Journal of Ubiquitous Computing and Communication Technologies (ISSN: 2582-337X)
www.irojournals.com/jucct/

Safety Monitoring and Auto Alert System in Underground Collieries

Suguna A.[1], Achiammal B.[2], Ayesha Mariyam A.[3], Iswarya S.[4]

Assistant ProfessoR, Department of Electronics and Instrumentation Engineering, Coimbatore, India

E-mail: [1]asuguna@gct.ac.in, [2]nannan@gct.ac.in, [3]ayeshamariyam2002@gmail.com, [4]ishusrinivasan11@gmail.com

Abstract

In the current scenario, significant mine disasters often occur due to explosions and fires. The occurrence of unexpected underground mine landslides poses a significant threat to miner safety, leading to fatalities. Consequently, the proposed system actively monitors key parameters and issues alerts to miners, aiming to mitigate this risk. The system primarily focuses on monitoring gas levels, air quality, temperature, and vibrations in underground mines, with the data being continuously recorded and stored in the ThingSpeak cloud. Visualization of the data is achieved through field charts created within the designated channel. The cloud-stored data is linked with MathWorks to predict missing values, providing a basis for proactive measures. The predicted missing values are then stored in a separate channel on ThingSpeak for continuous chart visualization. This data, complete with timestamps, can be exported to excel for further analysis and application. The scenario of the coal mine is monitored and regular reports are sent to the personnel's mail which is automated in Thingspeak. In the event of an explosion, the rescue team is promptly notified through SMS alerts using a GSM modem.

Keywords: Mines, ThingSpeak, visualization, Mathworks, Regular Reports, GSM alerts

1. Introduction

The occurrences of coal mine disasters are typically attributed to the intricate nature of the mine environment and the diverse working conditions within coal mines. Consequently,

DOI: https://doi.org/10.36548/jucct.2024.2.007
Received: 10.04.2024, received in revised form: 13.05.2024, accepted: 27.05.2024, published: 06.06.2024

it is imperative to continually monitor the working environment of mines[1]. However, the current safety production standards in coal mines remain suboptimal, particularly in recent years when coal mine disasters have become more frequent. These incidents result in significant losses of property and lives, posing substantial threats to the country's economic and social stability[2,3]. The safety challenges in coal mines have progressively become a focal point of national and societal concern. Implementing a monitoring and alarm system for coal mine safety emerges as an effective approach to prevent or alleviate such losses. Conventional monitoring systems in coal mines typically rely on wired networks, playing a crucial role in ensuring safe production[4]. However, as coal mining areas expand and reach greater depths, numerous laneways become inaccessible to monitoring, harbouring potential risks. Additionally, these wired systems incur high costs and involve complex design, construction, and maintenance processes. In response to these concerns, a coal mine safety monitoring system has been created utilizing wireless sensor networks. This system aims to enhance the effectiveness of safety monitoring in production and minimize accidents within the coal mine, offering a more efficient and cost effective solution[5-7].

2. Existing and Proposed System

A. Existing System

Current coal mine safety systems are like having a network of watchful eyes underground. These eyes come in the form of sensors that constantly check the air for dangerous gases like methane and dust, how hot and humid it is, how stable the ground is, and if there's enough fresh air circulating. All this information is sent wirelessly to a central hub, like a mission control center. Here, mine managers can see everything that's going on in real-time. If something gets risky, like too much gas building up or the ground starting to move, alarms go off to warn miners. Alerts can also be sent automatically by email or text message to key people on the surface. The main goal of this system is to keep miners safe by giving them an early warning of trouble. It also helps the mine run smoother by giving managers better information to make decisions about things like ventilation and where to send equipment[8-10].

B. Proposed System

The proposed system actively monitors crucial parameters and sends alerts to miners, aiming to reduce this risk. The system primarily concentrates on overseeing gas levels, air quality, temperature, and vibrations in underground mines. The collected data is continuously stored in the ThingSpeak cloud and visualized through field charts in the designated channel. To enhance the data, it is linked with MathWorks to predict missing values, forming a basis for proactive measures. The predicted missing values are stored separately on ThingSpeak for ongoing chart visualization. This comprehensive data, accompanied by timestamps, can be exported to excel for further analysis and application. Automated email alerts are generated when specific values surpass predetermined thresholds, providing an early warning to personnel. In case of an explosion, the rescue team is promptly notified through SMS alerts using a GSM modem.

3. Block Diagram

A. Block Diagram of Data Acqisition

The MQ5 sensor is employed to identify the concentration of carbon dioxide in the air and is linked to the analog pin (A0) of the ESP8266. The D2 pin of Nodemcu is connected to a relay, which is powered by a 12V DC supply from a battery. The NC pin of the relay is linked to the draft fan, which activates when the $CO2$ level surpasses a predefined threshold. To monitor temperature and humidity in underground collieries, the DHT11 sensor is connected to the D5 pin of ESP8266. The vibration sensor is attached to the digital pin (D4) of the Nodemcu. Data collected from these sensors is stored in the cloud. A MATLAB code is then utilized to predict missing data using information from the cloud. MATLAB is also employed to dispatch daily reports to personnel. The Fig 3.1 shows the block diagram of the data acquisition.

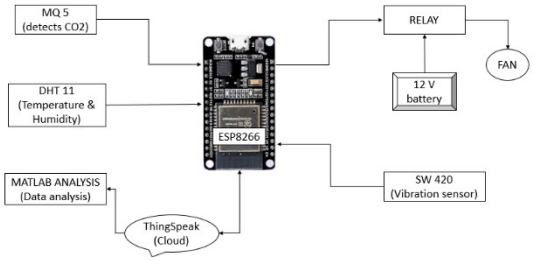

Figure 3.1. Block Diagram for Data Acquisition

B. Block Diagram of Alert System

The GSM module is integrated separately with the Arduino UNO microcontroller due to its requirement of a 5V signal, while the ESP8266 operates on a 3.3V signal. This necessitates connecting the GSM module to the Arduino UNO. Through the utilization of AT commands, messages and calls can be transmitted to designated numbers upon the fulfillment of specific conditions. The program is designed to prompt the GSM module to dispatch automatic alerts, such as SMS notifications and phone calls, to predetermined contacts when the $CO2$ and vibration readings exceed predetermined thresholds. This ensures that the rescue team receives timely notifications through SMS and calls facilitated by the GSM module. Fig 3.2 illustrates the block diagram for alert system.

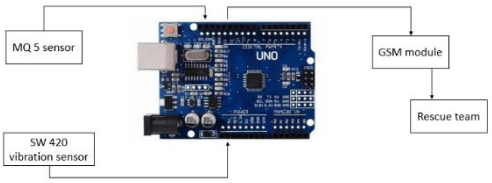

Figure 3.2. Block Diagram for Alert System

4. Description of the Components

A. MQ-5 Sensor

The MQ-5 gas sensor exhibits high sensitivity to butane, propane, and methane, capable of detecting both methane and propane simultaneously. Furthermore, it can identify various flammable gases, with a particular emphasis on liquefied petroleum gas (LPG), such as propane. Due to its affordability, it serves as a versatile and cost-effective sensor suitable for numerous applications.

B. DHT 11 Sensor

The DHT11 is an basic, highly affordable digital temperature and humidity sensor. Utilizing a capacitive humidity sensor and a thermistor, it gauges the ambient air conditions and transmits a digital signal through the data pin (without the necessity of analog input pins). While straightforward in operation, it necessitates precise timing for data retrieval

C. SW 420 Vibration Sensor

The Vibration Sensor (SW-420) is a non-directional sensor renowned for its high sensitivity to vibrations. In stable conditions, the circuit remains active with a high output. However, upon detecting movement or vibration, the circuit momentarily disengages, resulting in a low output. Additionally, users have the flexibility to adjust the sensor's sensitivity to suit specific requirements.

D. Relay Module

A relay is a fundamental electromechanical switch that functions differently from regular switches. Whereas typical switches manually control circuit connections, a relay serves to either connect or disconnect two circuits. In the case of a DC relay switch, it utilizes a Direct Current (DC) source to activate an electromagnetic coil within the relay.

D. DC Draft Fan

Draft fans are employed in systems to extract and expel flue gases from combustion chambers by generating a negative air pressure vacuum, typically around -10 mm Hg.

E. Rechargeable Battery

A rechargeable battery is an energy storage unit capable of being replenished after discharge by applying DC current to its terminals. This project utilizes lithium-ion batteries for power.

F. GSM Module SIM900A

A GSM module refers to a chip or circuit utilized to facilitate communication between a mobile device or computing machine and a GSM system.Nodemcu

The NodeMCU (Node MicroController Unit) constitutes an open-source software and hardware development platform centered on a cost-effective System-on-a-Chip (SoC) known as the ESP8266.

H. Arduino UNO

The Arduino Uno is an open-source microcontroller board that utilizes the Microchip technology and is compatible with various expansion boards and circuits.

5. Software Implementation

The Arduino IDE (Integrated Development Environment) is a software application that serves as the primary platform for programming and developing applications for Arduino microcontrollers. It provides a user friendly interface for writing, compiling, and uploading code to Arduino boards. The IDE supports the Arduino programming language, which is a simplified version of C and C++ with specific libraries and functions tailored for Arduino hardware. The Arduino IDE platform is used to program the ESP8266 for data acquisition and Arduino UNO for alerts.

6. ThingSpeak Cloud

A. Channel Creation

In ThingSpeak, a "channel" refers to a virtual container or storage space where data from Internet of Things (IoT) devices is collected, organized, and stored. It is a fundamental concept in ThingSpeak that allows users to manage and analyze data in a structured manner.

- Percentage complete: Calculated based on data entered into the various fields of a channel.
- Channel Name: A unique name for the ThingSpeak channel.
- Description: Description about the project to be fed to the channel.
- Field: It depicts the parameters to be visualized

B. Programming ThingSpeak

To make the code work, the network credentials should be entered.

constchar*ssid="SSID";

constchar*password=" PASSWORD";

Create a Wi-Fi client to connect to ThingSpeak.

WiFiClient client;

The number of channels as well as API key should be entered in the below code which can be viewed on the Private View tab.

unsigned long myChannelNumber = 1;

constchar*myWriteAPIKey="XXXXXXXXXXXXXXXX;

Initialize ThingSpeak.

ThingSpeak.begin(client);

Finally, writing to ThingSpeak can be achieved by,

Intx=ThingSpeak.writeField(myChannelNumber, fieldnumber, data, myWriteAPI key)

C. Data Acquisition

The information gathered from sensors is stored in designated field numbers according to the program's specifications. Various parameters can be allocated to distinct fields, and the stored data can be presented visually through columns, graphs, lines, and other formats. The visual representations in field charts act as a graphical interface for operators in the control room

7. MATLAB Analysis

A. MATLAB Code to Predict Missing Values

The code to predict missing values of CO2 and vibration is shown below

myReadChannel = 2434123;

myWriteChannel = 2467356;

readAPIKey = 'SASY2CAC8PTEABF2';

writeAPIKey = '4WCIOXW7AALJM4DE';

outData = thingSpeakRead(myReadChannel, 'ReadKey', readAPIKey, 'fields',1,'fields',4,...

```
'NumPoints', 30, 'OutputFormat', 'TimeTable');
outData.(1) = fillmissing(outData.(1),'linear');
myNewTData=timetable(outData.Timestamps,outData.(1));
response=thingSpeakWrite(myWriteChannel, myNewTData, 'Writekey', writeAPIKey
```

B. MATLAB Code to Send Regular Reports

The code to send regular reports using React in Thingspeak is shown below,

```
channelID=2467356;
alertApiKey='TAKPQjGCW9CxIyw9zPV';
alertUrl="https://api.thingspeak.com/alerts/send";
options = weboptions("HeaderFields", ["ThingSpeak-Alerts-API-Key", alertApiKey
]);
alertSubject = sprintf("CO2 data");
CO2Data=thingSpeakRead(channelID,'NumDays',30,'Fields',1);
if ( CO2Data < 30) alertBody = ' CO2 level is correct . ';
else alertBody = ' CO2 level is high . ';
end
try
webwrite(alertUrl , "body", alertBody, "subject", alertSubject, options);
catch
someException fprintf("Failed to send alert: %s\n", someException.message);
end
```

C. React to Send Regular Report

The react to send regular report is illustrated in Fig 7.1

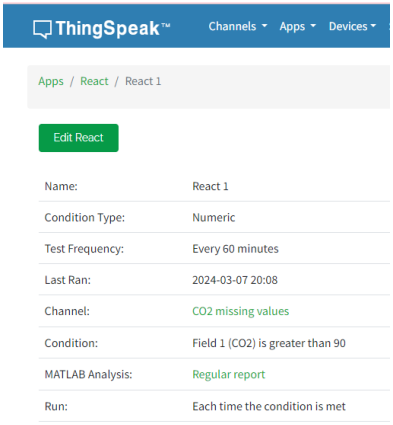

Figure 7.1. React to Send Regular Report

8. Results and Discussions

A. Hardware Implementation

The sensors are being integrated to ESP8266 and powered using USB. The relay is powered by a 12V Li-ion battery. The GSM is being interfaced with Arduino UNO which is powered using 12V adaptor from the main supply. The results observed are illustrated in figures 8.1 to 8.12

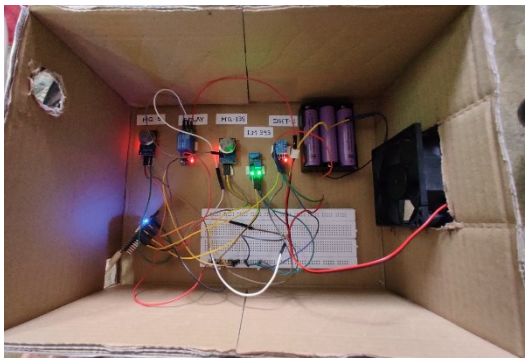

Figure 8.1. Hardware Implementation

B. Serial Monitor in Arduino IDE

The code is fed to the NODEMCU using Arduino IDE for acquiring the data from the sensors. The sensors are then interfaced with NODEMCU and the acquired data is displayed in the serial monitor of Arduino IDE.

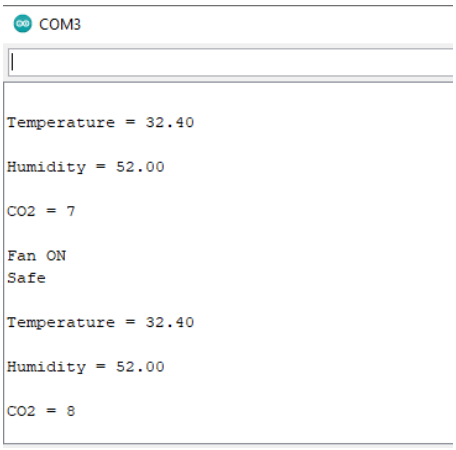

Figure 8.2. Sensor's Output

C. GSM Output

The program is uploaded to the NODEMCU via the Arduino IDE to gather data from MQ-5 and SW 420 sensors. These sensors are connected to the Arduino UNO, and the acquired data is shown in the serial monitor of the Arduino IDE.

Figure 8.3. GSM Output

D. Acquired CO2 data in ThingSpeak

The data being acquired from the MQ-5 sensor is pushed to the cloud using ESP8266 Wifi module and is being visualized in field chart 1 of the channel created as shown in Fig 8.4.

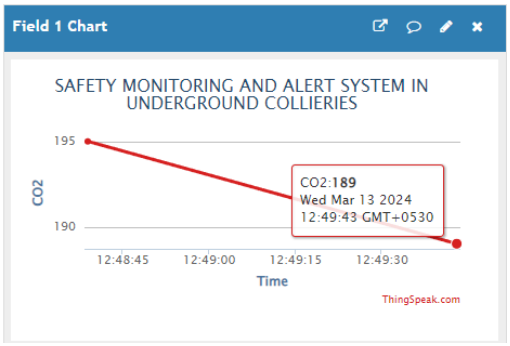

Figure 8.4.CO2 Data in ThingSpeak

E. Acquired Temperature Data in ThingSpeak

The data being acquired from the DHT 11 sensor is pushed to the cloud using ESP8266 Wifi module and is being visualized in field chart 3 of the channel created as shown in Fig 8.5.

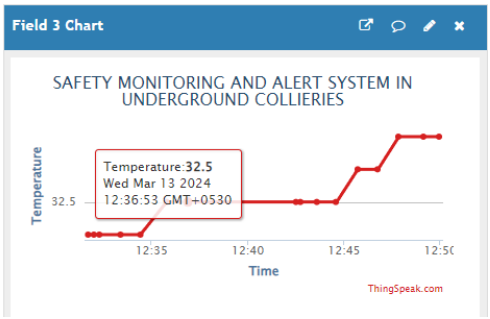

Figure 8.5. Temperature Data in ThingSpeak

F. Acquired Humidity Data in ThingSpeak

The data being acquired from the DHT 11 sensor is pushed to the cloud using ESP8266 Wifi module and is being visualized in field chart 4 of the channel created as shown in Fig 8.6.

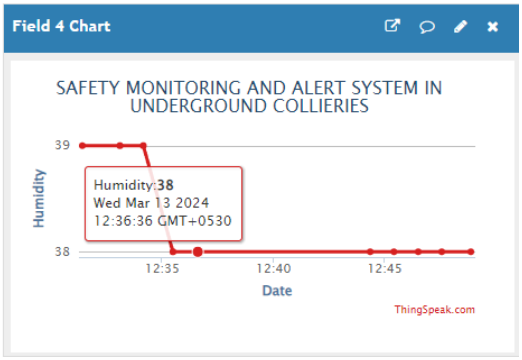

Figure 8.6. Humidity data in ThingSpeak

G. Acquired Vibration Data in ThingSpeak

The data being acquired from the SW 420 sensor is pushed to the cloud using ESP8266 Wifi module and is being visualized in field chart 5 of the channel created as shown in Fig 8.7.

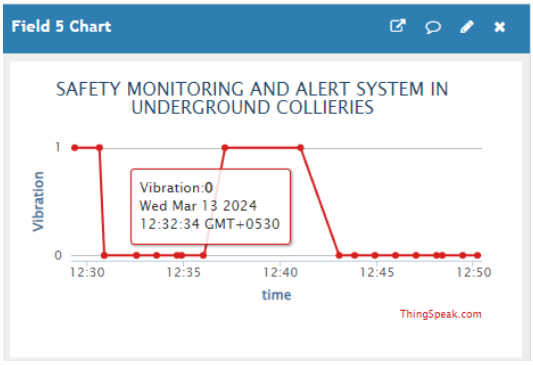

Figure 8.7. Vibration Data in ThingSpeak

H. Predicted CO2 Data with Missing Values

A MATLAB code is developed to analyze the $CO2$ data and forecast missing data points. The projected data is stored in a separate channel and displayed in Field Chart 1 of that channel as shown in Fig 8.8.

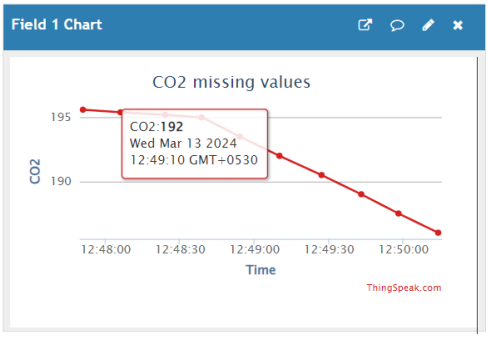

Figure 8.8. Predicted CO2 Data in ThingSpeak

I. Predicted Vibration Data with Missing Values

A MATLAB code is developed to analyze the vibration data and forecast missing data points. The projected data is stored in a separate channel and displayed in Field Chart 1 of that channel as shown in Fig 8.9.

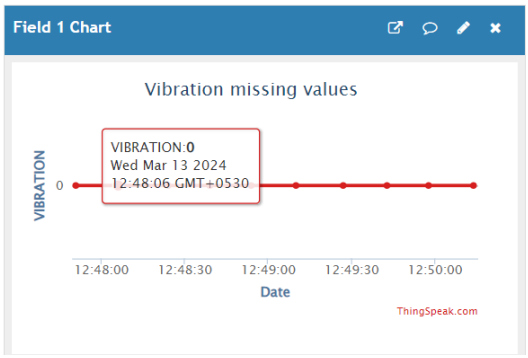

Figure 8.9. Predicted Vibration Data in ThingSpeak

J. Predicted Vibration Data with Missing Values

The React function in ThingSpeak is automated to send periodic updates about the situation via email to the designated personnel every 60 minutes. The mail being sent is shown in Fig 8.10.

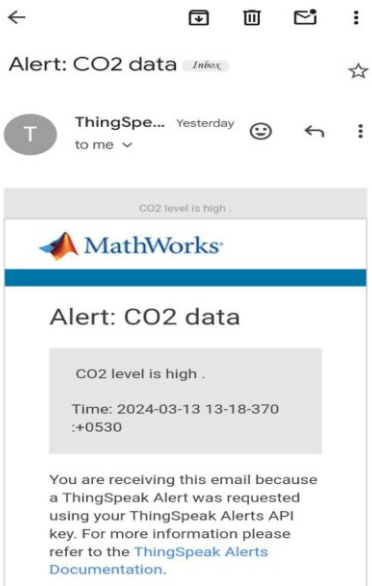

Figure 8.10. Regular Reports in Mail

K. SMS Alert through GSM

The AT commands in GSM are used to send SMS alerts to the rescue team when the $CO2$ and vibration levels surpass a predetermined threshold. The content of the SMS sent to the rescue team is displayed in Figure 8.11.

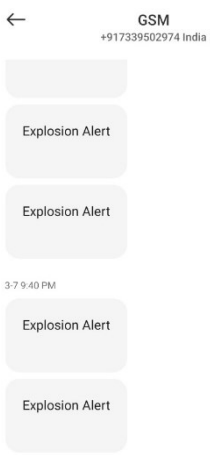

Figure 8.11. SMS Alert through GSM

L. Call Alert through GSM

The AT commands in GSM are used to send call alerts to the rescue team when the $CO2$ and vibration levels surpass a predetermined threshold. The call sent to the rescue team is displayed in Figure 8.12.

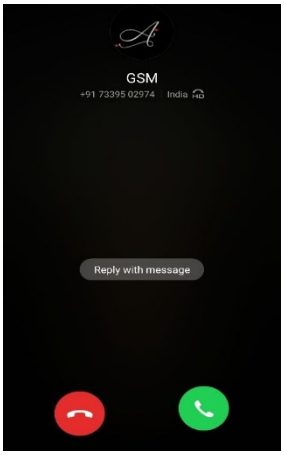

Figure 8.12. Call Alert through GSM

9. Conclusion

Thus, the proposed system actively monitors critical parameters and alerts miners to mitigate risks. It primarily focuses on supervising gas levels, air quality, temperature, and vibrations in underground mines. Collected data is continuously stored in the ThingSpeak cloud and displayed through field charts in a dedicated channel. To improve data quality, it's integrated with MathWorks to predict missing values, enabling proactive measures. The

projected missing values are stored separately on ThingSpeak for ongoing chart visualization. This comprehensive data, along with timestamps, can be exported to excel for further analysis and use. Automated email alerts are triggered when specific values exceed predefined thresholds, giving early warnings to personnel. In the event of an explosion, the rescue team is promptly notified via SMS alerts using a GSM modem.

References

[1] Chehri, Abdellah, Wissam Farjow, Hussein T. Mouftah, and Xavier Fernando. "Design of wireless sensor network for mine safety monitoring." In 2011 24th Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 001532-001535. IEEE, 2011.

[2] Cheng, Jie, Dian Wu Gao, Jun Wang and Dongge Wen. “Coal Mine Safety Monitoring System Based on ZigBee and GPRS.” Applied Mechanics and Materials 422 (2013): 215 - 220.

[3] Ansari, A. H., Karishma Shaikh, Pooja Kadu, and Nikam Rishikesh. "IOT based coal mine safety monitoring and alerting system." Int. J. Sci. Res. Sci. Eng. Technol 8 (2021): 404-410.

[4] Nettikadan, David, and Subodh Raj. "Smart community monitoring system using thingspeak iot plaform." International Journal of Applied Engineering Research 13, no. 17 (2018): 13402-13408.

[5] Choudhary, Suresh Kumar, Shubham Dadsena, and Saiprasad Balraj. "Gas Leakage Detector using Arduino and GSM Module with SMS Alert and Sound Alarm." Available at SSRN 3917754 (2019).”.

[6] Karna, Suwarna, Tanzila Noushin, and Shawana Tabassum. "IoT based Smart Helmet for Automated and Multi-parametric Monitoring of Underground Miners’ Health Hazards." In 2022 IEEE 15th Dallas Circuit And System Conference (DCAS), pp. 1-2. IEEE, 2022.

[7] Matloob, Sundas, Yang Li, and Khurram Zaman Khan. "Safety measurements and risk assessment of coal mining industry using artificial intelligence and machine learning." Open journal of business and management 9, no. 3 (2021): 1198-1209.

[8] Chen, Wei, and Xuzhou Wang. "Coal mine safety intelligent monitoring based on wireless sensor network." IEEE sensors journal 21, no. 22 (2020): 25465-25471.

[9] Porselvi, T., Sai Ganesh, B. Janaki, and K. Priyadarshini. "IoT based coal mine safety and health monitoring system using LoRaWAN." In 2021 3rd International Conference on Signal Processing and Communication (ICPSC), pp. 49-53. IEEE, 2021.

[10] Li, Mo, and Yunhao Liu. "Underground coal mine monitoring with wireless sensor networks." *ACM Transactions on Sensor Networks*