ENVIRONMENTAL MONITORING USIN IOT

TABLE OF CONTENT

ABSTRACT

The Internet of Things (IOT) is known to play a critical capacity in regular daily existence the entire way through inescapable sensor correspondence networks that epitomize our general climate. Such framework is gives the plan capacity to screen fundamental actual occasions produced information that can be moved and put away in the cloud from which it is feasible to share this data by means of utilization and choice is made to make a move for a happened occasion. Ecological Monitoring framework uses sensors for encompassing area moistness and temperature.

These information could be used to animate transient conduct like gadget becoming hot or getting cool down and other long haul insights of the gadgets. The detected information will be sent to cloud space, and the cloud is gotten to by a Smartphone application and results are introduced to end clients.

The review is done the sort sensors, microcontroller and its ability, investigation of various kind economical organization arrangement for ceaseless information assembling and checking. Different instruments used to investigate the information put away on the cloud

TABLE OF FIGURE

TABLE OF ABBREVATION

ABBREVATION	DEFINITION
IOT	Internet Of Things
TLS	Transport Layer Security
SDG	Sustainable Development Goals
ML	Machine Learning
ICT	Information and Communication technology
UDS	Urban Drainage Systen
UWM	Urban Water Management

I.INTRODUCTION

The Internet of Things (IOT) is set to modernize our environment by letting us to control and monitor essential phenomenon in our surroundings via using devices capable of capturing data, evaluating and wireless transmission of information to storage server, like the cloud, which collects, assesses and provides these data in a meaningful way. From the cloud this information can be obtained, based on appropriateness and demands, via numerous front end user interface design such as mobile and web applications.

The Internet strikes at the heart of this transition and plays a big part in secure, effective, and fast transmitting data between fog and applications and end users. The definition of a traditional end user or server in the Internet is changed in this new paradigm and hosts consist of the devices or objects therefore the term Internet of Things. The "devices and sensor" will detect and transmit data such as temperature, pressure, humidity, sound, emissions, object tracking, vitality of the patient. Environment aspects tracking is an important IOT system that mainly includes data collection through the sensing system, as well as reviewing this data for successful short-term measures such as remote management of heating or cooling devices and long-term data interpretation and observations. The Internet of Things (IOT) has proven extremely successful and is expanding rapidly in all fields. Automated monitoring system will control and monitors by using the data processing micro - controller, data gathering sensors, and wireless data transmission sensor network. Nodes have sensing and transmission/ receiving capabilities on the wireless sensor network.

Large number sensor nodes are either randomly fix on the confined area or are based on the structure defined. This paper focuses mainly on the diverse wireless sensor network issues considering such as higher energy consumption in the sensor nodes leads to node failure after some communications, reduction in data transfer rate causes network suffering, network destabilization leads to loss of data, long-range data transfer may result in even more energy, number of nodes can increase node traffic

CHAPTER 2

2.SYSTEM REQUIREMENT

2.1.Hardware Requirements:

Sensors:

Depending on the type of environmental data you want to monitor (e.g., temperature, humidity, air quality), you'll need the appropriate sensors.

Data Acquisition Systems:

These are needed to collect data from sensors and convert it into a digital format.

Communication Equipment:

To transmit data, you may need equipment such as modems, routers, or cellular modules.

Power Supply:

Ensure a reliable power source, and consider backup solutions in case of power outages.

Data Storage:

You'll need storage solutions for archiving data.

Environmental Protection:

Enclosures or housing to protect the hardware from the monitored environment.

Software Requirements:

Data Processing:

Software to process, clean, and analyze the collected data.

User Interface:

A user-friendly interface for viewing and interacting with the data.

Data Storage Management:

Systems to store and manage the historical data.

Alarm and Notification System:

To alert users or stakeholders in case of abnormal conditions.

Remote Access:

Software for remote monitoring and control.

Data Visualization:

Tools for creating charts, graphs, and reports to make the data understandable.

Security:

Implement security measures to protect the data and system from unauthorized access.

Environmental Requirements:

Environmental Conditions:

- Specify the range of environmental conditions your system will monitor (e.g., temperature range, humidity levels, pollution levels).
- Ensure that the hardware is designed to withstand these conditions.

Location:

- Determine where the monitoring system will be deployed (indoors, outdoors, remote locations).

CHAPTER 3

3 .SYSTEM DESIGN

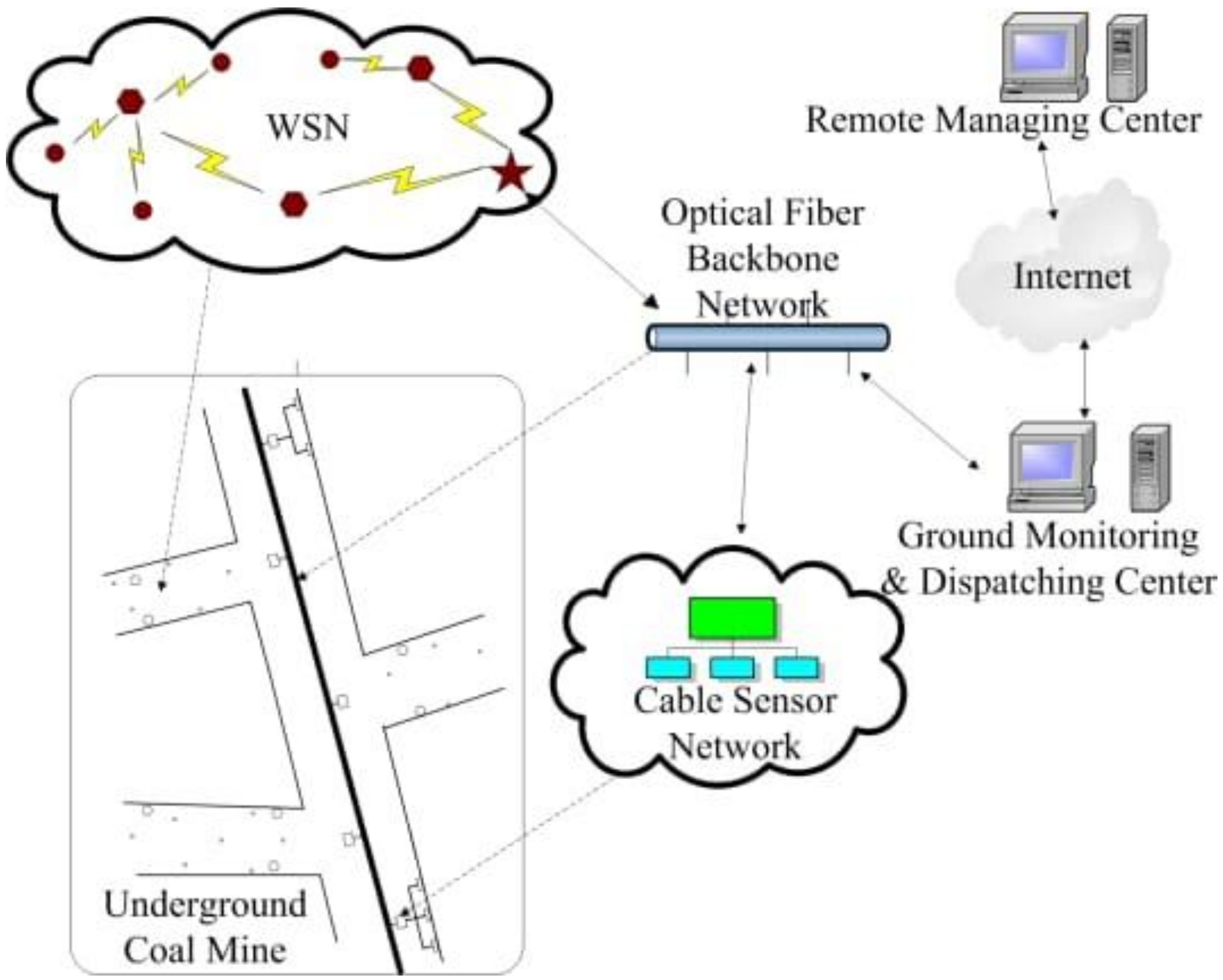


Fig.3.1.General architecture of an environmental monitoring

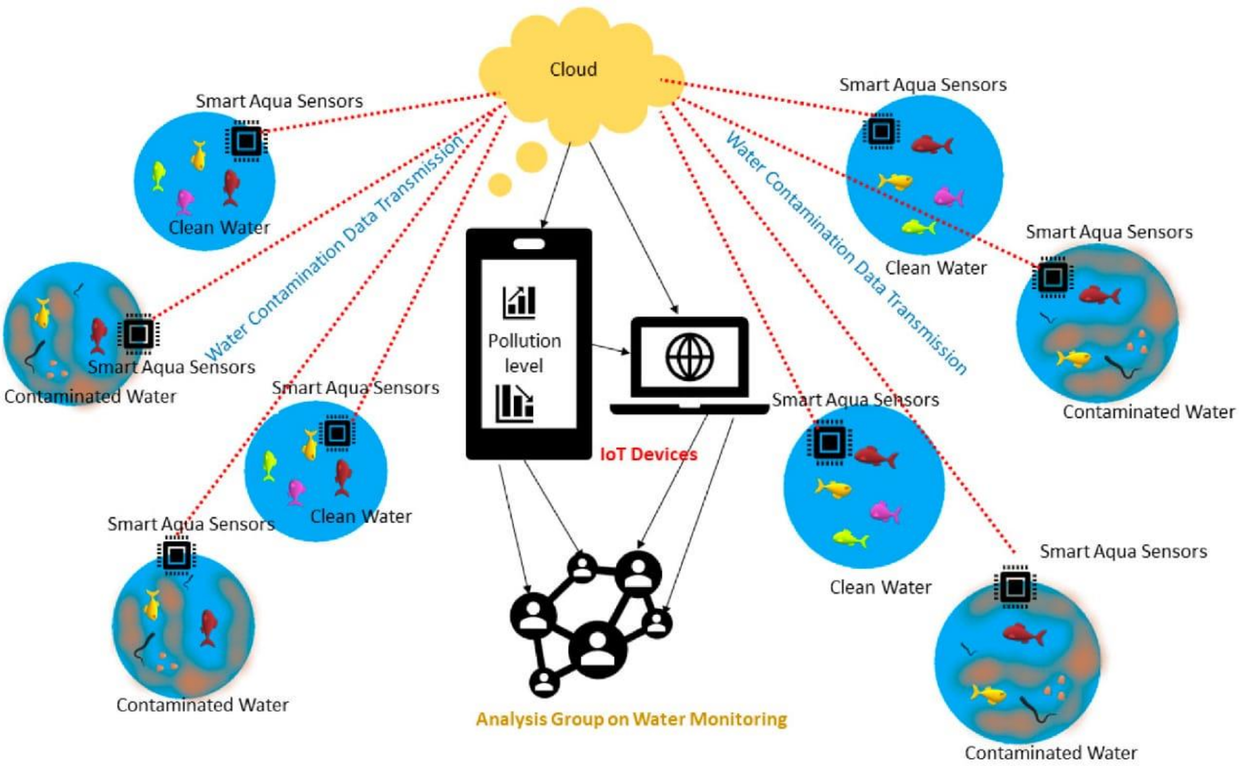


Fig3.2.Overall structure of environmental monitoring

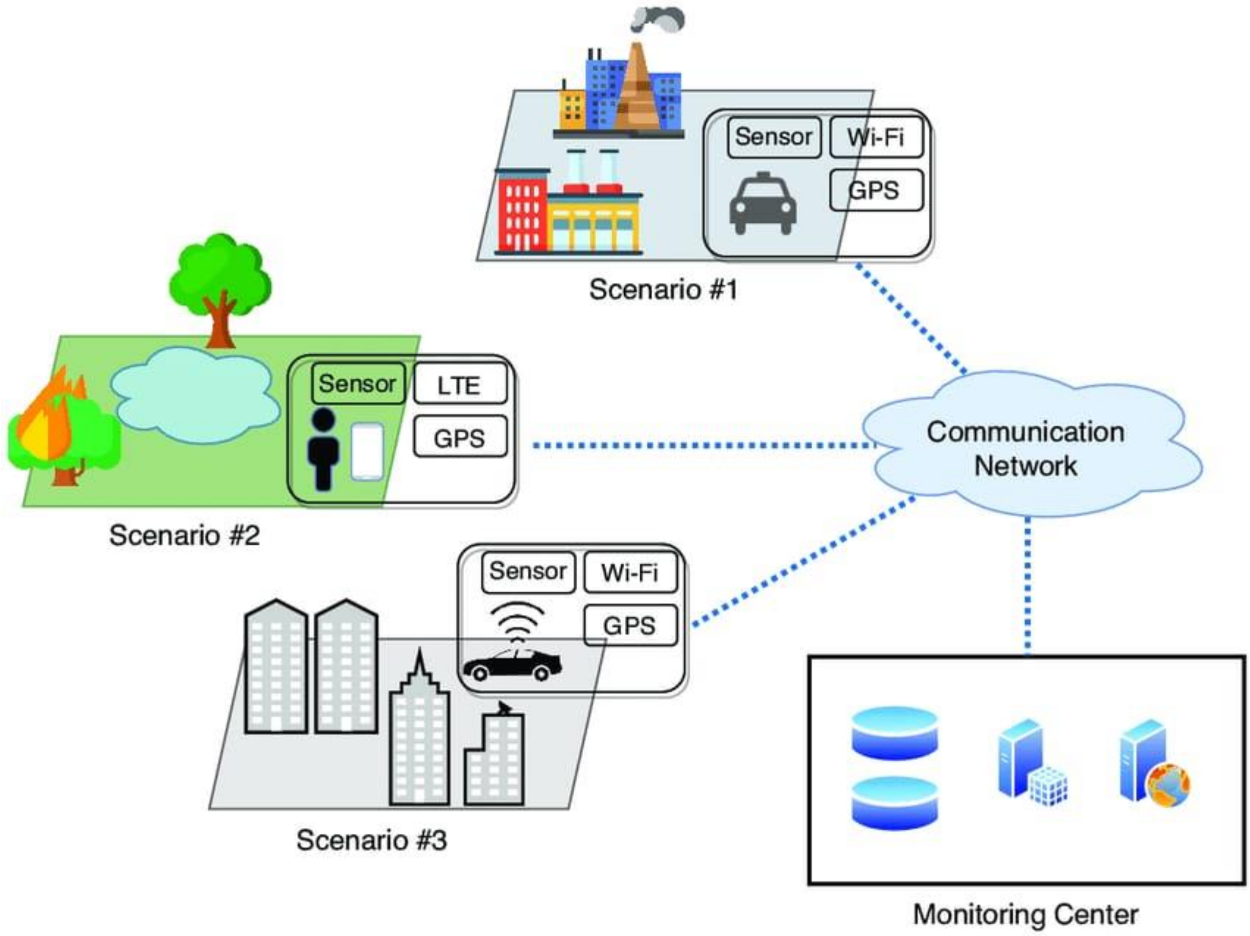


Fig.3.3.Architecture of an environmental monitoring

CHAPTER 4

4 . SYSTEM ANALYSIS

4.1 Air quality measurements standards

The measurement of air quality is carried out by state services based on standards acceptable levels of various substances in the air. Polish and EU standards limit the usual average levels of these pollutants.

Polish standard[26] defines maximum levels for dust pollution for two sizes: PM2.5 and PM10. PM2.5 is the concentration of particulate matter with an aerodynamic diameter of grains to 2.5 microns, and PM10 – to 10 microns. Acceptable levels of dust concentrations are determined for the average daily (PM10, the permissible level is 50 µg/m3), or an average annual (PM2.5, the permissible level is 25 µg/m3). The standard [26] was defined two additional levels of pollution (only for PM10):

- the level of informing the public (value - 200 µg/m3)

- the alarm level (300 µg/m3).

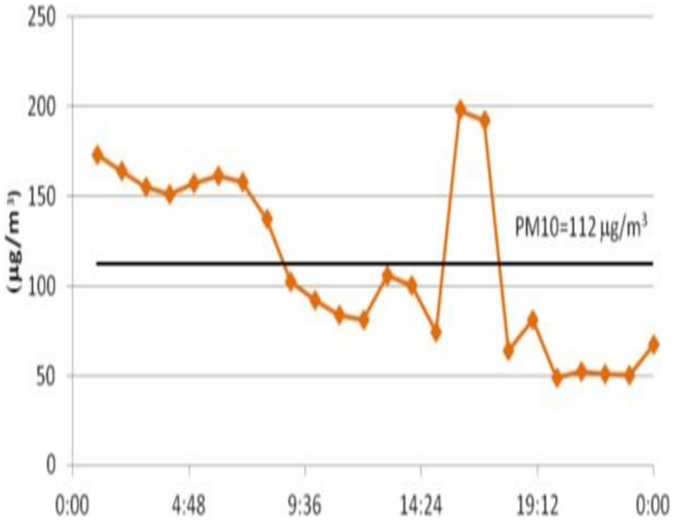


Fig 4.1 The results of the daily measurement of the dust concentration (PM10; Dietla str., Krakow, Poland; March 18, 2016) with the marked daily average value (Source: [27])

.

Both of these levels are defined in the standard for the average daily. Short-term increases in the level of pollution do not therefore constitute formal causes for the alarms (Fig. 2).

Such information may, however, be useful in the choice time and places of jogging or resting by citizens. For example, afternoon jogging would be better start after 18 and even better after 20 in the neighbourhood of the measuring station with the data from Fig. 2.

The concentration at this time was four times less than 16-17 hours.

4.2 Inexpensive environmental monitoringsystem architecture

Inexpensive Environmental Monitoring system consistsof two basic components (Fig. 3):

- Mobile measuring stations,
- Server-side software.

Many mobile measuring stations, placed on the public transport vehicles (buses, taxi) or private (volunteers) during motion collect data on theconcentration of dust in the air. They used algorithms to optimize energy consumption and make sense of measurement and act as IoT agents.

Measuring stations transfer the data to the server ina situation where they will in the field of public WiFi network (e.g. accessible to the public in buses or other free Hot Spots located in different places of the city).

This solution reduces the operating costs of the system.In addition, the algorithms of transfer the measurement data to the server optimize energy consumption andtransmission time.

On the server side are implemented three basicfeatures:

- Mobile measuring stations (IOT agents) registration,
- Measurement data acquisition,
- Data presentation in various forms, layouts and profiles.

Server-side part of the system uses:

- HTTP server (Apache)
- PHP parser,
- database server (My SQL)
- external services (Google Maps)
- dedicated software (HTML, CSS, JavaScript, PHP).

The software has been developed using the responsive web design (RWD) approach and model-view-controller (MVC) architecture.

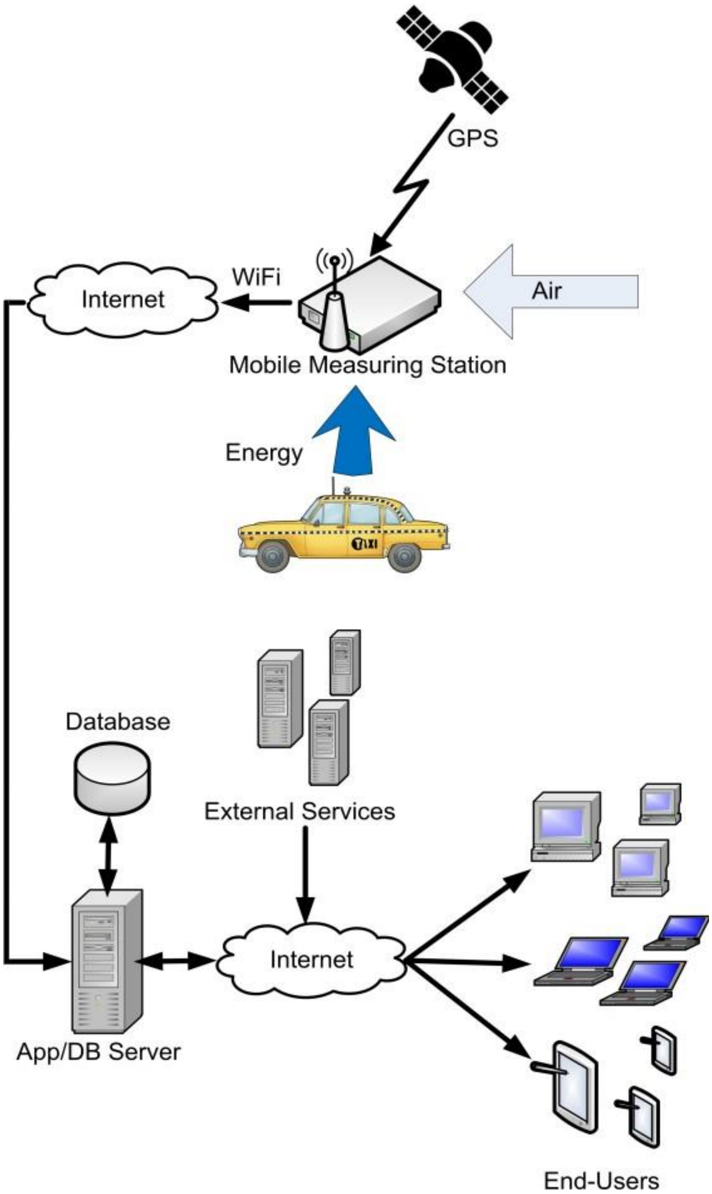


Fig 4.2 Architecture of the Inexpensive Environmental Monitoring System.

4.3 The moving IOT agent for dustconcentration measurements

The measuring station, which also acts as IOT agent, is designed in the compact way to fulfil the mobility requirements.

The overall architecture of the station is shown in Fig. 4.

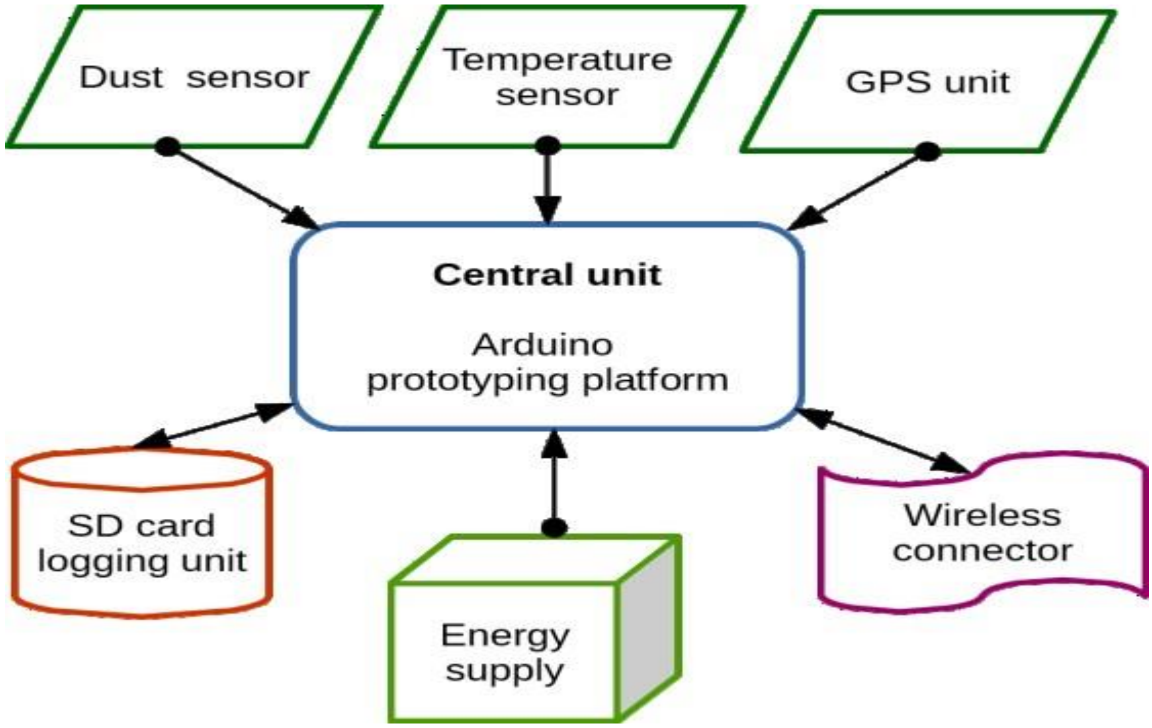


Fig . 4 .3 Architecture of the moving measuring station of dust concentration.

Station consists of the few sensors. The main one is the dust sensor for measuring the dust concentration, which gives a good indication of the air quality in an environment. However due to the specific way of dust sensor automatic measurement, this values cannot be assumed as reference values for dust concentration.

The other two sensors are: GPS for determining the position and temperature sensor for acquire working temperature. The range of working temperatures is important due to the dust sensor operation temperatures which are from -10 ^{o}C to +65 ^{o}C.

The heart of the IOT agent is the Arduino prototyping platform with At mega chipset onboard. Arduino platform allows for programming all components together and act as single IOT agent. The measurement data is written to SD memory card for later transmission. In the designed system the WiFi module connected tothe Arduino board acts as wireless connector. Wiring of the designed measurement station is shown in Fig. 5.

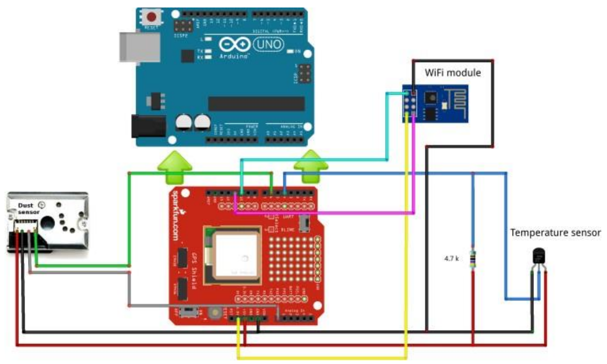


Fig.4.3 Wiring of the dust measuring IoT agent.

On the top of Arduino Uno board the GPS shield is set. This shield is also equipped in SD card slot and additionally allows for transparent wire connection of external devices to Arduino board.

All other devices of an IOT agent i.e. dust sensor, temperature sensor and WiFi module are properly connected to the supplying pins (GND, +5V, +3.3V) and digital and analogue input pins (Fig. 5). An agent is supplied with the electrical system of the car or the USB power bank of about

10000 mAh capacity. That allows for approximately6 hours measurement and communication session.

The hardware parameters of an IoT mobile dust level measuring station are shown in Table 1. Total cost of thestation with additional hardware (cables, resistor, andcasing) is about 125 €.

The agent performs measurements according to flowchart shown in Fig. 6. The series of measured data are averaged (10 measurements of dust level and temperature per second) and the averaged value is written to the file. That was assumed, that with allowed speed limit in the cities in Europe which is 50 km/h the vehicle changes its position by the 13.8 m during one second.

Table 1. Hardware parameters of IOT mobile dustconcentration measuring station.

Hardware element	**Description**
Arduino programmable board	Arduino Uno board which is used as a main platform for sensors and dedicated agent software. Maximum current consumption is 46.5 mA with supplying voltage equals to 5V. Price 25€
Optical Dust Sensor	The sensor GP2Y1010AU0F has a very low current consumption (20mA maximum), supplying voltage up to 7V of direct current. The output of the sensors is an analogue voltage proportional to the measured dust density, with a sensitivity of 0.5V/0.1mg/m3. Price 14€
WiFi module	Wireless ESP8266 module, WiFi 802.11 b/g/n, working frequency 2.4 GHz, PCB antenna, current consumption 140 mA during transmission. Price 3€
GPS and SD card shield	The shield is SparkFun GPS Logger Shield - GPS GP3906-TLP with SD Card slot. It allows for GPS data recording and storing data on SD Card. Current consumption is 30 mA. Price 50€
Power bank	Polymos 10 AIR - 10000mAh power bank. Output current 2.1 A. Price 25€

However according to the [28] average speed varies from 19 to 35 km/h, which translates into a travelled distance from 5.3 to 9.7 meters per one second.

During the movement of the vehicle the IOT agent tries to connect to predefined access points or Hot Spots. Shared by smart phone and other WiFi connection can also be used. It agent succeeds in connecting to Internet it tries to send to server stored data. The flowchart ofdata sending is shown in Fig. 7. Agent during sending data is working according to FIFO algorithm.

The structure of sending to the server data frames is as follow:

Frame := MAC, Number of measurements, {GPS position, Data Time, Type, Measurement;}

.

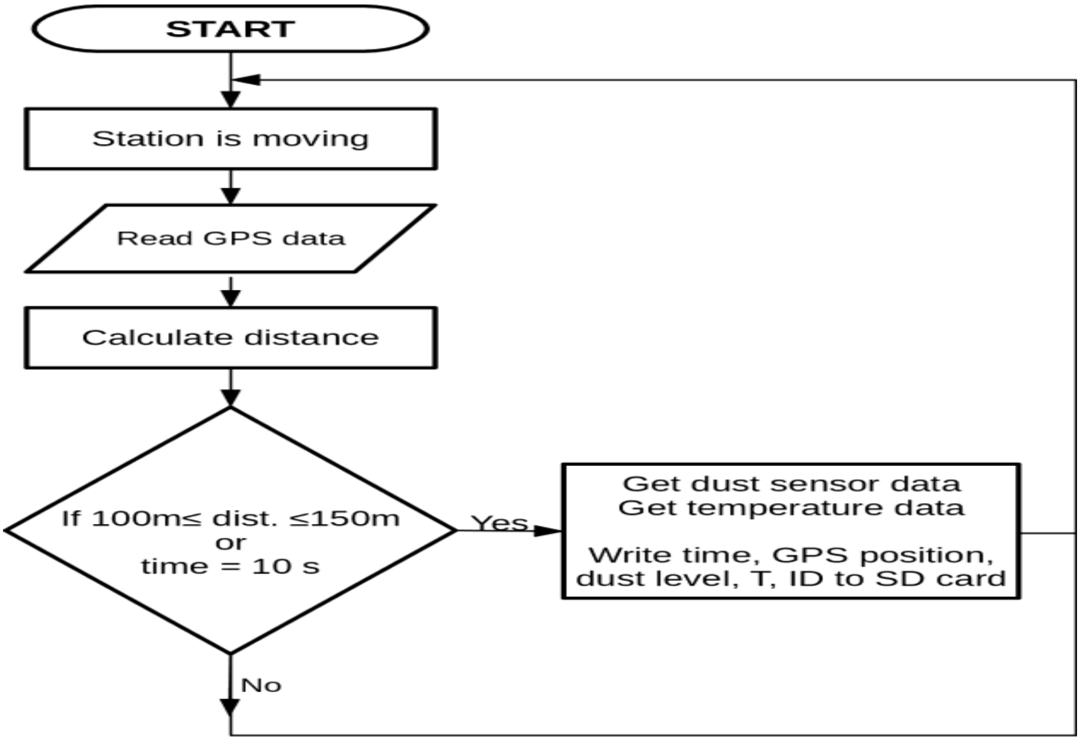


Fig 4.3 Flowchart of the sending the measurement data to server protocol

4.4 Advantages

- Data. Enhanced data collection. Instant data access. Improved data-based decision making. Time & Money. Better time management. Lower operating costs.

- The main objective of environmental monitoring is to manage and minimize the impact an organization's activities have on an environment, either to ensure compliance with laws and regulations or to mitigate risks of harmful effects on the natural environment and protect the health of human beings.

- Implementing IOT-based environmental monitoring allows for continuous detection and measurement of VOC levels in the environment. By detecting increased VOC levels at an early stage, immediate action is taken to mitigate potential health risks and ensure the safety of individuals

- Environmental monitoring or management is the process of measuring and assessing workplace conditions to evaluate health risks to workers. This practice is especially important in businesses that use hazardous substances, such as heavy metals. It includes periodic health examinations of workers and environmental impact tests.

- The advantages of environmental monitoring include preventing occupational diseases, improving the company's public image and reducing environmental pollution.

Disadvantages

- Expensive.
- Requires regular calibration.
- Difficult to analyze statistics and examine long-term trends.
- Precludes use of data in computational tools.
- The stigma around employee monitoring.
- The stigma with employee monitoring is that it negatively impacts employee morale.
- Feelings of distrust. Some employees may feel like their employers don't trust them.
- Employee privacy concerns.
- Legal issues in monitoring

CHAPTER 5

5.PROJECT DESCRIPTION

Internet of Things is a platform where every day devices surrounding us become smarter, every day collecting the data and processing becomes intelligent, and every day communication becomes more useful for people. The Internet of Things (IOT) has been defined in Recommendation ITU-T Y.2060 (ITU-T Study, 2015) as “a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies”. While the Internet of Things is still evolving, its effects have already stared in making incredible changes in many areas as a universal solution media for providing services in completely different ways. Gartner (Rivera, Rob, 2014) reported that 25 billion devices will be connected to the internet by 2020 and those connections will facilitate the used data to

analyze, preplan, manage, and make intelligent decisions autonomously. The US National Intelligence Council (NIC) has embarked IOT as one of the six ‘‘Disruptive Civil Technologies” (Council National Intelligence, 2008). In this context, we can see that service several sectors, such as: smart city, e-health ,e-governance, transportation, e-education, retail, agriculture, process automation, industrial manufacturing, logistics and business management etc., are already getting benefited from various forms of IOT paradigm and technology. First wave of IOT systems in smart city domain emphasized on connecting sensor using lightweight protocols such as Co AP and XMPP (old fashioned publish/subscribe communication model). The Smart-Object devices with domain specific smart automatic rules are rapidly replacing first wave of IOT devices (Kortuem et al., 2010). Although these devices do not utilize semantic technologies (Park et al., 2014), they provide higher-level of sensor-to-application communication than just plain transfer of raw sensor data .Sensor data representation formats used in a sensor networks is various, and most of them cannot understand the value’s meaning in other systems. Therefore, values and information from the systems must be prepared shared and provided additional metadata information for other applications .The second challenge is develop knowledge discovery techniques and services for big smart city data storing, transfer and provide useful analytics.

Most of the current IOT services (figure 1) require IOT applications (BIG IOT API) to have knowledge of IOT middleware (communication software) and sensors or sensor networks for accessing IOT resources. Heterogeneous IOT middleware are not easy to be accessed by different applications since each IOT gateways has no open and standardized Application Programming Interface (APIs). Various organizations such as the Open IOT Alliance, All Seen Alliance, and IPSO Alliance are working on standardization of „Internet of Everything” communication protocols to provide interoperability between various solutions.

CHAPTER 6

6 .PROJECT IMPLEMENTATION

The implementation of the IOT-based Environmental Monitoring System involves a systematic integration of hardware, software, and communication protocols to achieve real-time data collection, analysis, and presentation.

In the hardware layer, an array of environmental sensors, including air quality sensors, temperature and humidity sensors, water quality sensors, and soil moisture sensors, are strategically deployed across the target area. Microcontrollers, such as Arduino or Raspberry Pi, serve as the brains of the operation, interfacing with the sensors, collecting raw data, and performing initial processing. These microcontrollers are equipped with wireless communication modules, such as Wi-Fi or Lo Ra, ensuring seamless transmission of data to the central monitoring system.

The communication layer employs protocols like MQTT or HTTP to facilitate the efficient exchange of data between the sensors and the central hub. Robust encryption mechanisms are implemented to safeguard data integrity during transmission, ensuring the privacy and security of the collected environmental information.

Data processing and storage occur in the cloud-based layer, where advanced algorithms convert raw sensor data into meaningful insights. A secure cloud database is utilized for storing processed data, enabling historical analysis and trend identification. This cloud-based architecture not only ensures scalability but also allows users to access environmental data remotely through a user-friendly web interface or mobile application.

The web interface or application layer is designed to provide an intuitive and interactive platform for users to access real-time and historical environmental data. Visualizations, such as graphs and charts, enhance data interpretation, empowering users to make informed decisions based on the presented information.

An intelligent alert system is implemented to notify stakeholders of any abnormal environmental conditions. Dynamic threshold values for each parameter trigger alerts, which are sent through multiple channels, such as email or SMS, ensuring that timely actions can be taken in response to potential environmental threats.

To optimize power consumption and promote sustainability, the power management layer incorporates energy-efficient measures. This includes implementing sleep modes for sensors during periods of inactivity and harnessing renewable energy sources, such as solar panels, to power the monitoring system.

Throughout the implementation, scalability is a key consideration, allowing the system to easily accommodate additional sensors or expand the monitoring network. Regular maintenance checks and procedures for software updates and hardware upgrades are established to ensure the reliability and longevity of the Environmental Monitoring System.

6.1. CIRCUIT DIAGRAM

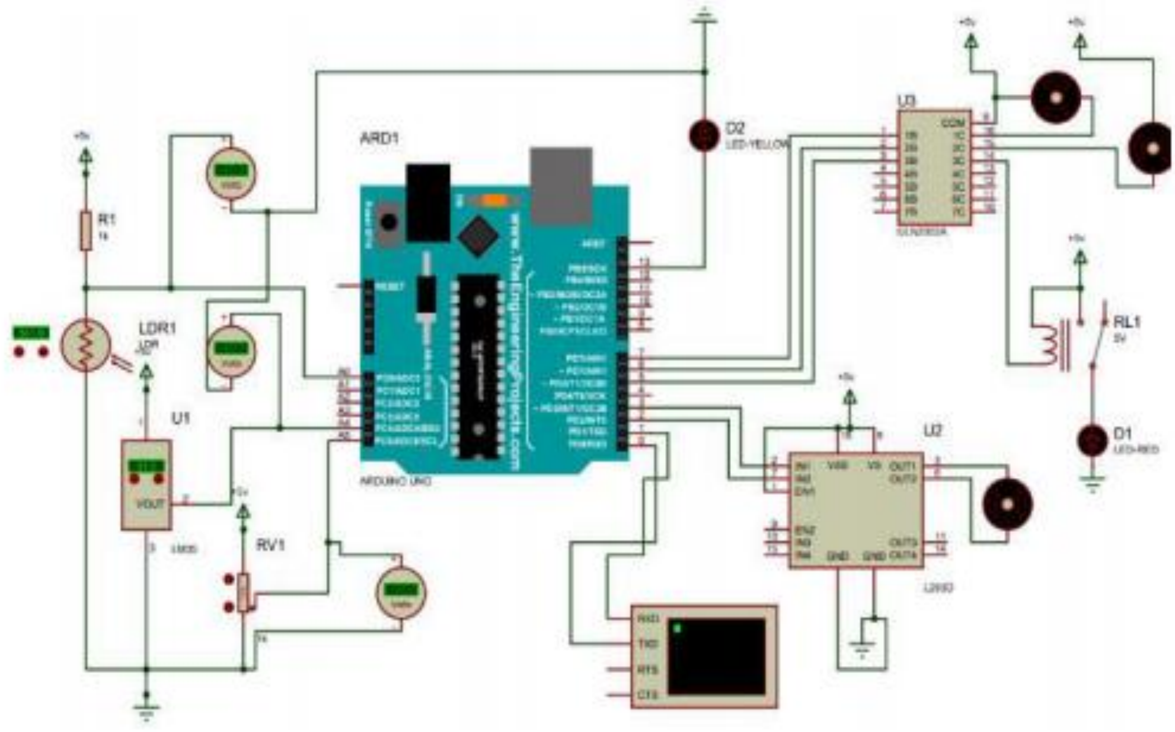


Fig.6.1.Circuit diagram

Fig. Top and Back view of Arduino UNO

6.2.SOURCE CODE

6.2.1 FRONT END(Python)

```
from nfs import datasets
import numpy as np
from nnfs.datasets import spiral_data
import nnfs
from numpy import core
nnfs.init()
import matplotlib.pyplot as plt
# Importance of transposing
inp = np.array([[1, 2, 3, 2.5],
                [2, 5, -1, 2],
                [-1.5, 2.7, 3.3, -0.8]])
w = np.array([[0.2, 0.8, -0.5, 1],
              [0.5, -0.91, 0.26, -0.5],
              [-0.26, -0.27, 0.17, 0.87]])
b = [2,3,0.5]
w2 = np.array([[0.1, -0.14, 0.5],
               [-0.5, 0.12, -0.33],
               [-0.44, 0.73, -0.13]])
b2 = [-1, 2, -0.5]
out1 = np.dot(inp,w.T) + b
```

```
out2 = np.dot(out1, w2.T) + b2

# Introducing non linear data

X, y = spiral_data(samples = 100, classes = 3)

print(X.shape, y.shape)

# plt.scatter(X[:,0], X[:,1], c= y, cmap = 'brg')

# plt.show()

# print(X.shape)

# Defining dense layers

class Layer_Dense:

    def __init__(self, n_inputs, n_neurons):            # n_inputs are inputs, n_output are
output neurons

        self.weights =  0.01*np.random.randn(n_neurons, n_inputs)

        self.biases = np.zeros((1, n_neurons))

    def forward(self, inputs):

        self.inputs = inputs                        # to remember for backpropagation

        self.output = np.dot(inputs, self.weights.T) + self.biases

    def backward(self, dvalues):

        self.dweights = np.dot(dvalues.T, self.inputs)

        self.dbiases = np.sum(dvalues, axis = 0, keepdims = True)

        self.inputs = np.dot(dvalues, self.weights)

class activation_Relu:

    def forward(self, inputs):

        self.inputs = inputs
```

```
        self.output = np.maximum(0,inputs)
    def backward(self, dvalues):
        self.dinputs  = dvalues.copy()   #np.copy(dvalues)
        self.dinputs[self.inputs <= 0] = 0
class activation_softmax:
    def forward(self, inputs):
        self.inputs = inputs
        exp_values = np.exp(inputs-np.max(inputs, axis = 1, keepdims = True))
        self.output = exp_values / np.sum(exp_values, axis =1 , keepdims =True)        #
Probabilties
        def backward(self,dvalues):
        self.dinputs = np.empty_like(dvalues)
            for index, (single_output, single_dvalues) in enumerate (zip(self.output,
dvalues)):
            single_output  = single_output.reshape(-1,1)
            jacobian_matrix = np.diagflat(single_output) - np.dot(single_output,
single_output.T)
            self.dinputs[index] = np.dot(jacobian_matrix, single_dvalues)
class Loss:
    def calculate(self, output, y):
        sample_losses = self.forward(output, y)
        data_loss = np.mean(sample_losses)
        return data_loss
class Loss_categoricalcrossentropy(Loss):
```

```
    def forward(self, y_pred, y_true): # suppose if we have predicted output and ground truth labels

        y_pred_clipped = np.clip(y_pred, np.exp(-7), 1-np.exp(-7)) # Excluding the values less than e-7 and higher that 1=e-7

     def backward(self, dvalues, y_true):

        labels = len(dvalues[0])

        if len(y_true.shape) == 1:

            y_true = np.eye(labels)[y_true]



        self.dinputs = -y_true/dvalues

        self.dinputs = self.dinputs/len(dvalues)

class activation_softmax_loss_categoricalentropy():

    def __init__(self):

        self.activation = activation_softmax()

        self.loss = Loss_categoricalcrossentropy()



    def forward(self, inputs, y_true):

        self.activation.forward(inputs)

        self.output = self.activation.output

        return self.loss.calculate(self.output, y_true)

    def backward(self, dvalues, y_true):

        if len(y_true.shape) == 2:

            y_true = np.argmax(y_true, axis = 1)
```

```
        self.dinputs = dvalues.copy()

        self.dinputs[range(len(dvalues)), y_true] -=1

        self.dinputs = self.dinputs /len(dvalues)

class optimizer_SGD():

    def __init__(self, learning_rate = 1.0):

        self.learning_rate = learning_rate

    def update_params(self, layer):

        layer.weights += -self.learning_rate*layer.dweights

        layer.biases += -self.learning_rate*layer.dbiases

# Creating a object class of Layer_Dense()

layer1 = Layer_Dense(2, 64)

# create a object of class Relu

activation1 = activation_Relu()

# second layer

layer2 = Layer_Dense(64,3)

# pass through activation function

activation1.forward(layer1.output)

# pass through second layer

layer2.forward(activation1.output)

# pass through second activation layer

activation2.forward(layer2.output)

# computing loss for the forward pass

loss = loss_function.calculate(activation2.output, y)
```

```
#loss = loss_activation.forward(layer2.output,y)

# Accuracy calculation

predictions = np.argmax(activation2.output, axis = 1)

if len(y.shape) == 2:

    y = np.argmax(y, axis = 1)

accuracy = np.mean(predictions == y)

print(loss)
```

6.2.2.BACK END(JSON)

```
// Import necessary modules

const express = require('express');

const bodyParser = require('body-parser');

const mongoose = require('mongoose');


// Connect to MongoDB (replace 'your-database-url' with your actual MongoDB connection string)

mongoose.connect('your-database-url', { useNewUrlParser: true, useUnifiedTopology: true });


// Create a MongoDB schema and model for the environmental data

const environmentalDataSchema = new mongoose.Schema({

  pm25: Number,

  temperature: Number,
```

```
  humidity: Number,
  timestamp: { type: Date, default: Date.now }
});

const EnvironmentalData = mongoose.model('EnvironmentalData', environmentalDataSchema);

// Create the Express application
const app = express();

// Middleware to parse JSON in the request body
app.use(bodyParser.json());

// Endpoint to receive environmental data from IoT devices
app.post('/api/environmental-data', (req, res) => {
  // Extract data from the JSON request body
  const { pm25, temperature, humidity } = req.body;

  // Create a new EnvironmentalData instance
  const newEnvironmentalData = new EnvironmentalData({
    pm25,
    temperature,
    humidity
```

```
  });
  // Save the data to the MongoDB database
  newEnvironmentalData.save((err) => {
    if (err) {
      console.error(err);
      res.status(500).send('Internal Server Error');
    } else {
      res.status(201).send('Data received and saved successfully');
    }
  });
});

// Start the server (replace 3000 with your desired port)
const PORT = 3000;
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```

6.2.3 SAMPLE OUTPUT

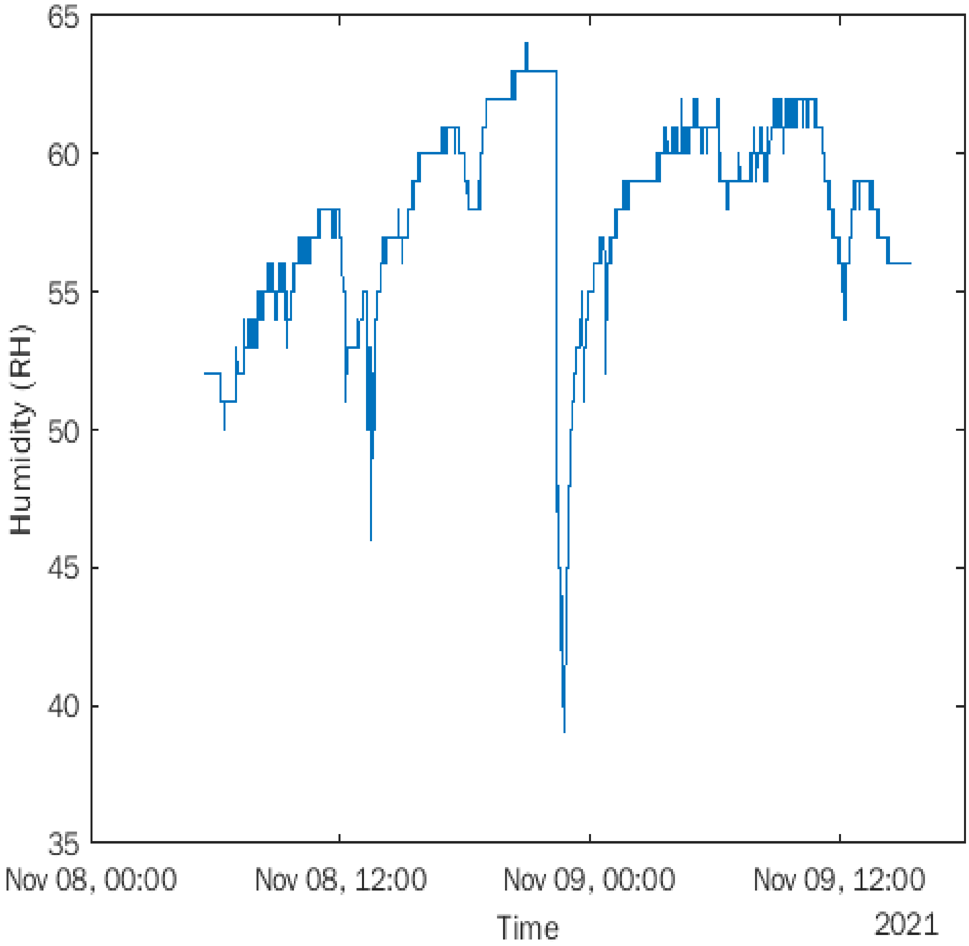
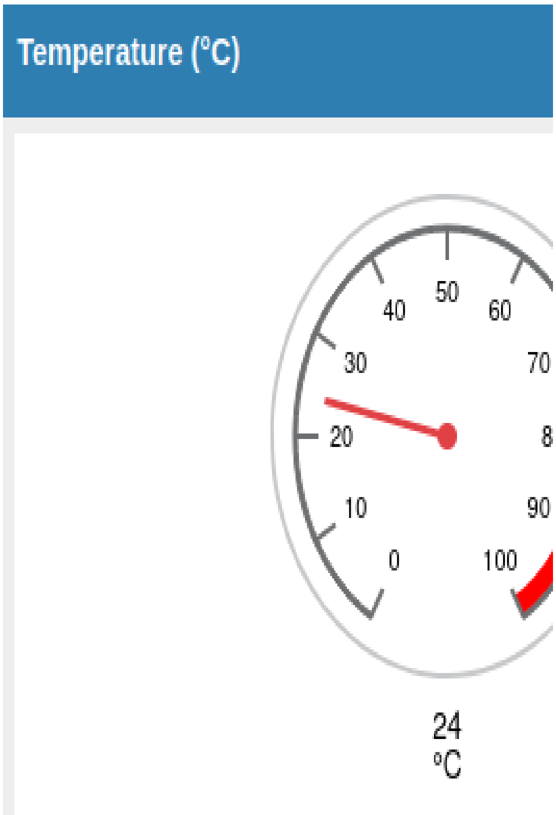


Fig 6.2.3. **Past_humidity_reading**

SAMPLE OUTPUT



CHAPTER 7

7. CONCLUSION AND FUTURE ENHANCEMENT

Essentially, in this paper we focus on the past proposed strategies where we see that there are many examination openings that should be talked about. The paper tends to water wellbeing, air pollution, ecological observation, radiation location, synthetic defilement, normal risk, ranch guideline, squander the executives, etc. In smart Climate, IOT could likewise oversee and break down the ecological stream attributes on the both water, air and anticipated alterations which can trigger any human , creature and plant issues. Moreover, IOT plays a vital job in overseeing ecological harm, regular and non-catastrophic events, just as in controlling vegetation bosses in the climate.

This paper has presented an extensive and critical review of research studies on various environment monitoring systems used for different purposes. The analysis and discussion of the review suggested major recommendations. The need of extensive research on deep learning, handling big data and noisy data issues, and a framework of robust classification approaches has been realized. We have focused mainly on water quality, and air quality monitoring as smart agriculture systems that can deal with environmental challenges. The major challenges in implementation of smart sensors, AI and WSN need to be addressed for sustainable growth through SEM. The participation of environmental organizations, regulator bodies and general awareness would strengthen SEM efforts. The poor quality of sensory data can be preprocessed using appropriate filters and signal processing methods to make the data more suitable for all subsequent tasks associated in SEM. The future scope of the work aims at studying other factors of environment such as sound pollution and disaster etc,

CHAPTER 8

8.REFERENCE

[1] Deekshath MR, Dharanya MP, Kabadia MK, Dinakaran MG, Shanthini S. IoT based environmental monitoring system using Arduino UNO and thingspeak. International Journal of Science Technology & Engineering. 2018;4(9).

[2] Behera TM, Mohapatra SK, Samal UC, Khan MS, Daneshmand M, Gandomi AH. I-SEP: An improved routing protocol for heterogeneous WSN for IOT-based environmental monitoring. IEEE Internet of Things Journal. 2019 Sep 12;7(1):710-7.

[3] Sunny AI, Zhao A, Li L, KantehSakiliba S. Low-Cost IoT-Based Sensor System: A Case Study on Harsh Environmental Monitoring. Sensors. 2021 Jan;21(1):214.

[4] Abraham S, Beard J, Manijacob R. Remote environmental monitoring using internet of things (iot). In2017 IEEE Global Humanitarian Technology Conference (GHTC) 2017 Oct 19 (pp. 1-6). IEEE.

[5] Cabra J, Castro D, Colorado J, Mendez D, Trujillo L. An IoT approach for wireless sensor networks applied to e-health environmental monitoring. In2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) 2017 Jun 21 (pp. 578-583). IEEE.

[6] Khan N, Khattak KS, Ullah S, Khan Z. A low-cost IoT based system for environmental monitoring. In2019 International Conference on Frontiers of Information Technology (FIT) 2019 Dec 16 (pp. 173-1735). IEEE.

[6] Shah J, Mishra B. IoT enabled environmental monitoring system for smart cities. In2016 international conference on internet of things and applications (IOTA) 2016 Jan 22 (pp. 383-388). IEEE.

[7] Lazarescu MT. Design of a WSN platform for long-term environmental monitoring for IoT applications. IEEE Journal on emerging and selected topics in circuits and systems. 2013 Mar 7;3(1):45-54.

[8] Mois G, Folea S, Sanislav T. Analysis of three IoT-based wireless sensors for environmental monitoring. IEEE Transactions on Instrumentation and Measurement. 2017 Mar 27;66(8):2056-64.

[9] Jaladi AR, Khithani K, Pawar P, Malvi K, Sahoo G. Environmental monitoring using wireless sensor networks (WSN) based on IOT. Int. Res. J. Eng. Technol. 2017 Jan;4(1):1371-8.