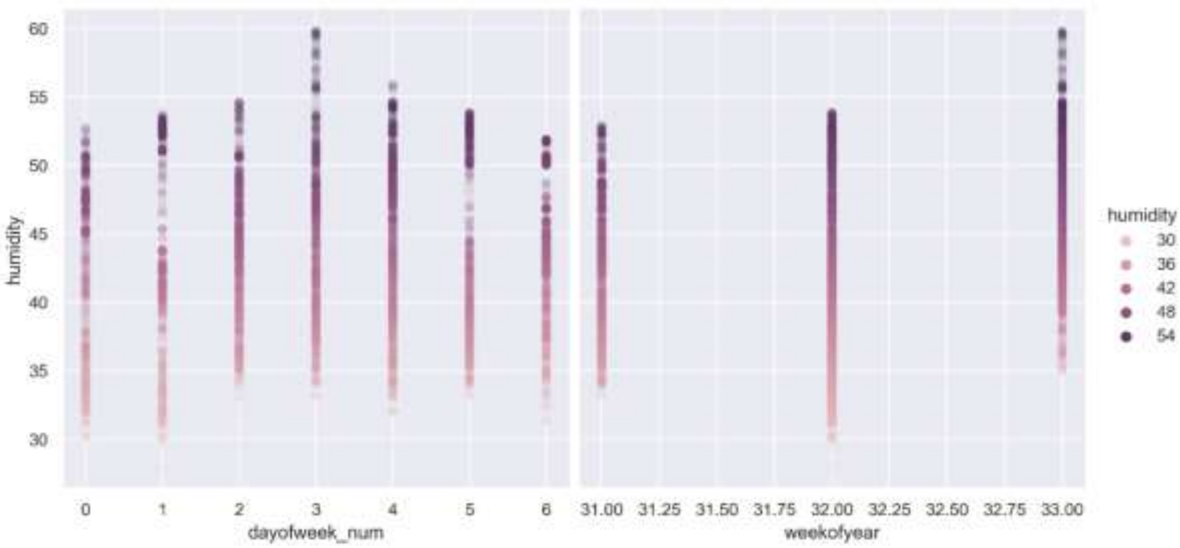


Figure 77: Humidity distribution by day and week

Both the humidity distribution by day of the week and week of the year highlight variations in peak values, indicating different humidity patterns throughout the respective time periods.

4.4.4 Modelling

The dataset was split into a train dataset and test dataset in a ratio of 0.8:0.2 respectively. Temperature and humidity were the target features. ARIMA, Facebook Prophet and ES algorithms were employed in training the models using the train dataset.

(i) Facebook Prophet

Facebook prophet is an open-source additive regression time series forecasting model developed by Facebook that is available in both R and python programming languages. It manages outliers effectively in addition to being robust to trend shifts and missing data. As a result, it facilitates easy and fast training of time series models as it requires less feature engineering.

Facebook Prophet is based on the concept of breaking down a time series into trend, seasonality and holidays components, and then combining them to forecast future values in the series. The components are highlighted as follows:

(a) **Trend:** This component models non-periodic changes using a piecewise logistic or linear growth curve.

(b) **Seasonality:** This component models periodic changes using Fourier series.

(c) **Holidays:** This component models holidays and events using dummy variables.

The value of the time series at time t denoted as y(t) is expressed as:

$$\boldsymbol{y(t) = g(t) + s(t) + h(t) + \varepsilon(t)}$$

where:

- g(t) is the trend component,
- s(t) is the seasonality component,
- h(t) is the holiday component,
- ε(t) is the error term.

The hyper parameters used in the training of the prophet models are illustrated in Table 7.

Table 7: Hyper parameters used for the Facebook Prophet models

Hyper parameter	Value
Growth	linear
changepoint_prior_scale	0.01
seasonality_mode	additive
daily_seasonality	true
weekly_seasonality	true
yearly_seasonality	false

The growth of the prophet models was set to linear as no saturation was observed in the panel data while the seasonality mode was set to additive since a constant trend change was desired.

The trained models exhibit the trend, daily and weekly seasonality as illustrated in Figs. 78 and 79.

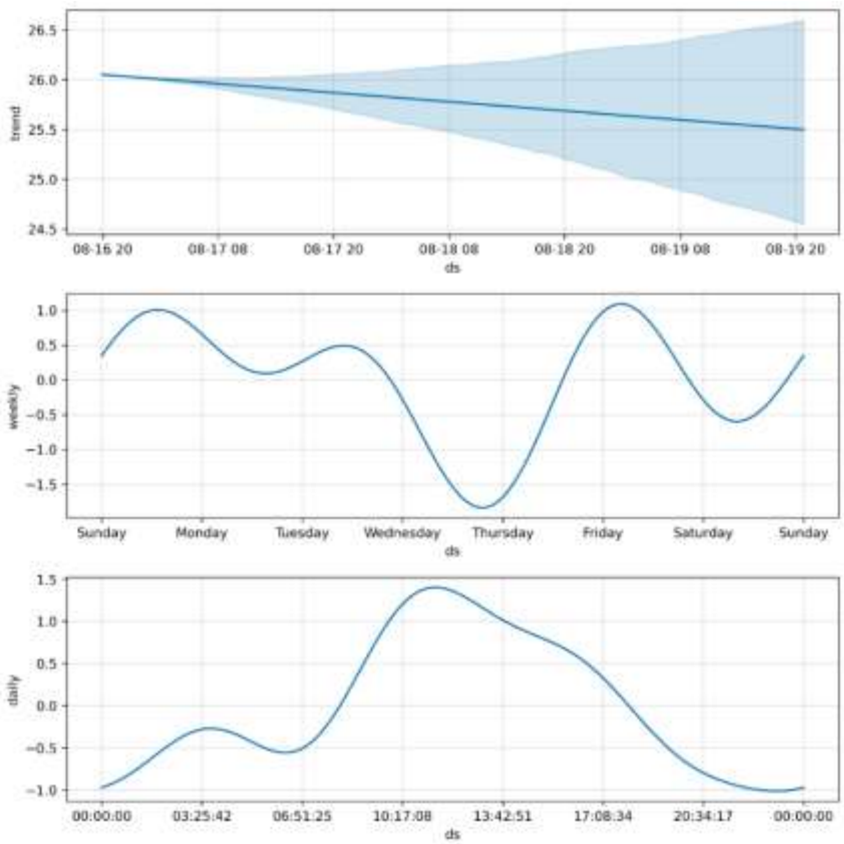


Figure 78: Temperature model trend, daily and weekly seasonality

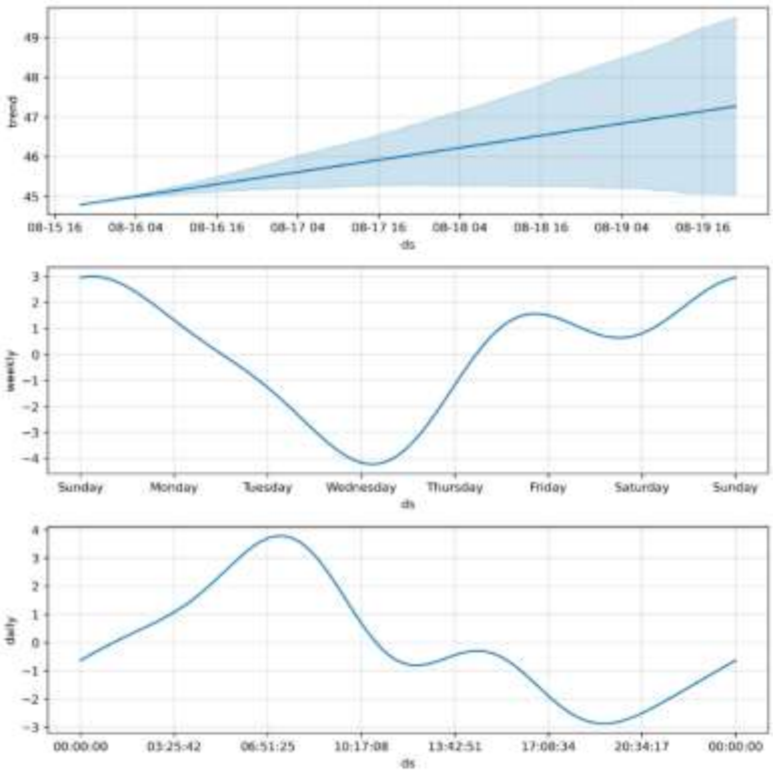


Figure 79: Humidity model trend, daily and weekly seasonality

(ii) Autoregressive Integrated Moving Average (ARIMA)

Autoregressive Integrated Moving Average (ARIMA) is a class of statistical analysis models that are fitted to time series data to better comprehend the data or to forecast future data points in the series. It is one of the most widely used time series forecasting approaches and only requires prior data to generalize forecasts. Furthermore, its model accuracy can be improved while maintaining a minimum number of parameters.

The ARIMA model is predicated on the hypothesis that time series data can decomposed into three components as follows:

(a) **Autoregressive (AR):** This component models the dependence of the present observation of the time series data on its previous observations.

(b) **Integrated (I):** This component models the difference between the present observation and the previous observation of the panel data so as to achieve stationarity in a time series.

(c) **Moving Average (MA):** This component models the time series' residual error as a function of the previous errors.

The ARIMA model takes the general form of:

ARIMA (p.d,q)

where:

- p represents the order of the AR component,
- d represents the order of the I component,
- q represents the order of the MA component.

The value of the time series at time t denoted as y(t) is expressed as:

$$\boldsymbol{y(t) = c + \Sigma\_\{i = 1\}\hat{}\{p\}\varphi\_iy\_\{t - i\} + \Sigma\_\{j = 1\}\hat{}\{q\}\theta\_je\_\{t - j\} + e\_t}$$

where:

- c is a constant
- φ_i are the AR terms
- θ_j are the MA terms
- e_t is the error term at time t

The optimal order of the temperature and humidity ARIMA models was automatically obtained using the pmdarima python library and their respective hyper parameter order values are illustrated in Table 8.

Table 8: Hyper parameters used for the ARIMA models

	p	d	Q
Temperature	5	0	4
Humidity	3	0	2

The behavior of the fitted ARIMA models was examined using a plot of the residuals over time, Q-Q plot to determine whether the residuals are normally distributed, a histogram of the residuals and correlogram to indicate the autocorrelation of the residual errors at different lags and these are illustrated in Figs. 80 and 81.

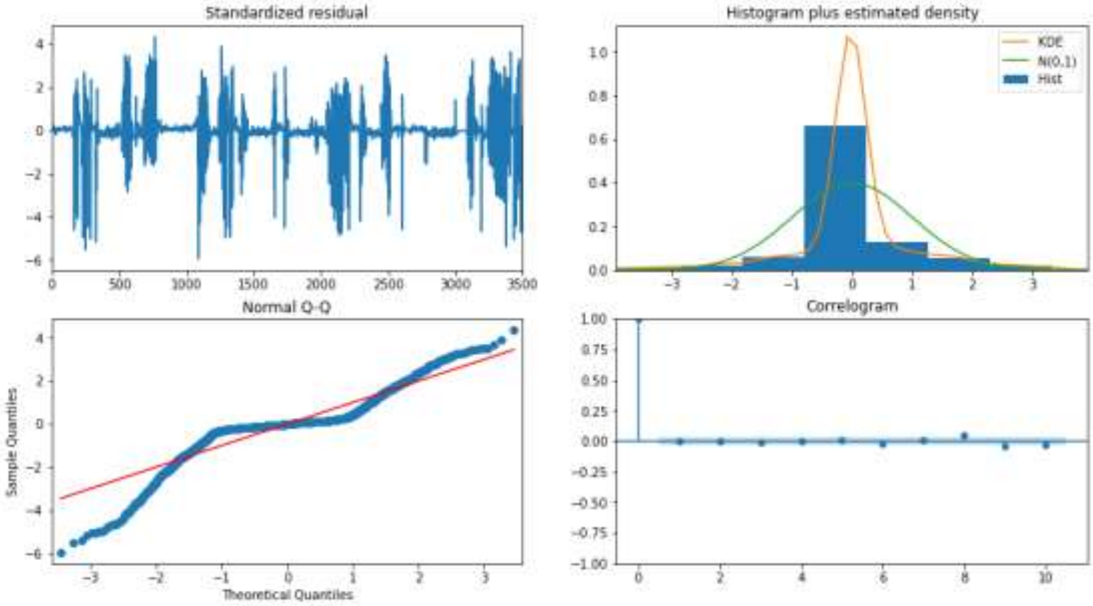


Figure 80: Temperature model plot diagnostics

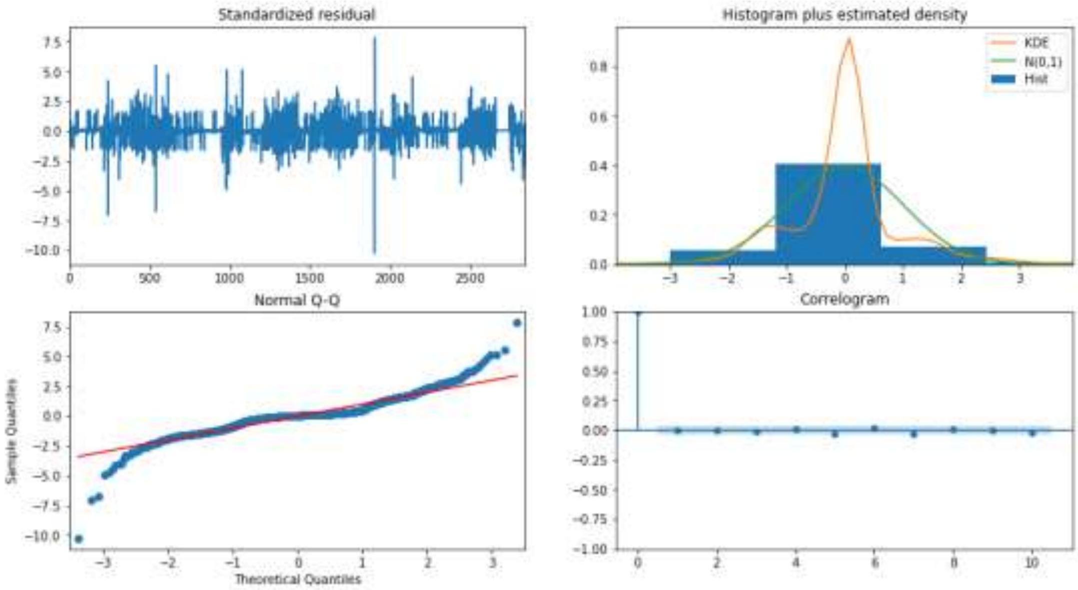


Figure 81: Humidity model plot diagnostics

(iii) Exponential Smoothing (ES)

Exponential Smoothing (ES) is a popular time series forecasting method employed for univariate data. It allocates exponentially decreasing weights to past observations unlike in single moving averages-based methods like ARIMA where the past observations are weighted equally. The most recent observations are assigned relatively more weight in forecasting as compared to the older observations. Exponential Smoothing is therefore usually used for short term forecasting as its working principle doesn't favor long term forecasting.

The value of the time series at time t denoted as y(t) is expressed as:

$$\boldsymbol{y(t) = \alpha * x(t) + (1 - \alpha) * y(t-1)}$$

where:

- x(t) is the actual value at time period t
- α is the smoothing factor (a value between 0 and 1)
- y(t-1) is the forecast for the previous time period (t-1)

The hyper parameters used in the training of the ES models are illustrated in Table 9.

Table 9: Hyper parameters used for the ES models

Hyper parameter	Value
smoothing_level	0.85
smoothing_seasonal	0.5
smoothing_slope	0.6

4.4.5 Evaluation

The trained models were tested on the test dataset and the performance for each algorithm was obtained in terms of MAPE, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as summarized in Tables 10 and 11 as well as in Figs. 82 and 83.

Table 10: Comparison of the performance of the employed algorithms for the temperature models

Metric	Facebook Prophet	ARIMA	ES
MAPE	5.771	6.459	6.495
MAE	1.480	1.667	1.672
RMSE	1.819	2.041	2.037

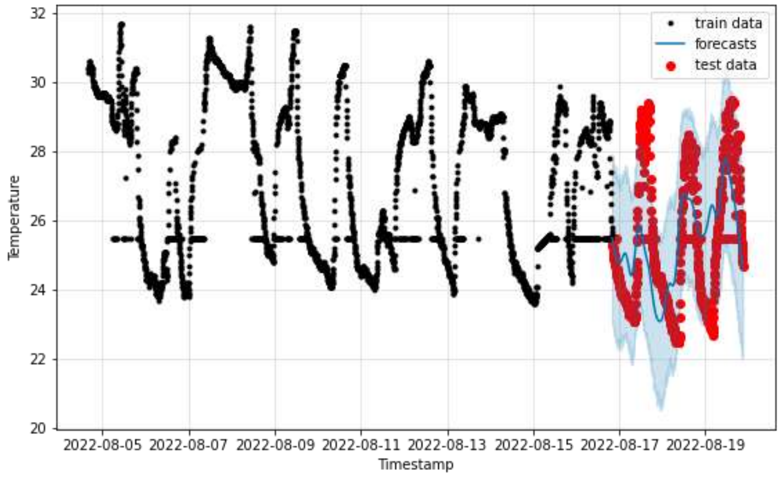


Figure 82: Correlation of train data, forecasts and test data for the temperature model

Table 11: Comparison of the performance of the employed algorithms for the humidity models

Metric	Facebook Prophet	ARIMA	ES
MAPE	8.981	10.294	10.160
MAE	4.199	4.635	4.594
RMSE	5.256	5.272	5.218

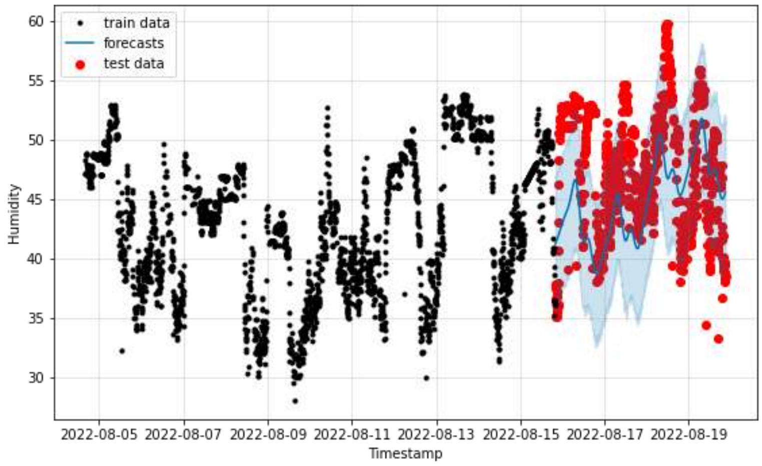


Figure 83: Correlation of train data, forecasts and test data for the humidity model

Based on the evaluation of Table 10 and Table 11, it is evident that Facebook Prophet outperformed ARIMA and ES models consistently for both the temperature and humidity forecasting. Facebook Prophet's superior performance across all evaluation metrics suggests that it provides more precise forecasts and captures the underlying patterns and trends in the data more effectively than ARIMA and ES models.

The Facebook Prophet models for temperature and humidity were deployed separately on the sink node. This approach fosters focused analysis and enhanced result interpretability. Furthermore, it allows for individual optimization, ensuring the unique characteristics of temperature and humidity data contribute to accurate forecasting. Additionally, this approach enables adaptability to changing conditions thereby facilitating appropriate fine-tuning of each model based on evolving environmental patterns and factors.

4.5 Validation of the Developed IoT Environmental Monitoring System

4.5.1 Overview

The developed IoT monitoring system was subjected to a user acceptance test to determine if it met the needs of the end users. Five (5) end users from netLabs!UG were allowed to interact with the system and each of them was requested to complete a user experience survey as illustrated in Appendix 2.

4.5.2 Survey Findings

Figures 84–90 depict some of the questions in the user experience survey and their corresponding analyzed results from the 5 participants.

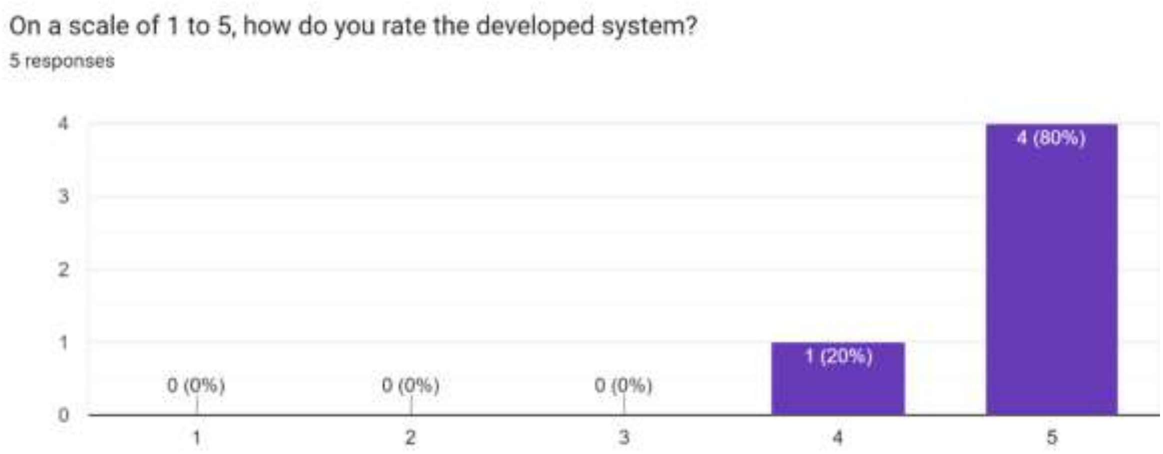


Figure 84: Response to how the users rated the performance of the system

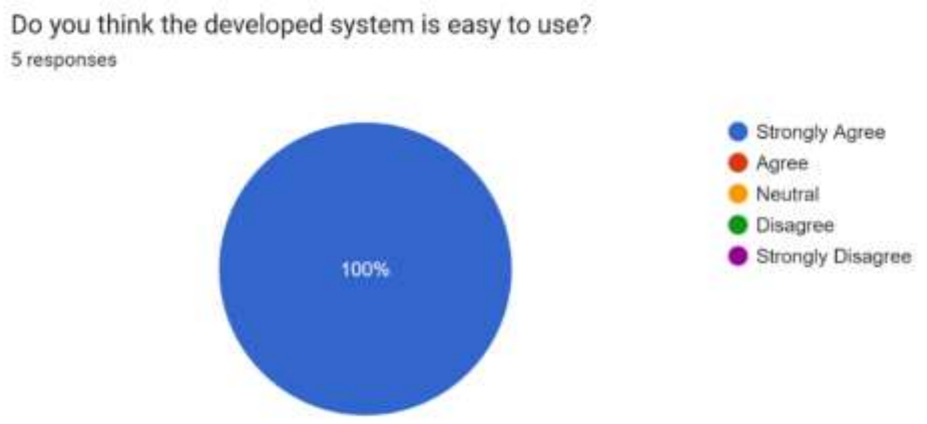


Figure 85: Response to the usability of the system

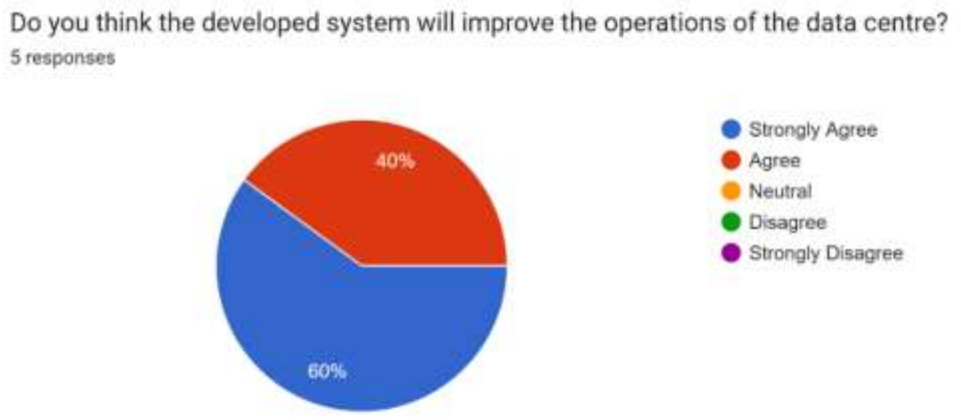


Figure 86: Response to whether the users thought the system would improve the data centre operations

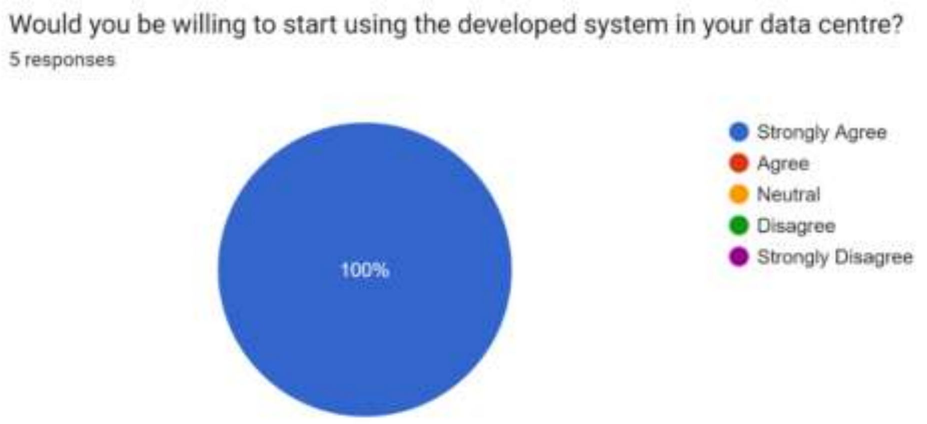


Figure 87: Response to whether the users were willing to start using the system

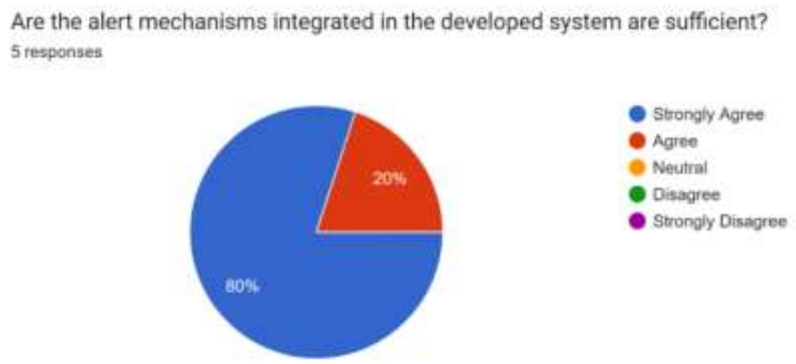


Figure 88: Response to whether the integrated alert mechanisms were sufficient

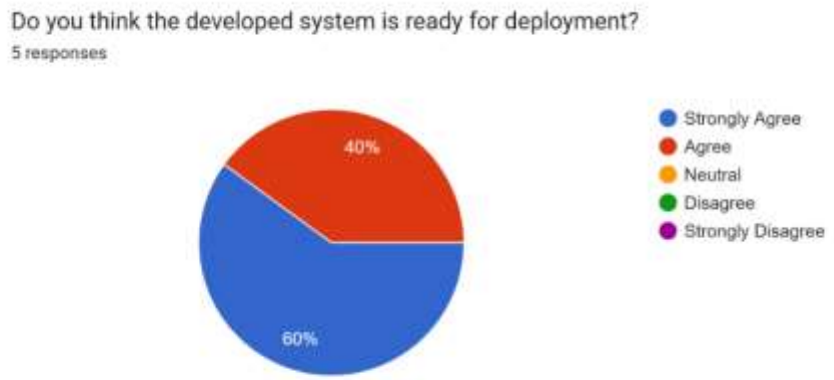


Figure 89: Response to whether the users the system was ready for deployment

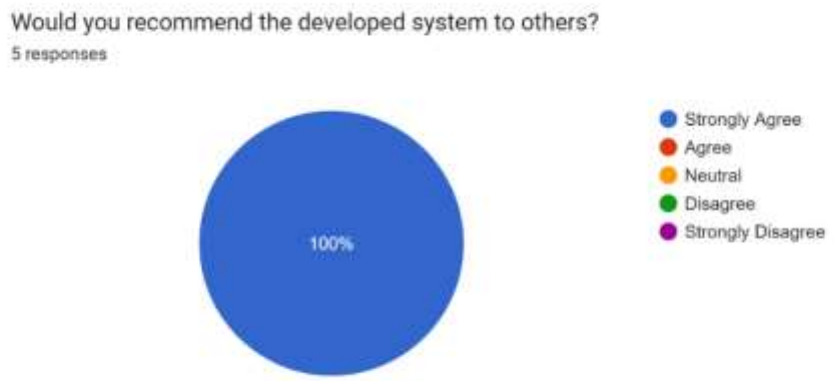


Figure 90: Response to whether the users would recommend the system to others

4.6 Discussion

The requirements gathering phase revealed a unanimous consensus among all participants regarding the importance of implementing an environmental monitoring system for their data centers. Surprisingly, 57.9% of the institutions represented in the study did not have such systems in place, indicating a significant gap between the recognized need and the actual implementation. Furthermore, a striking 84.2% of the participants acknowledged that they had experienced work disruptions due to unfavorable environmental conditions. This highlights the critical impact of environmental factors on data center operations and underscores the urgency of addressing this issue. Among the institutions that did have environmental monitoring systems in place, several limitations were identified. These included monitoring only a single parameter, relying on a single means of alert mechanism, and lacking prediction capabilities. It is noteworthy that all participants recognized that multiple environmental parameters affect data centers, and they also acknowledged the potential benefits of predicting significant environmental conditions to enhance data center operations.

The developed system comprises several key components i.e., a sensing unit, processing and storage unit, alerting unit, monitoring and analysis unit, and a time series forecasting unit. The sensing unit is responsible for measuring various environmental parameters i.e., temperature, humidity, smoke, water, voltage, and current. These readings serve as the foundation for monitoring and analyzing the data center's conditions. The processing and storage unit stores and processes the collected readings from the sensing unit. This unit ensures that data is efficiently managed and readily accessible for further analysis and decision-making. The monitoring and analysis unit offers real-time visualizations of the sensed parameters, allowing users to gain insights into the current environmental conditions of the data center. The alerting unit is responsible for providing timely alerts in the event of any undesirable environmental conditions. It employs various alert mechanisms such as audio alarms, WhatsApp messages, email notifications, and SMS alerts. This ensures that relevant stakeholders are promptly notified when predefined environmental parameter thresholds are exceeded. The time series forecasting unit utilizes advanced machine learning models to predict future temperature and humidity patterns. By leveraging historical data and machine learning algorithms, the system can anticipate trends and provide valuable insights for proactive measures and decision-making. Overall, the developed system ensures that organizations are continuously aware of the environmental conditions within their data centers. By promptly notifying stakeholders of any deviations from desired parameters, organizations can take proactive measures to mitigate risks, minimize downtime, increase reliability, prevent equipment damage, and ultimately reduce long-term operating costs.

The developed system stands out from existing works due to several key advantages. Firstly, it possesses forecasting capabilities, which are lacking in other data centre monitoring systems. This forecasting ability enables firms to plan ahead, optimize resources, and implement appropriate predictive measures. Furthermore, unlike many related works that focus primarily on temperature and humidity, the developed system monitors a wide range of environmental parameters. This comprehensive monitoring provides companies with a holistic understanding of the data center environment, enabling more informed decision-making and analysis. In terms of alert mechanisms, the developed system outperforms the majority of existing works. While many projects rely on a single alert mechanism, the developed system incorporates four (4) different alert mechanisms. This multi-channel approach offers numerous benefits, including flexibility and redundancy. It ensures that notifications are delivered reliably, covers a broader range of users' preferences, and enhances overall awareness to critical events. Additionally, the developed system exhibits superior scalability compared to existing projects. It allows for the seamless addition or relocation of sensor nodes without disrupting the existing infrastructure. This scalability feature offers firms the flexibility to adapt to changing requirements and expand their data centre monitoring capabilities without significant costs or technical complications.

The system validation process yielded positive feedback from the respondents, indicating a high level of satisfaction with the developed system. An overwhelming 80% of the participants rated the overall performance of the system as 5 out of 5, demonstrating their confidence and positive perception of its capabilities. The remaining 20% rated it as 4 out of 5, indicating a still favorable assessment of the system. Furthermore, all participants strongly agreed that the system was user-friendly and expressed their willingness to start using it. This feedback reflects the ease of use and intuitive nature of the system, which contributes to its usability and adoption. The participants also recognized the potential for the system to enhance the operations of their data centers. They acknowledged the sufficiency of the incorporated alert mechanisms, indicating that the system effectively provides timely notifications and alerts in response to undesirable environmental conditions. Overall, the participants expressed their readiness for system deployment and conveyed their intention to recommend it to others. This positive endorsement highlights their confidence in the system's functionality, reliability, and ability to address their environmental monitoring needs effectively.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

Data centres are an important aspect of many firms in the modern world facilitating their various business operations. Undesirable environmental conditions such as extreme temperatures, humidity, fire, voltage spikes and surges, and water leaks can lead to a decline in performance, unexpected failures and total damage of equipment in the data centers. An environmental monitoring system for data centres that utilizes IoT and machine learning was developed to improve reliability as a result of decreased unplanned downtime, reduced risk of equipment damage and data loss. It monitors environmental conditions in data centres in real-time and provides alerts of undesirable environmental situations thereby allowing corrective measures to be taken a timely manner. The Facebook prophet algorithm proved to be the best algorithm for forecasting temperature and humidity patterns, returning a MAPE of 5.77% and 8.98%, MAE of 1.48 and 4.2, and RMSE of 1.82 and 5.26, respectively. By providing prediction capabilities, monitoring multiple critical parameters, and offering scalability, the system effectively addresses the technical gap identified in related works. As a result, it significantly advances the state-of-the-art in environmental monitoring for data centers, ensuring enhanced reliability and performance.

5.2 Recommendations

The developed system utilizes IoT technology that has demonstrated great results over time across the environmental monitoring domain. Researchers can explore a Web of Things (WoT) approach and its results compared with those of the developed IoT system. By investigating the potential benefits and drawbacks of WoT in comparison with the established IoT system, researchers can gain valuable insights into the effectiveness and suitability of WoT for their specific applications. Such a comparative analysis would contribute to advancing knowledge and fostering innovation in the field of environmental monitoring.

Researchers should consider broadening the scope of the project beyond environmental monitoring by incorporating conditioning mechanisms. By enabling the system to take immediate corrective actions upon detecting any undesirable environmental conditions, this enhancement will significantly improve the system's capabilities. The integration of conditioning mechanisms will empower the system to proactively respond to environmental anomalies, ensuring a more resilient and automated approach to environmental management in diverse settings.

In the future, a deep learning approach is envisioned to be employed for forecasting of temperature and humidity patterns. The performance of deep learning models would then be compared to that of the time series models incorporated in this system.

The system is currently powered by AC power supply. An alternative DC power supply is envisaged to be integrated to ensure uninterrupted power supply in the events where AC power supply is not available.

REFERENCES

Kaliyamurthie, K. P., Nalini, C., Priya, N., & Allin Geo, A. V. (2019). Smart server IOT based temperature monitoring system. *International Journal of Innovative Technology and Exploring Engineering, 8(9S3).* https://doi.org/10.35940/ijitee.I3118.0789S319.

Khan, W. Z., Aalsalem, M. Y., Zangoti, H. M., Zahid, M., & Afzal, M. K. (2018). Internet of Things based Physical and Environmental Monitoring System for Data Centers. *2018 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications,* 41-44.

Kurniawan, D. E., Iqbal, M., Friadi, J., Borman, R. I., & Rinaldi, R. (2019). Smart Monitoring Temperature and Humidity of the Room Server Using Raspberry Pi and Whatsapp Notifications. *Journal of Physics: Conference Series*, *1351*(1), 012006.

Marzuki, M. J., & Newell, G. (2019). The emergence of data centres as an innovative alternative property sector. *Journal of Property Investment and Finance, 37*(2), 140-152. https://doi.org/10.1108/JPIF-08-2018-0064.

Nasution, T. H., Muchtar, M. A., Seniman, S., & Siregar, I. (2019). Monitoring temperature and humidity of server room using Lattepanda and ThingSpeak. *Journal of Physics: Conference Series*, *1235*(1), 012068.

Olushola, T., & Abiola, J. (2017). The Efficacy of Technology Acceptance Model: A Review of Applicable Theoretical Models in Information Technology Researches. *Journal of Research in Business and Management, 4*(11), 70-83.

Onibonoje, M. O., Bokoro, P. N., Nwulu, N. I., & Gbadamosi, S. L. (2019). An IoT-Based Approach to Real-Time Conditioning and Control in a Server Room. *2019 International Conference on Artificial Intelligence and Data Processing Symposium,* 1-6. https://doi.org/10.1109/IDAP.2019.8875880.

Qureshi, M. R. J., & Ikram, J. S. (2015). Proposal of enhanced extreme programming model. *International Journal of Information Engineering and Electronic Business*, *7*(1), 37.

Rasool, H., Rasool, A., Rasool, U., Raza, A., & Ahmad, W. (2014). Centralized environment and battery monitoring system for server rooms. *2014 IEEE Conference and Expo*

Transportation Electrification Asia-Pacific, 1-5. https://doi.org/10.1109/ITEC-AP.2014.6941035.

Roy, A., Datta, A., Siddiquee, J., Poddar, B., Biswas, B., Saha, S., & Sarkar, P. (2016). Energy-efficient Data Centers and smart temperature control system with IoT sensing. *7th IEEE Annual Information Technology, Electronics and Mobile Communication Conference,* 1-4. https://doi.org/10.1109/IEMCON.2016.7746251.

Yamanoue, T. (2020). Monitoring of servers and server rooms by iot system that can configure and control its terminal sensors behind a nat using a wiki page on the internet. *Journal of Information Processing*, *28,* 204-213. https://doi.org/10.2197/ipsjjip.28.204.

Yumang, A. N., Paglinawan, C. C., Sejera, M. M., Lazam, A. S., Pagtakhan, J. C., & Santos, J. S. B. (2017). ZigBee Based Monitoring of Temperature and Humidity of Server Rooms using Thermal Imaging. *Proceedings 6th IEEE International Conference on Control System, Computing and Engineering,* 452-454.

APPENDICIES

Appendix 1: Requirements Gathering Questionnaire Sample

Does your company data centre/server room have an environmental monitoring system? *

- ○ Yes
- ○ No
- ○ Not Applicable

If you have selected yes, briefly describe how your current data centre environmental monitoring system operates.

Your answer ____________________

Have environmental factors e.g. extreme temperatures, humidity, water, fire, power surges and spikes caused work disruptions? *

- ○ Yes
- ○ No
- ○ Not Applicable

Which environmental factors do you think affect data centres/server rooms the most? *

- ☐ Temperature
- ☐ Humidity
- ☐ Water
- ☐ Fire
- ☐ Power surges and spikes
- ☐ Other: ____________________

Appendix 2: User Acceptance Testing Questionnaire Sample

On a scale of 1 to 5, how do you rate the developed system? *

1	2	3	4	5
○	○	○	○	○

Do you think the developed system is easy to use? *

- ○ Strongly Agree
- ○ Agree
- ○ Neutral
- ○ Disagree
- ○ Strongly Disagree

Do you think the developed system will improve the operations of the data centre?

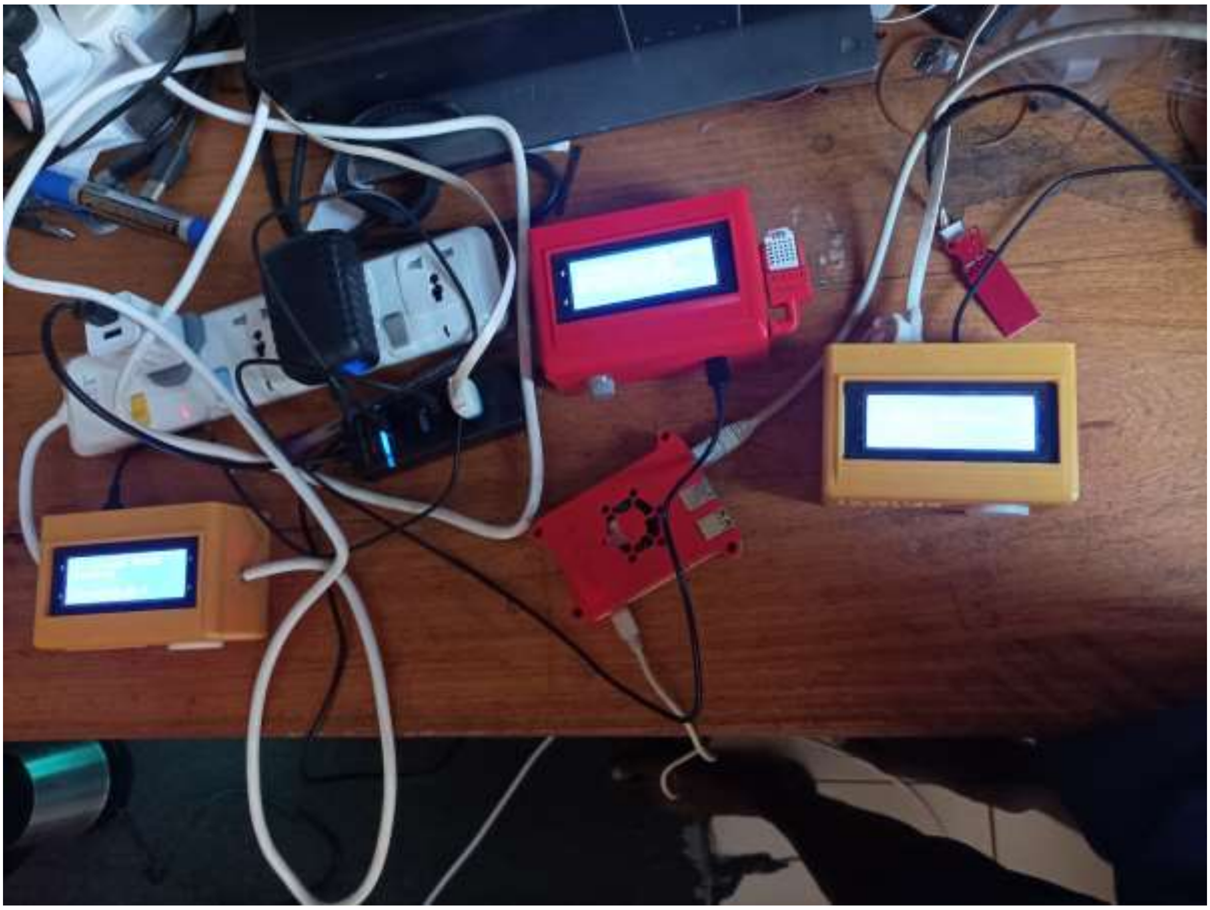
- ○ Strongly Agree
- ○ Agree
- ○ Neutral
- ○ Disagree
- ○ Strongly Disagree

Would you be willing to start using the developed system in your data centre?

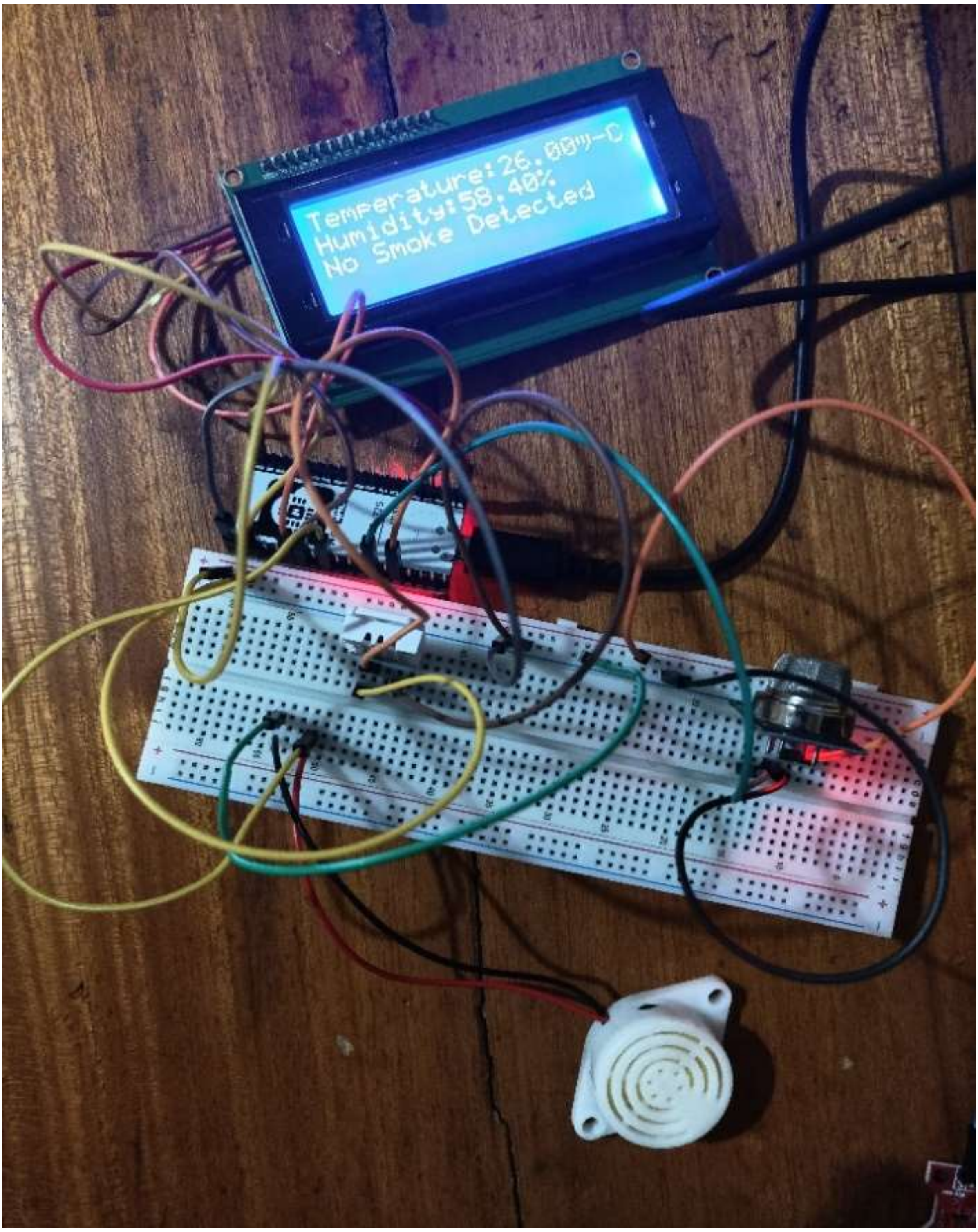
- ○ Strongly Agree
- ○ Agree
- ○ Neutral
- ○ Disagree
- ○ Strongly Disagree

Appendix 3: Ender-3 3D Printer in Action

Appendix 4: System Sensor Nodes and Sink Node

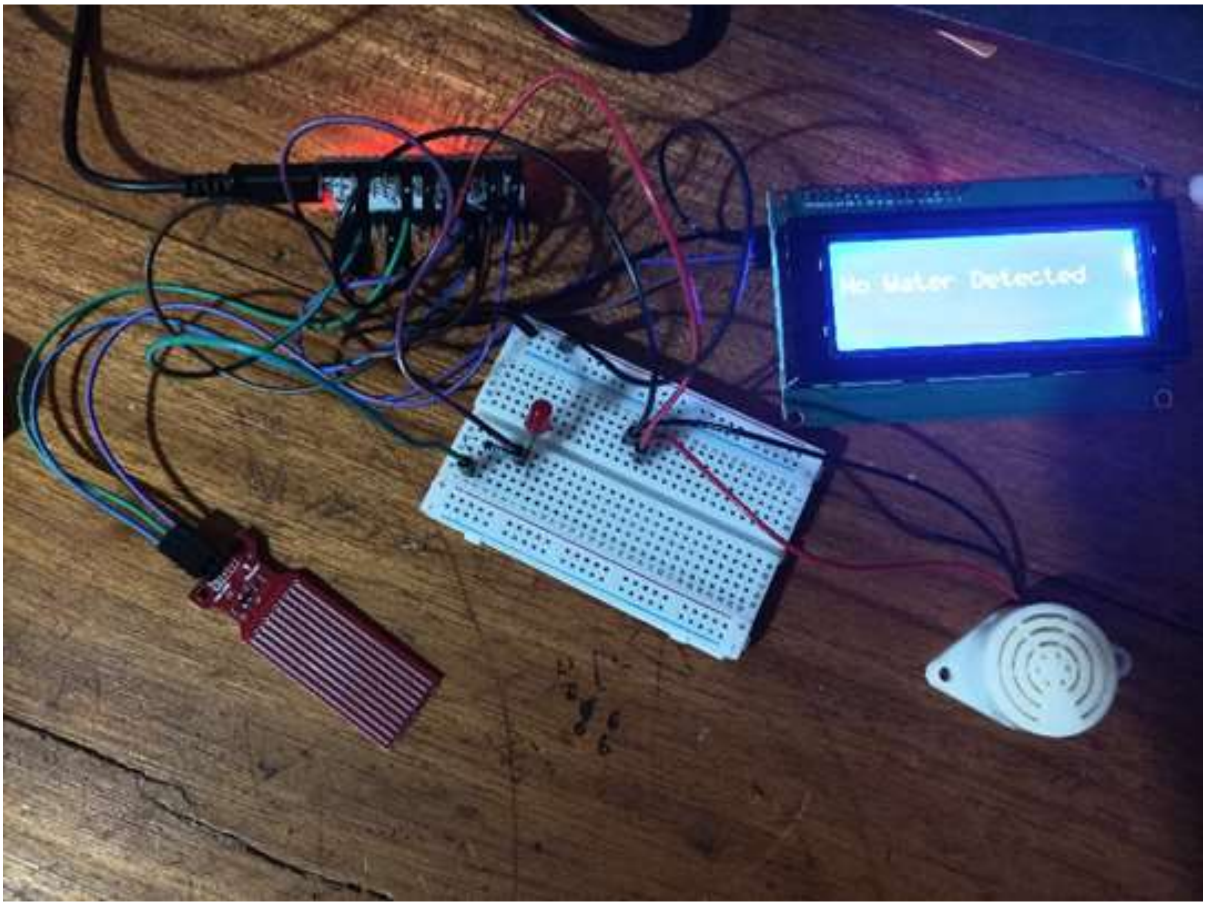


Appendix 5: Sensor Node 1 Breadboard Configuration

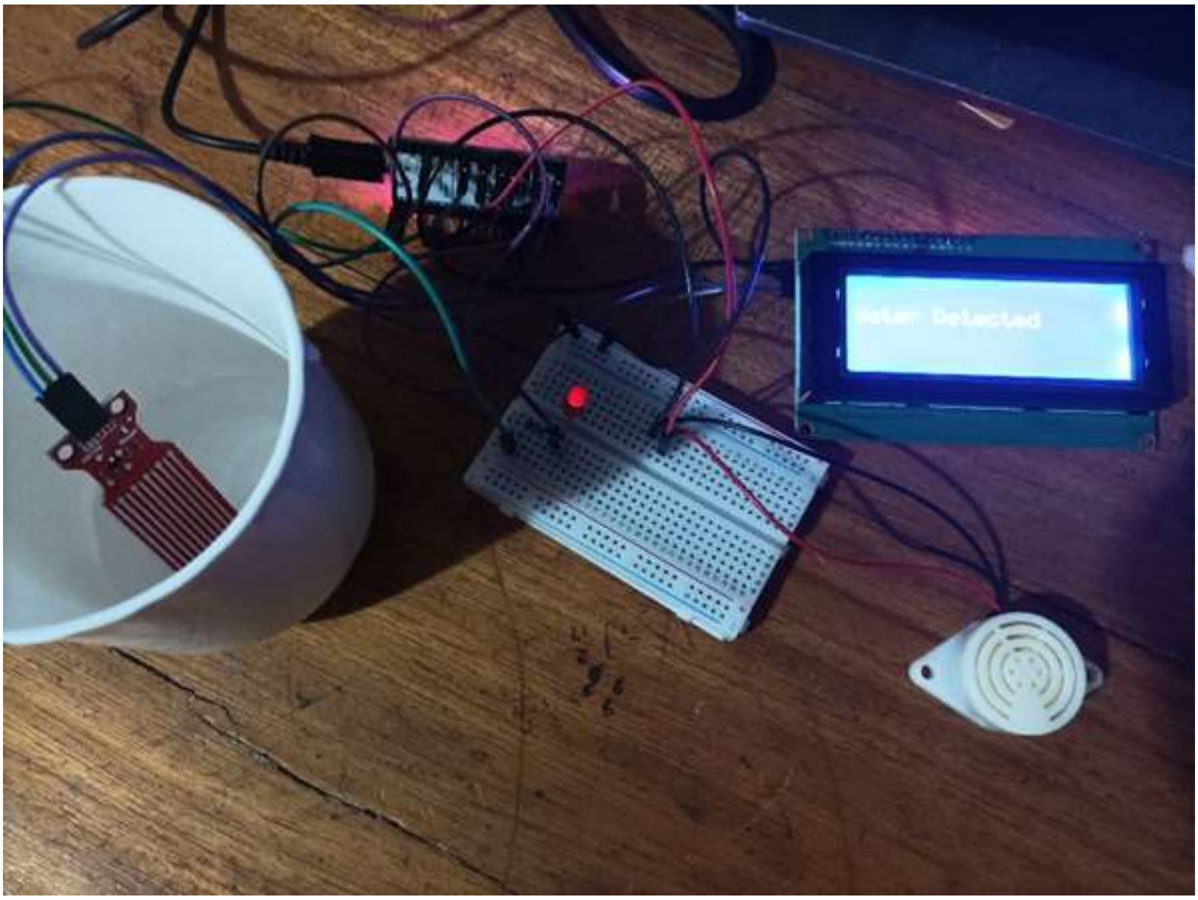


Appendix 6: Sensor Node 1 Side View

Appendix 7: Sensor Node 3 Breadboard configuration



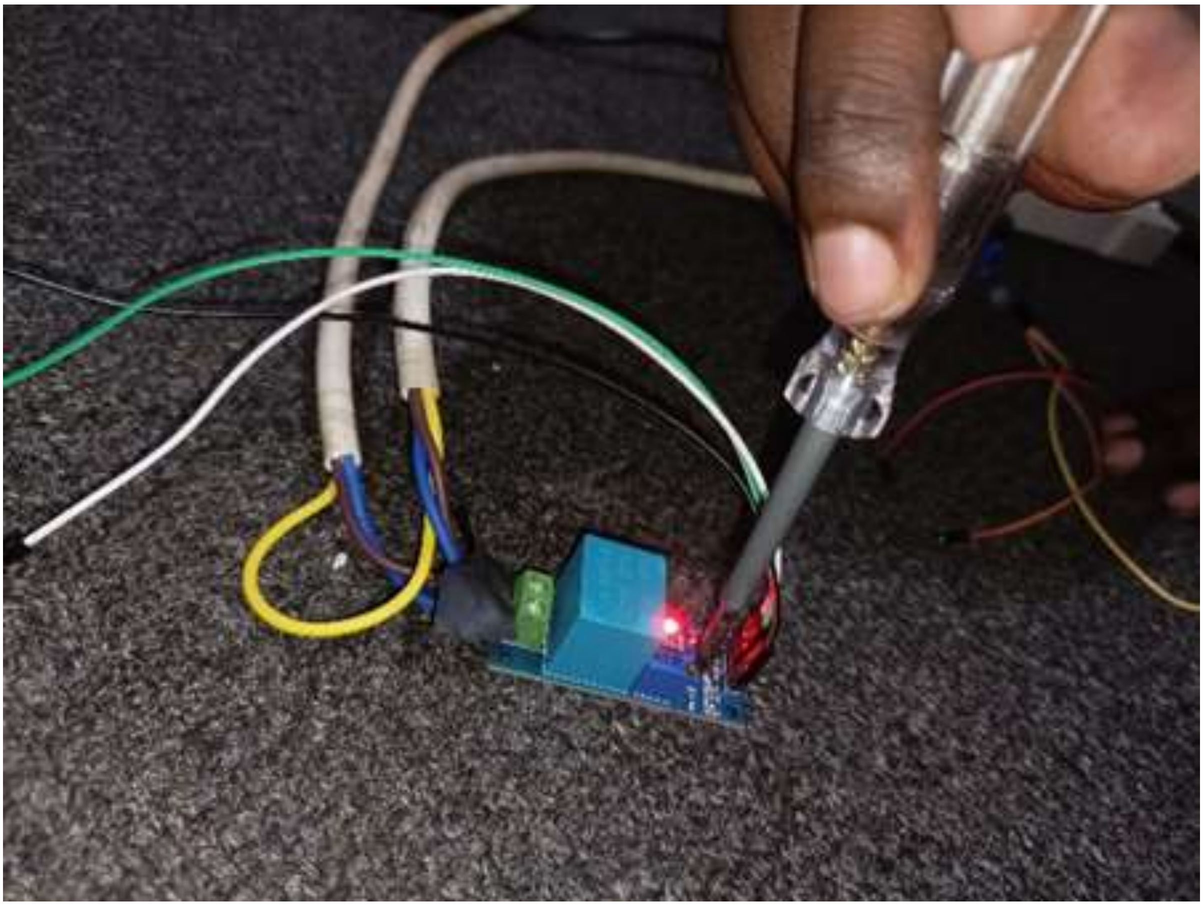
Appendix 8: Sensor Node 3 Breadboard Configuration with Water Level Sensor in Water



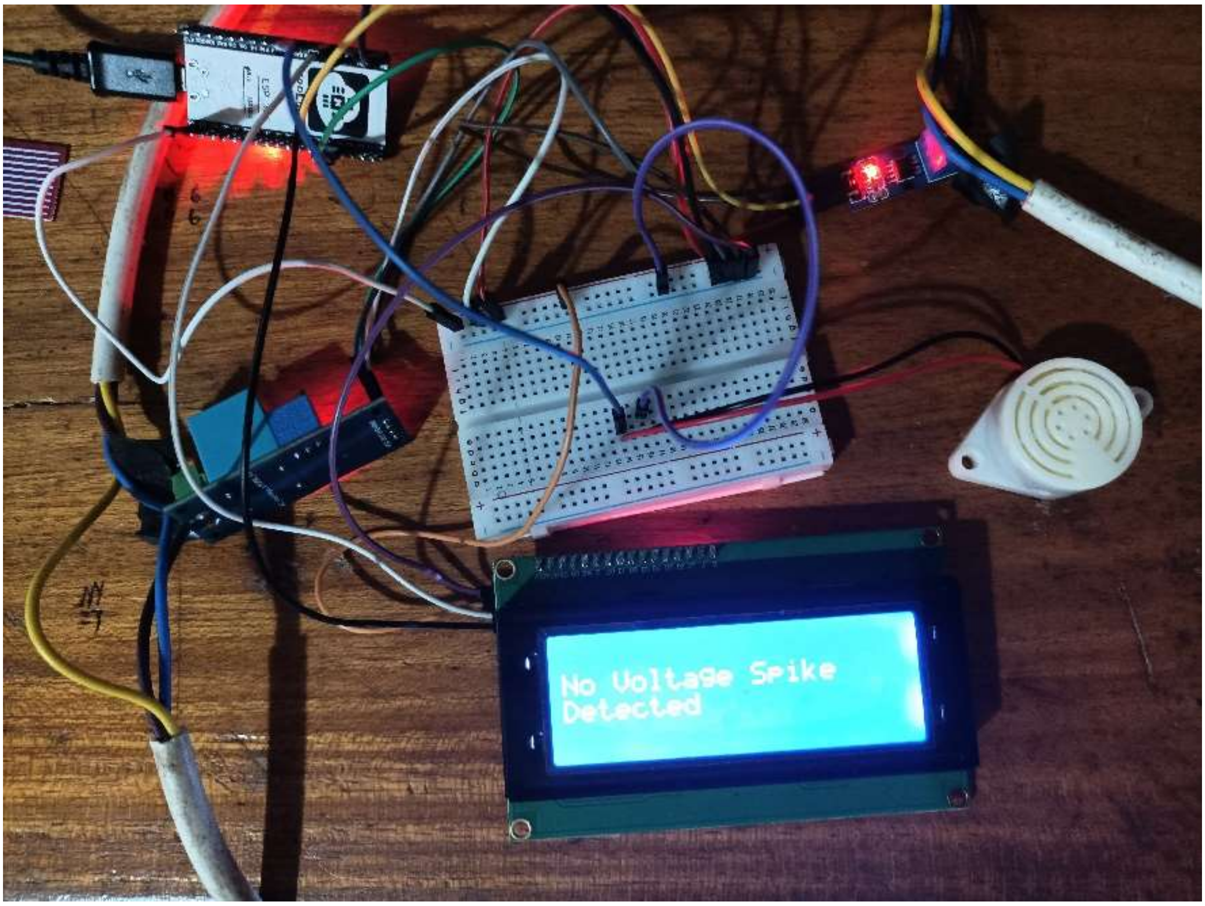
Appendix 9: Sensor Node 3 Side View

Appendix 10: Sensor Node 3 with Water Level Sensor in Water

Appendix 11: Calibration of the Voltage Sensor



Appendix 12: Sensor Node 4 Breadboard Configuration



Appendix 13: Sensor Node 4 Side View

Appendix 14: Sink Node Deployed in netLabs!UG Data Centre

Appendix 15: Sink Node Lower View

Appendix 16: Sample Code for the Navigation Bar of the Web Dashboard

```
import dash
import dash_core_components as dcc
import dash_html_components as html
import dash_bootstrap_components as dbc


colors = {
    'background': '#111111',
    'text': '#003366'
}

SIDEBAR_STYLE = {
    "position": "fixed",
    "top": 0,
    "left": 0,
    "bottom": 0,
    "width": "20rem",
    "padding": "2rem 1rem",
    "background-color": "#f8f9fa",
    "color":"#336699",
}


CONTENT_STYLE = {

    "margin-left": "16rem",
    "margin-right": "3rem",
    "padding": "2rem 6rem",


}

def create_navbar():
    sidebar = html.Div(
        [
            html.H2("Monitoring Dashboard", className="display-4"),
            html.Hr(),
            html.H6(
                "System Menu", className="lead", style={'align':'center'}
            ),
            dbc.Nav(
                [
                    dbc.Button(
                        dbc.NavLink("Sensor Node 1 ", href="/",
active="exact"),
                        style={
```

```
                            'width': '150px'
                        }
                    ),
                    html.Br(),
                    dbc.Button(
                        dbc.NavLink("Sensor Node 2", href="/node_2",
active="exact"),
                        style={
                            'width': '150px'
                        }
                    ),

                    html.Br(),
                    dbc.Button(
                        dbc.NavLink("Sensor Node 3", href="/node_3",
active="exact"),
                        style={
                            'width': '150px'
                        }
                    ),

                    html.Br(),
                    dbc.Button(
                        dbc.NavLink("Sensor Node 4", href="node_4",
active="exact"),
                        style={
                            'width': '150px'
                        }
                    ),


                    html.Br(),
                    dbc.Button(
                        dbc.NavLink("Forecasts    ", href="forecasts",
active="exact"),
                        style={
                            'width': '150px'
                        }
                    ),
                ],
                vertical=True,
                pills=True,
            ),
        ],
        style=SIDEBAR_STYLE,
    )
    return sidebar
```

Appendix 17: Poster Presentation

A Machine Learning-Based IoT Environmental Monitoring Platform for Data Centres

Research Team: Wayne Steven Okello, Dr. Silas Mirau, Prof. Michael Kisangiri, Dr. Andrew Katumba and Dr. Edwin Mugume

Introduction

Data centres are a crucial part of many organizations in the world today consisting of expensive assets that store and process critical business data as well as applications responsible for their daily operations. Unconducive environmental conditions can lead to decline in performance, sporadic failures and total damage of equipment in the data centers which can consequently lead to data loss as well as disruption of the continuity of business operations. To address this, we have developed an environmental monitoring system that employs IoT and machine learning to monitor and predict important environmental parameters within a data centre setting.

System Architecture

The system comprises of a heterogeneous WSN with four (4) sensor nodes and a sink node as illustrated in Figure 1

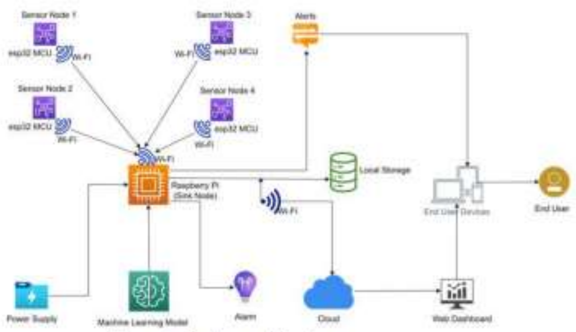


Figure 1: System architecture

The system was further decomposed into five (5) components i.e. sensing, processing and storage, alerting, monitoring and analysis and time series forecasting.

Sensing Unit

Sensor Node 1 and Sensor Node 2: They are composed of an ESP32 microcontroller, a DHT22 sensor, an MQ2 sensor, a 20x4 LCD display module and a buzzer.

Sensor Node 3: It consists of an ESP32 microcontroller, a water level sensor, a 20x4 LCD display module and a buzzer.

Sensor Node 4: It is composed of an ESP32 microcontroller, a ZMPT101B voltage sensor, an ACS721 current sensor module, a 20x4 LCD display module and a buzzer.

Processing and Storage

A Raspberry Pi 4 was set up as the sink node. The sensor nodes send or publish their readings to a local MySQL database via Wi-Fi by making an HTTP POST request to the Raspberry Pi LAMP server.

Alerting

Each sensor node consists of a buzzer that produces an audio alarm in case an undesired event is detected. In addition, sensor nodes are integrated with email, SMS and WhatsApp alerts that are sent to data centre administrators in the event of unfavorable environmental conditions in the data centre.

Monitoring and Analysis

A customized web dashboard that utilizes the ThingSpeak platform iframes was developed using the dash framework, to displays real-time trends of the environmental parameters measured by the different sensors.

Figure 2:. Web dashboard displaying temperature and humidity trends

Time Series Forecasting

The dataset was collected from sensor node 1 deployed in the netLabs!UG data centre, Uganda

S/N	Feature	No. of Observations
1	id	8,010
2	date_time	8,010
3	temperature	7,998
4	humidity	7,998

Table 1: Dataset description

The models were trained using Facebook Prophet, ARIMA and ES algorithms. Facebook Prophet manifested the best performance with a MAPE of 5.77% and 8.98% for the temperature and humidity models respectively.

Metric	Facebook Prophet	ARIMA	ES
MAPE	5.771	6.459	6.495
MAE	1.480	1.667	1.672
RMSE	1.819	2.041	2.037

Table 2:. Performance evaluation for the temperature models

Metric	Facebook Prophet	ARIMA	ES
MAPE	8.981	10.294	10.160
MAE	4.199	4.635	4.594
RMSE	5.256	5.272	5.218

Table 3:. Performance evaluation for the humidity models