CODE

#include SoftwareSerial.h

include the library code

#include LiquidCrystal.h

initialize the library with the numbers of the interface pins

LiquidCrystal lcd(13, 12, 6, 5, 4, 3); LCD connections

D13 == RS

GND == RW

D12 == Enable

D6 == DB4

D5 == DB5

D4 == DB6

D3 == DB7

float t=0;

char data = 0;

String apiKey = XBQDVORXXGAROWDW; Write API key

connect 8 to TX of ESP

connect 9 to RX of ESP

SoftwareSerial ser(8,9); RX, TX

void setup()

{

enable debug serial

Serial.begin(9600); serial data transmission at Baudrate of 9600

enable software serial

ser.begin(9600);

lcd.begin(16, 2); to intialize LCD

lcd.setCursor(0,0);

lcd.print( Welcome);

lcd.setCursor(0,1);

lcd.print( To );

delay(3000);

lcd.clear();

lcd.setCursor(0,0);

lcd.print( AIR);

lcd.setCursor(0,1);

lcd.print(QUALITY MONITOR);

delay(3000);

ser.println(AT); Attenuation

delay(1000);

ser.println(AT+GMR); To view version info for ESP-01 output 00160901

and ESP-12 output 0018000902-AI03

delay(1000);

ser.println(AT+CWMODE=3); To determine WiFi mode

1 = Station mode (client)

2 = AP mode (host)

3 = AP + Station mode (ESP8266 has a dual mode)

delay(1000);

ser.println(AT+RST); To restart the module

delay(5000);

ser.println(AT+CIPMUX=1); Enable multiple connections

0 Single connection

1 Multiple connections (MAX 4)

delay(1000);

String cmd=AT+CWJAP=SSID,PASSWORD; connect to Wi-Fi

ser.println(cmd);

delay(1000);

ser.println(AT+CIFSR); Return or get the local IP address

delay(1000);

lcd.clear();

lcd.setCursor(0,0);

lcd.print( WIFI);

lcd.setCursor(0,1);

lcd.print( CONNECTED);

}

void loop()

{

delay(1000);

t = analogRead(A0); Read sensor value and stores in a variable t

Serial.print(Airquality = );

Serial.println(t);

lcd.clear();

lcd.setCursor (0, 0);

lcd.print (Air Qual );

lcd.print (t);

lcd.print ( PPM );

lcd.setCursor (0,1);

if (t=500)

{

lcd.print(Fresh Air);

Serial.print(Fresh Air );

}

else if( t=500 && t=1000 )

{

lcd.print(Poor Air);

Serial.print(Poor Air);

}

else if (t=1000 )

{

lcd.print(Very Poor);

Serial.print(Very Poor);

}

lcd.scrollDisplayLeft();

delay(10000);

lcd.clear();

lcd.setCursor(0,0);

lcd.print( SENDING DATA);

lcd.setCursor(0,1);

lcd.print( TO CLOUD);

esp\_8266();

}

void esp\_8266()

{

TCP connection AT+CIPSTART=4,TCP,184.106.153.149,80

String cmd = nAT+CIPSTART=4,TCP,; Establish TCP connection

AT+CIPSTART=id,type,addr,port

id 0-4, id of connection

type String, “TCP” or “UDP”

addr String, remote IP

port String, remote port

cmd += 184.106.153.149; api.thingspeak.com

cmd += ,80;

ser.println(cmd);

Serial.println(cmd);

if(ser.find(Error))

{

Serial.println(AT+CIPSTART error);

return;

}

String getStr = GET updateapi\_key=; API key

getStr += apiKey;

getStr +=&field1=;

getStr +=String(h);

getStr +=&field1=;

getStr +=String(t);

getStr += rnrn;

send data length

cmd = AT+CIPSEND=; Send data AT+CIPSEND=id,length

cmd += String(getStr.length());

ser.println(cmd);

Serial.println(cmd);

delay(1000);

ser.print(getStr);

Serial.println(getStr);

thingspeak needs 16 sec delay between updates

delay(17000);

}

<!DOCTYPE html>

<html>

<head>

<title>Air Quality Monitoring</title>

<style>

/\* Add some basic CSS for styling \*/

Body {

Font-family: Arial, sans-serif;

Text-align: center;

}

#air-quality {

Font-size: 24px;

}

</style>

</head>

<body>

<h1>Air Quality Monitoring</h1>

<div id=”air-quality”>

<!—Air quality data will be displayed here 

</div>

<script>

// Function to fetch and update air quality data (replace with actual data source).

Function fetchAirQuality() {

// Replace the following with your actual air quality data.

Const airQualityData = {

Location: “City Name”,

Quality: “Good”, // Replace with the actual air quality measurement.

};

Const airQualityElement = document.getElementById(“air-quality”);

airQualityElement.innerHTML = `Air Quality in ${airQualityData.location}:

${airQualityData.quality}`;

}

// Call the function to update air quality when the page loads.

Window.onload = fetchAirQuality;

</script>

</body>

</html>

**AIR QUALITY MONITORING**

**PROBLEM STATEMENT**:

 Air quality monitoring is a critical environmental concern due to the

significant impact of air pollution on human health, ecosystems, and

climate change. The problem encompasses various aspects, including

measuring air quality, analyzing the data, and taking appropriate

actions. The following problem statement outlines key challenges

and potential solutions.

**CHALLENGES:**

**Collection and Monitoring:** Gathering real-time, accurate, and comprehensive air

quality data across various locations is a challenge. Traditional monitoring stations are limited in

number and coverage.

**Accuracy and Calibration:** Ensuring the accuracy and calibration of air quality

monitoring devices is essential to produce reliable data for analysis and decision-making.

**Data Integration:** Integrating data from multiple sources and types of sensors (e.g.,

particulate matter, gases, meteorological data) into a coherent dataset can be complex.

**Data Quality and Preprocessing:** Raw sensor data often contains noise, outliers, and

missing values that need to be cleaned and preprocessed to obtain meaningful insights.

**Data Analysis:** Analyzing air quality data to identify trends, pollution sources, and health

risks requires advanced statistical and machine learning techniques.

**Public Awareness:** Educating and raising public awareness about the importance of air

quality and its health implications is a challenge.

**Regulatory Compliance:** Ensuring compliance with air quality regulations and standards,

as well as enforcing them, can be resource-intensive.

**SOLUTIONS:**

 Sensor Networks: Deploy a network of low-cost air quality sensors across urban

areas to increase data coverage. Crowdsourced data from citizen science projects can

supplement official monitoring.

**Sensor Calibration:** Regularly calibrate and maintain monitoring sensors to ensure

data accuracy. Automated calibration processes can help in this regard.

**Data Integration Platforms:** Develop centralized platforms that can collect,

process, and integrate data from various sources, including government agencies, IoT

sensors, and satellites.

**Data Quality Assurance:** Implement data quality control procedures to clean and

preprocess raw data, including outlier detection, imputation, and validation.

**Advanced Analytics:** Utilize machine learning algorithms for predictive modeling,

source apportionment, and health risk assessment based on air quality data.

**Public Engagement:** Launch public awareness campaigns, mobile apps, and

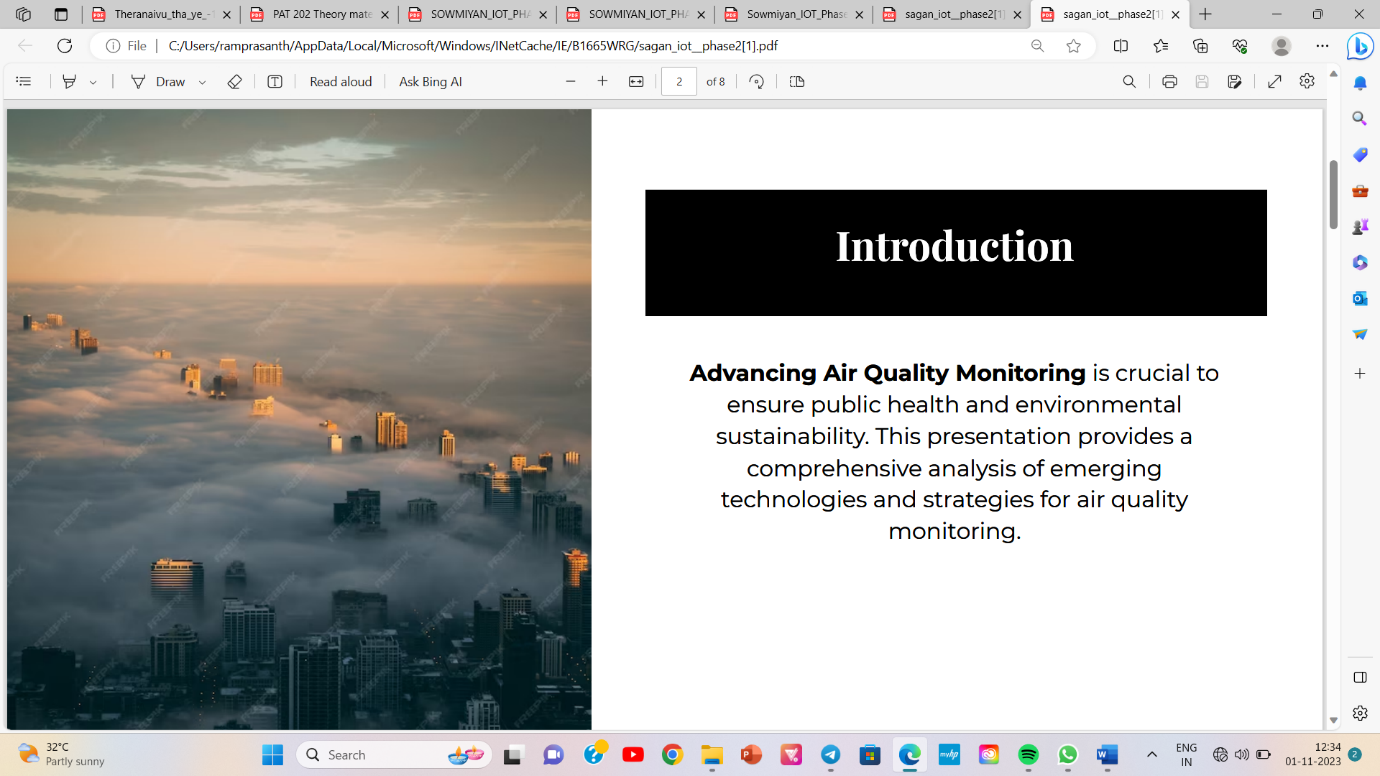
websites that provide real-time air quality information, health recommendations, and

pollution source tracking

**Advancing Air Quality Monitoring: A Comprehensive Analysis of Emerging Technologies and Strategies**

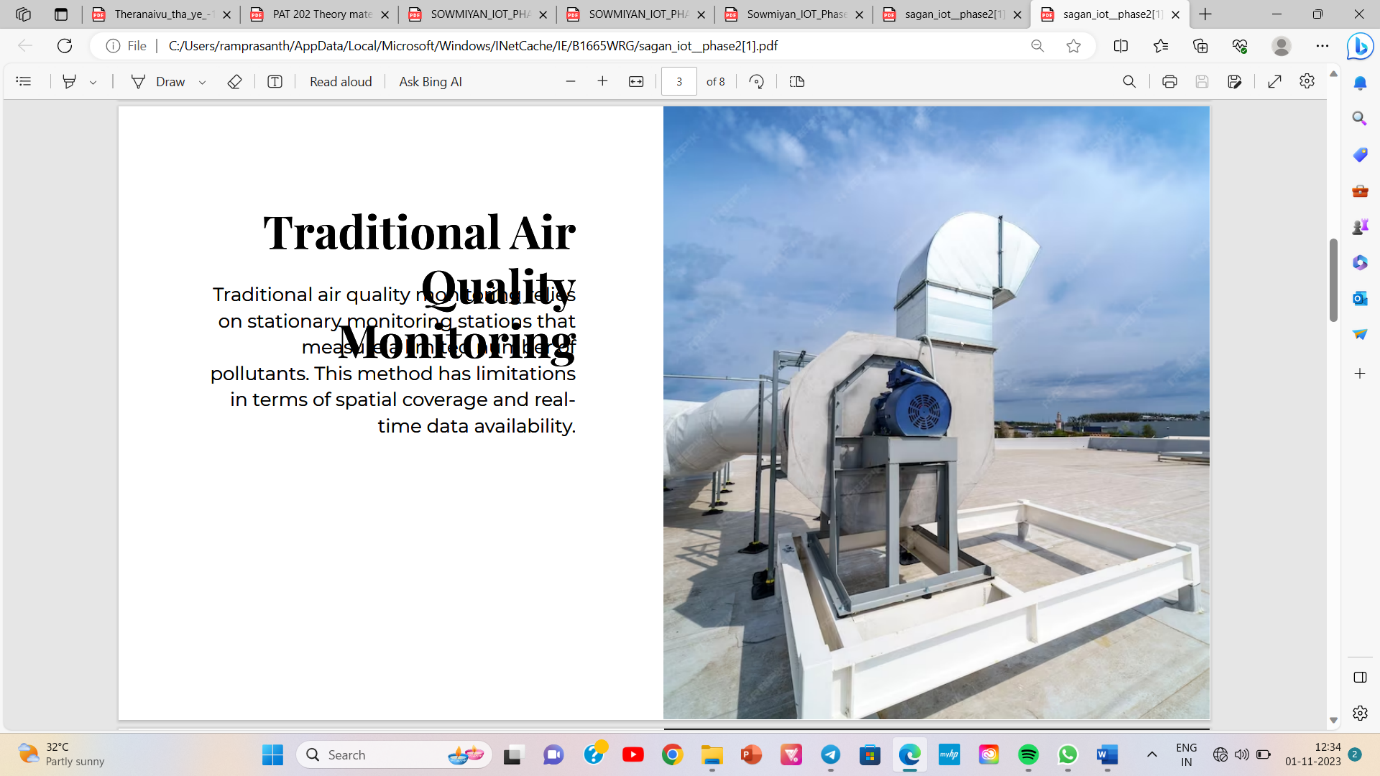
**Introduction**

Advancing Air Quality Monitoring is crucial to ensure public health and environmental sustainability. This presentation provides a comprehensive analysis of emerging technologies and strategies for air quality monitoring.



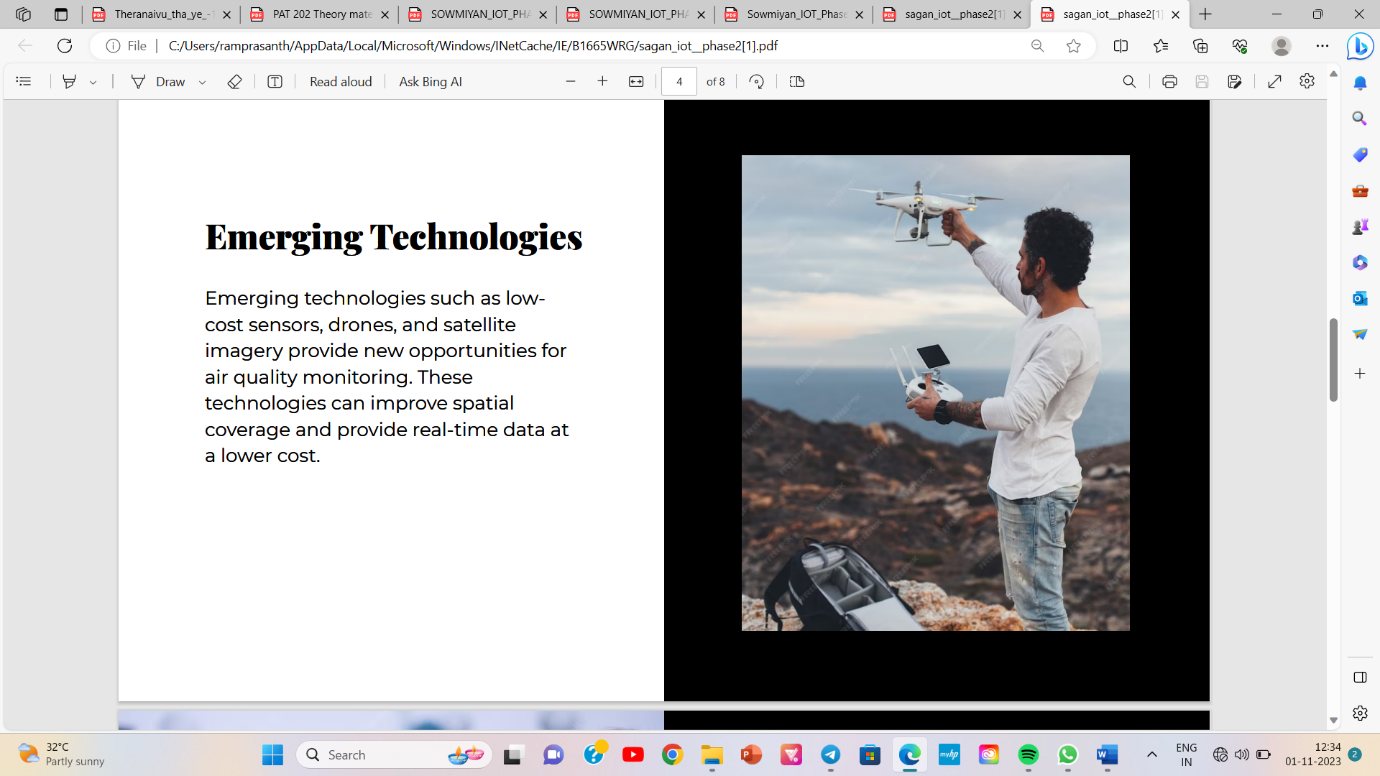
**Traditional Air Quality Monitoring**

Traditional Air Quality Monitoring Traditional air quality monitoring relies on stationary monitoring stations that measure a limited number of pollutants. This method has limitations in terms of spatial coverage and real time data availability.



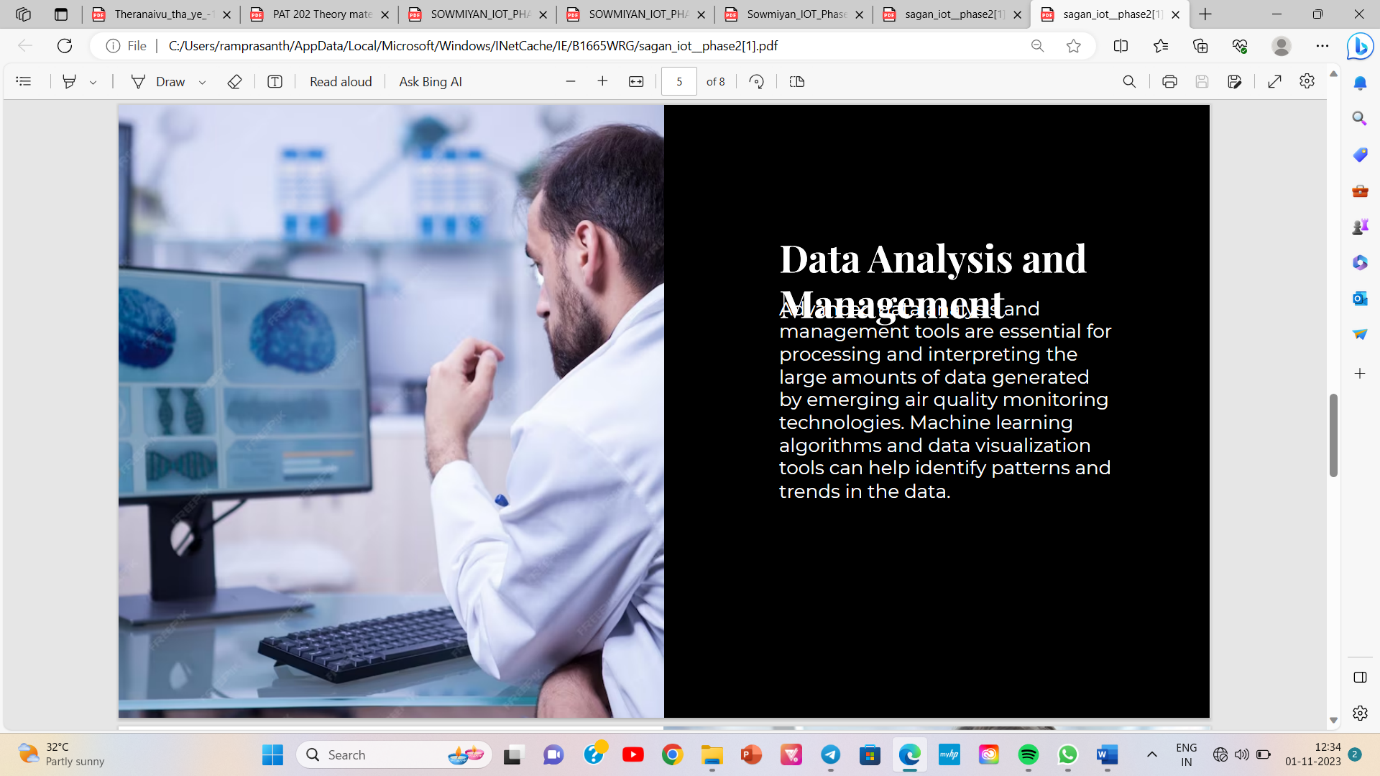
**Emerging Technologies**

Emerging technologies such as lowcost sensors, drones, and satellite imagery provide new opportunities for air quality monitoring. These technologies can improve spatial coverage and provide real-time data at a lower cost.



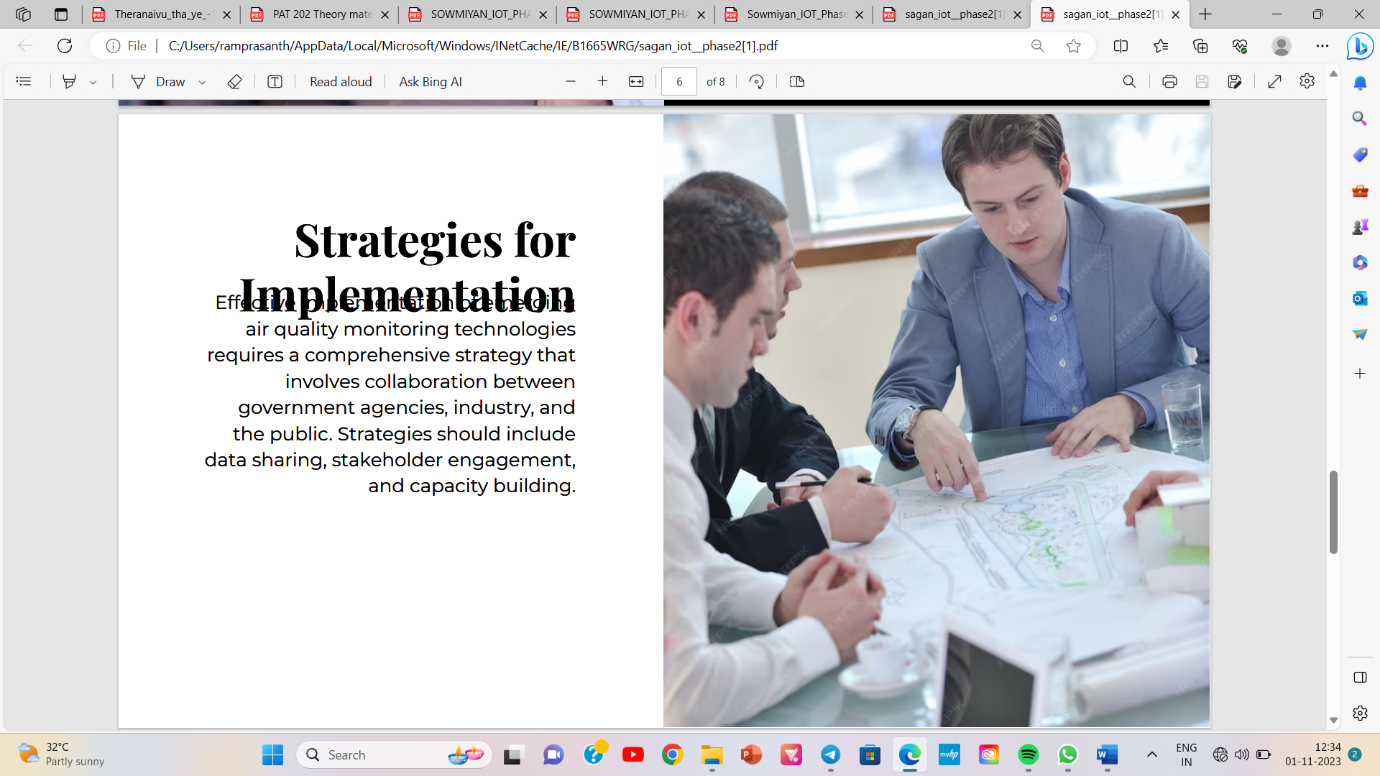
**Data Analysis and Management**

Data Analysis and Advanced data analysisand management tools are essential for processing and interpreting the large amounts of data generated by emerging air quality monitoring technologies. Machine learning algorithms and data visualization tools can help identify patterns and trends in the data.



**Strategies for Implementation**

Strategies for Effective implementation of emerging air quality monitoring technologies requires a comprehensive strategy that involves collaboration between government agencies, industry, and the public. Strategies should include data sharing, stakeholder engagement, and capacity building.



**Conclusion**

Advancing air quality monitoring through the use of emerging technologies and strategies is essential for protecting public health and the environment. With the right tools and strategies in place, we can create a more sustainable future f or all. 