iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: V Month of publication: May 2026

DOI: https://doi.org/10.22214/ijraset.2026.82132

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

IoT Based Real-Time Air Quality Monitoring and Prediction Using Machine Learning Approach

Mrs. Chinnakka Sudha[1], Suragani Revathi[2], Patloori Durga[3]
Department of Information Technology, Mahatma Gandhi Institute of Technology, Hyderabad, Telangana

Abstract: Air pollution is a problem. It is getting worse and worse. This is bad for the earth. It is also bad for the health of people. Air pollution is now an issue that the whole world is worried, about. Air pollution is something that we should all be concerned about. We currently check air quality using methods. These methods are not practical for monitoring air quality in time. Here, in this research we present a low-cost IoT-based smart air quality monitoring and prediction system. Data from multiple sensors are used to assess the air quality. These sensors can detect the level of PM2.5 and PM10 in the air. They also measure Carbon Monoxide using the MQ7 sensor and Nitrogen Dioxide using the MQ135 sensor. The Air Quality Index is also affected by the temperature and humidity, around us which are monitored by the Arduino Uno microcontroller. The Air Quality Index is figured out using the parameters I mentioned earlier like the level of PM2.5 and PM10. It follows the method used by the CPCB. The Air Quality Index is determined based on these parameters. Cleaned sensor data is then fed into the system via smoothing and feature scaling to train the model. A one-dimensional CNN model is designed to identify and analyze patterns within the time-series data to predict the AQI. Techniques to handle data are also applied in order to prevent data leakage. Performance metrics such as accuracy, recall, precision and F1 score were used in order to measure the efficacy of the model. A streamlit-based web interface is developed which displays current AQI value, air quality safety status, minute-wise future predictions and health recommendations.
Keywords: Internet of Things, Air Quality Monitoring, Convolutional Neural Network, AQI Prediction, Environmental Sensors, Real Time Monitoring.

I. INTRODUCTION

Air pollution is an issue because of all the bad things that people are releasing into the air from cars and factories and stuff. The problem of air pollution is getting bigger because of air pollution from cars and trucks and air pollution from factories and air pollution, from all the things that people do. Unfortunately, there are no common methods of measuring air quality that are within the range required, due to their high cost. We only have air quality readings every now. Then and this means we cannot keep a constant check on the air quality. So, the main issue is to make a system that works well does not cost much and is good at checking and predicting the air quality level, which is also known as the air quality index or air quality. We need a system, for air quality that can do this job. The air quality is very important. We need to find a way to monitor the air quality all the time.

The emergence of IoT and the advent of machine learning algorithms have made it possible to create and develop intelligent air quality monitoring systems. IoT technology enables integration of different types of sensors (e.g., temperature, pressure, PM2.5, PM10, air quality sensors (MQ7, MQ135) for measuring CO, NO_2) to collect real time data. Collected data can then be processed using machine learning algorithms to recognize patterns and predict future states of air quality. This is particularly true for Convolutional Neural Networks (CNN) that are primarily designed for time-series data analysis.

The existing conventional systems have their own set of problems, and therefore this paper proposes a much intelligent approach for air quality monitoring and prediction using Internet of Things (IoT). Various types of smart sensors have been used to capture the real time data of air quality parameters, which then goes for smoothing and scaling process and then these data are put into a 1D CNN model to predict. All the results are displayed on a web based Streamlit web application with current AQI and safety indicator along with minute-by-minute forecast and health guidelines.

II. RELATED WORK

In recent years, there has been growing interest in combining Internet of Things (IoT) technology with machine learning to improve air quality monitoring. While conventional environmental monitoring methods have several inherent disadvantages, including poor spatial coverage and delayed data transmission and processing, IoT technology offers real-time environmental monitoring data analysis. This study is, about making a system to check the air quality.

We want to use devices that can send information to other devices. Some other people have already done work on this. They figured out that when we use these devices with computer programs the information we get from them becomes more accurate and reliable. We are now working on creating a system that uses the internet to make this happen.
This system will help us get information from these devices.
The goal is to make the most of these devices and the internet to get information. These devices to check the air quality.

A. IoT-Based Environmental Parameter Monitoring Systems

In this paper, we present several research works conducted to develop and monitor various environmental IoT systems that collect different environmental parameters. We design and implement a system consisting of several wireless sensors for detecting particulate matter (PM2.5 and PM10), CO, NO_2, temperature and humidity levels. All collected data are processed, stored and visualized on the developed IoT system. Several challenges exist to efficiently monitor the environment, the most significant of them being the unreliability of the collected environmental data because of noise and/or erroneous readings from faulty sensors.

B. Machine Learning Models

Several approaches of machine learning like decision trees, SVM and ANN have already utilized the dataset of air quality historical data to predict AQI. In recent times, CNN, as one of the deep learning approaches, has also been applied to capture the time series dynamics of sensor readings, which potentially can further improve the prediction accuracy. To effectively apply CNN, it is important to appropriately process input feature, possess sufficient amount of training data, and have adequate computational resources.

C. Data Pre-processing and Visualizations

Data preprocessing and visualization techniques play significant roles in the development of air quality monitoring systems. In data preprocessing, techniques such as data filtering and normalization, and time series data analysis are performed to obtain meaningful results. In visualizing the web-based air quality data, dashboards provide easy monitoring of AQI levels, trends and safety with simple navigation and interactivity. Web dashboards present an interactive form of data visualization which relies on continuity of data collection and transmission.

D. Existing System Issues

Current systems have some problems. They struggle to keep data consistent with the latest technology. Managing time-series data. Making real-time predictions is also tricky. Most systems today focus on collecting data. They do not have features to generate predictions, like safety guidelines or health advice. Linking an IoT system with machine learning technologies and creating a user-friendly graphical interface is not easy.

E. Summary

Even though currently available strategies can solve individual air quality problems in an urban space, a unified approach is also required for the entire space. Such an integrated approach will involve sensing the real-time environmental parameters, predicting the levels of air quality using Deep Learning (DL) techniques like Convolutional Neural Networks (CNN), and visualization of air quality indices through interfaces such as a Streamlit dashboard.

F. Survey and Review Studies

There are review articles on IoT, cloud and machine learning applications for environmental monitoring. Integration between sensing capabilities and efficient applications of sophisticated analytical techniques is critical. Real time monitoring, prediction and visualisation are critical features of environmental monitoring systems.
While there are existing methods and techniques within the literature for air quality monitoring, there is scope for the development of a system that incorporates real-time monitoring, prediction and visualisation.

III. METHODS AND MATERIALS

We propose a system which integrates a solution that can enable real-time air quality monitoring and prediction of air quality index using IoT-based air quality sensors data and deep learning. The system collects, preprocess, stores and predicts air quality data by using Convolutional Neural Network (CNN) approach.

A. Data Acquisition from Environmental Sensors
Environmental sensors like a dust sensor, gas sensors (MQ7 & MQ135 for CO & various air pollutants like NO_2), temperature & humidity sensors are interfaced to make this project. These sensors feed the data to an Arduino microcontroller which logs the values in real time and then sends the logged data to the computer where it would be processed and analyzed.

B. Storage and Logging of Sensor Readings
The sensor readings are saved into a CSV file so the user can look at them later. The sensor readings have a lot of information, like the time and dust levels and gas readings and temperature and humidity. This is all saved in the CSV file for the user to use later with the sensor readings.

C. Data Preprocessing and Feature Engineering
In the first step, the quality of the available data is improved by applying a smoothing effect (rolling average) to it. Additional air quality parameters like PM2.5, PM10, CO, NO2 are calculated from the raw sensor readings. The data is then normalized to the same scale for all features by using Minmax normalization. Finally, sequences are build from the scaled data in order to incorporate temporal dependency into the data that shall be used for training a model.

D. 1D Convolutional Neural Network (CNN)
To analyse the time-series sensor data and classify it into different Air Quality Index (AQI) categories, a one-dimensional convolutional neural network (CNN) architecture is employed. The network consists of many Conv1D layers that extract features from the time-series data. To improve training and reduce the dimensionality of the extracted features, batch normalization and 1D pooling layers are used in between the convolutional layers. Finally, dropout layers are included to prevent overfitting, followed by dense layers for final classification.

E. Real Time Predictive System
A deep model is trained to build a convolution neural network model. The trained model uses a sequence of sensor reading data as an input, and predicts AQI in real time. The output of the predicted AQI value is converted to a more meaningful form by applying CPCB sub-index formula to categorize the obtained AQI value into: Good, Satisfactory, Moderate, Poor, Very Poor, and Severe. In addition to AQI category, the model will also provide the health status according to the AQI category value predicted.

F. Visualization & Dashboard
This project is designed as a web application that provides interaction to users through a Streamlit framework. Real time data such as Air Quality Index (AQI) along with AQI meter, current status of the sensors and minute by minute predictions of AQI will be depicted in the user interactive web interface along with relevant charts showing AQI readings.

G. Process Flow
The system gets real time information from sensors. This information includes how much dust is in the air how much gas is in the air the temperature, in the room and how humid the room is. The system uses this information to predict the Air Quality Index. It also groups the air quality into groups and tells us about the health problems that can happen because of the Air Quality Index. The system does all this to help us understand the Air Quality Index and how it affects our health.

1. Real-world sensor data is collected.
2. The data is transmitted and saved in a timestamped CSV file.
3. Smoothing and other pre-processing operations (smoothing, feature transformation, normalization) have been performed on the data.
4. Time-series datasets are generated.
5. CNN predicts AQI categories.
6. The predictions are then displayed on the Streamlit-based user interface in the form of plots and warnings.

This study presents a smart air quality monitoring system based on IoT-sensor, deep learning of time-series, and visualization technology for real-time monitoring and AQI prediction with smart, effective, and economic benefits.

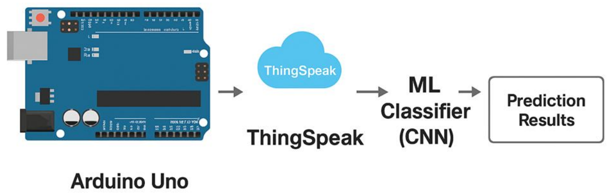


Fig 1. Architecture Diagram of the proposed system

IV. EXPERIMENTAL STUDY

In this section, we present an experiment for evaluation of the proposed solution performance in terms of data collection, data processing, and deep learning air quality predictions. We also discussed integration of all the parts of the solution. The main objective of presented work is efficient air quality assessment and monitoring.

For the experiments, a CSV file with time series data of several parameters like dust concentration and the values obtained from MQ7 (CO), MQ135 (NO_2) sensors, temperature and humidity with their respective timestamps were collected and verified. Information about the collected data was used for the experimentation and also for the correctness and accuracy of the data collected. A data pre-processing approach that includes rolling average smoothing and conversion of features (PM2.5, PM10, CO, NO_2) are shown to enhance the quality of the data collected.

Regarding software experiments, a Python script has been written to process data for the upcoming software experiments. The script first performs Normalization (Minmax scaling) and then converts the data into Time Series format suitable for a Deep Learning Model. The data is then split into two datasets: One set is used for training and another set for testing purposes. The second dataset is created as a sequential dataset.

For prediction a 1-Dimensional Convolutional Neural Network (CNN) model was designed and trained on the provided cleaned data. The designed model contains some layers of convolution, pooling and dense layers which are responsible for feature extraction and classification. Accuracy, precision, recall and F1-Score were used to measure the performance of the model. Experiments performed on the model are performing well and learning from the environmental data.

The model has been deployed to Streamlit application for live predictions. The application uses a dataset for training purposes and makes predictions on real time data, it outputs the AQI value, quality and safety advice. There’s also an AQI meter and safety status on the streamlit dashboard that helps to better understand the current air quality.

We also provide 1-hour ahead forecasts of air quality in the form of minute-by-minute values of the air quality index. These average values and forecasts for each sensor are provided on an time-line with corresponding timestamps.

We also experimented with displaying AQI trend charts and accuracy metrics for prediction (accuracy, precision, recall, F1 score, etc.). These provided insights into the environment and system performance. Prediction combined with visualization made the system more user-friendly.

The experiments demonstrate a successful integration of time series analysis tools, deep learning based air quality indicator (AQI) forecasting CNN, and data visualization functionalities, and the system is efficient enough for real-time monitoring and decision making

V. RESULTS

From the experiment result, it is obvious that the system is able to monitor the real-time air quality reading and forecast the Air Quality Index (AQI) correctly. The system is able to capture the input readings from different types of sensors and utilize the CNN approach in predicting the AQI. The integration of time series, machine learning approach and data visualization enable the system to function effectively.

A. *Data Collection and Processing*

The sensor data used for this study consists of dust (at different times) in form of particles, MQ7 (for CO), MQ135 (for NO2), temperature and humidity level. The collected data has been processed and organized for usage. Smoothing (Rolling Average), feature extraction (PM2.5 and PM10) and data normalization has been done for preprocessing.

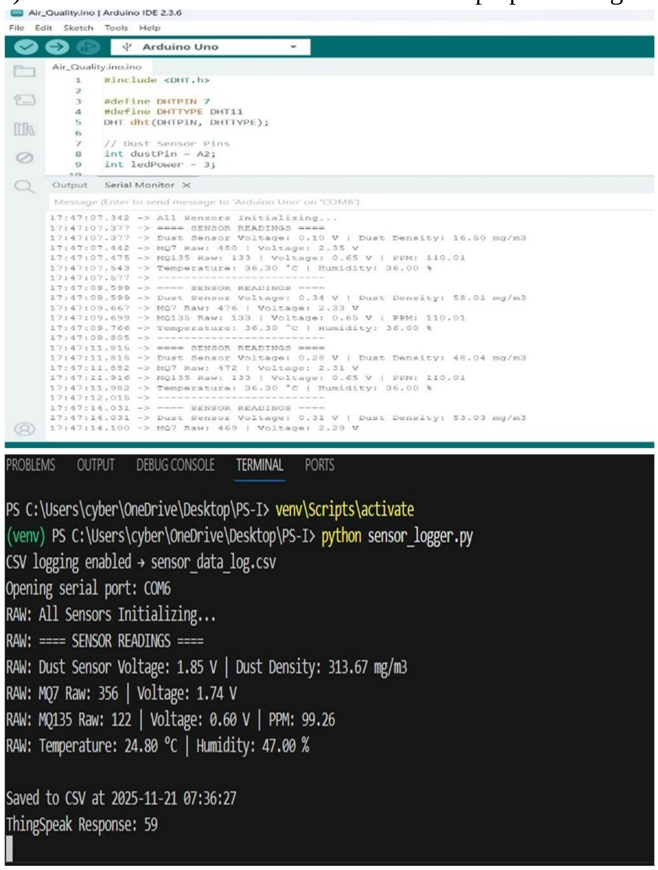


Fig 2. Capturing sensor data

From Fig 2., we can see how data is getting captured from the different sensors.

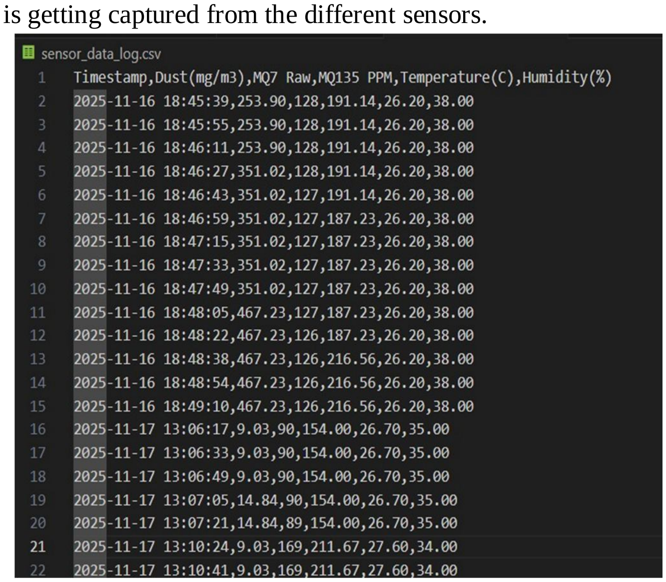


Fig 3. CSV file of data

From Fig 3., the data from sensors is saved as a csv file

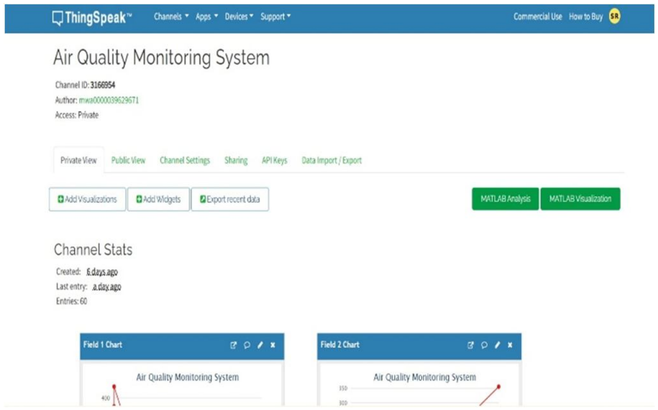


Fig 4. ThingSpeak overview

From Fig 4., the data is transmitted to the thingspeak and a channel is created called air quality monitoring system.

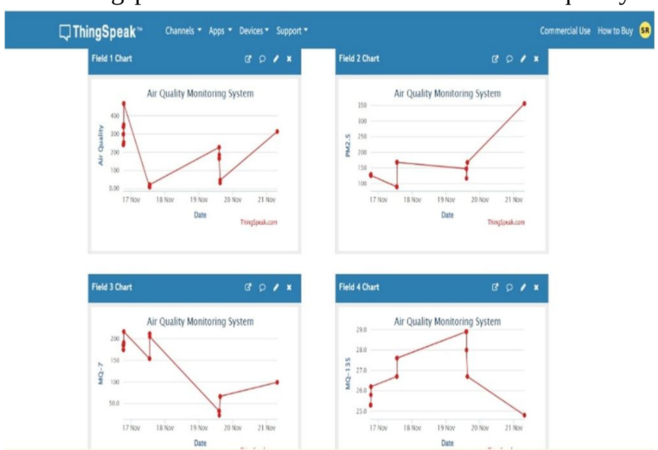


Fig 5. Live visualization in thingspeak

From Fig 5., the sensor data is imported into the thingspeak and live graphs are displayed for each attribute.

B. Predicting AQI using CNN Model

We used 1D CNN in our experiment to predict the AQI level by training the model with time series data. The model was able to learn the environmental factors that affected the AQI level. In addition, we evaluated the performance of the model in terms of accuracy, precision, recall and F1-score.

Good job on classification of AQI into Good, Satisfactory, Moderate and Poor categories. Utilization of time series feature enhanced the accuracy of predictions..

```
(venv) PS C:\Users\cyber\OneDrive\Desktop\ps-2> python train.py
2026-04-19 18:02:06.999925: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly
due to floating-point round-off errors from different computation orders. To turn them off, set the environment variab
.
2026-04-19 18:02:09.562731: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly
due to floating-point round-off errors from different computation orders. To turn them off, set the environment variab
.

◆ STEP 1: RAW DATA
             Timestamp  Dust(mg/m3)  MQ7 Raw  MQ135 PPM  Temperature(C)  Humidity(%)
0  2026-02-22 23:28:16        55.52      273     135.43            31.3         46.0
1  2026-02-22 23:28:32        55.52      273     135.43            31.3         46.0
2  2026-02-22 23:28:48        55.52      273     135.43            31.3         46.0
3  2026-02-22 23:29:04        89.55      273     135.43            31.3         46.0
4  2026-02-22 23:29:21        89.55      272     135.43            31.3         46.0

◆ STEP 2: AFTER CONVERSION
             Timestamp  Dust(mg/m3)  MQ7 Raw  MQ135 PPM  Temperature(C)  Humidity(%)   PM2.5     PM10    CO     NO2
0  2026-02-22 23:28:16        55.52      273     135.43            31.3         46.0  33.312   66.624  2.73  67.715
1  2026-02-22 23:28:32        55.52      273     135.43            31.3         46.0  33.312   66.624  2.73  67.715
2  2026-02-22 23:28:48        55.52      273     135.43            31.3         46.0  33.312   66.624  2.73  67.715
3  2026-02-22 23:29:04        89.55      273     135.43            31.3         46.0  53.730  107.460  2.73  67.715
4  2026-02-22 23:29:21        89.55      272     135.43            31.3         46.0  53.730  107.460  2.72  67.715
```

Fig 6. Collected data

From Fig 6., the data preview of the collected data.

```
STEP 3: AFTER SMOOTHING
          Timestamp  Dust(mg/m3)    MQ7 Raw   MQ135 PPM  Temperature(C)  Humidity(%)   PM2.5    PM10        CO        NO2
2026-02-22 23:28:48   55.520000  273.000000  135.430000            31.3         46.0  33.312   66.624  2.730000  67.715000
2026-02-22 23:29:04   66.863333  273.000000  135.430000            31.3         46.0  40.118   80.236  2.730000  67.715000
2026-02-22 23:29:21   78.206667  272.666667  135.430000            31.3         46.0  46.924   93.848  2.726667  67.715000
2026-02-22 23:29:37   89.550000  272.333333  135.753333            31.3         46.0  53.730  107.460  2.723333  67.876667
2026-02-22 23:29:53   89.550000  272.000000  136.076667            31.3         46.0  53.730  107.460  2.720000  68.038333

STEP 4: AQI VALUES
       AQI
109.125000
109.125000
109.083333
109.041667
109.000000

STEP 5: LABEL DATA
       AQI  AQI_Label
109.125000          2
109.125000          2
109.083333          2
109.041667          2
109.000000          2
```

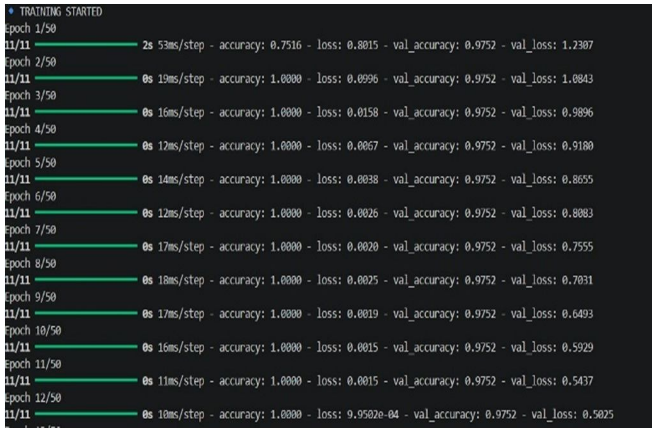
```
◆ STEP 6: FEATURES SAMPLE
[[ 33.312       66.624       2.73        67.715       31.3
   46.        ]
 [ 40.118       80.236       2.73        67.715       31.3
   46.        ]
 [ 46.924       93.848       2.72666667  67.715       31.3
   46.        ]
 [ 53.73       107.46       2.72333333  67.87666667  31.3
   46.        ]
 [ 53.73       107.46       2.72        68.03833333  31.3
   46.        ]]

◆ STEP 7: SPLIT DONE

◆ STEP 8: SCALED SAMPLE
[[0.28991707 0.28991707 0.23076923 0.40669759 0.10638298 0.91666667]
 [0.38894224 0.38894224 0.23076923 0.40669759 0.10638298 0.91666667]
 [0.48796741 0.48796741 0.22916667 0.40669759 0.10638298 0.91666667]
 [0.58699258 0.58699258 0.2275641  0.40890283 0.10638298 0.91666667]
 [0.58699258 0.58699258 0.22596154 0.41110808 0.10638298 0.91666667]]
```

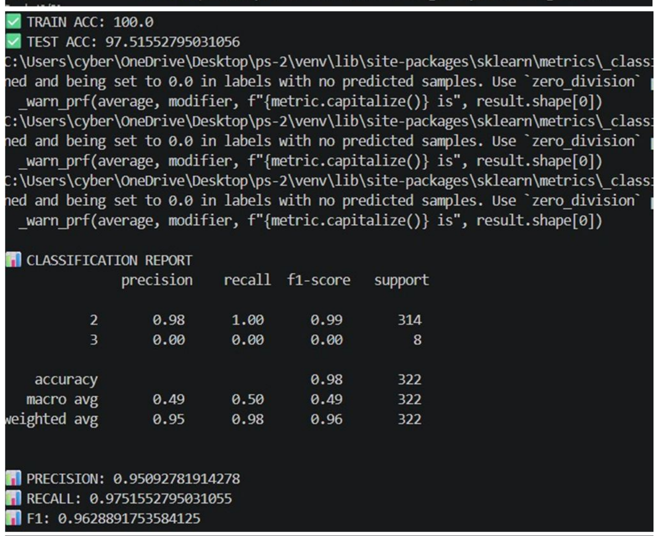
Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 10, 10)	0
conv1d (Conv1D)	(None, 8, 32)	992
batch_normalization (BatchNormalization)	(None, 8, 32)	128
conv1d_1 (Conv1D)	(None, 6, 64)	6,208
batch_normalization_1 (BatchNormalization)	(None, 6, 64)	256
max_pooling1d (MaxPooling1D)	(None, 3, 64)	0
dropout (Dropout)	(None, 3, 64)	0
flatten (Flatten)	(None, 192)	0
dense (Dense)	(None, 64)	12,352
dense_1 (Dense)	(None, 32)	2,080
dense_2 (Dense)	(None, 4)	132

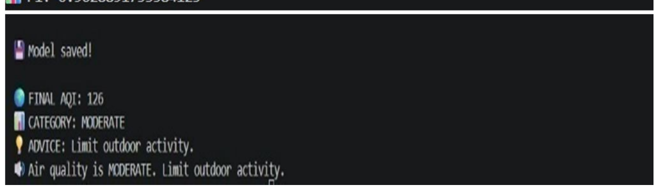
Total params: 22,148 (86.52 KB)
Trainable params: 21,956 (85.77 KB)
Non-trainable params: 192 (768.00 B)

Fig 7. CNN model processing

From Fig 7., the sensor data is processed at each layer of the CNN model.


TRAINING STARTED
Epoch 1/50
11/11 2s 53ms/step - accuracy: 0.7516 - loss: 0.8015 - val_accuracy: 0.9752 - val_loss: 1.2307
Epoch 2/50
11/11 0s 19ms/step - accuracy: 1.0000 - loss: 0.0996 - val_accuracy: 0.9752 - val_loss: 1.0843
Epoch 3/50
11/11 0s 16ms/step - accuracy: 1.0000 - loss: 0.0158 - val_accuracy: 0.9752 - val_loss: 0.9896
Epoch 4/50
11/11 0s 12ms/step - accuracy: 1.0000 - loss: 0.0067 - val_accuracy: 0.9752 - val_loss: 0.9180
Epoch 5/50
11/11 0s 14ms/step - accuracy: 1.0000 - loss: 0.0038 - val_accuracy: 0.9752 - val_loss: 0.8655
Epoch 6/50
11/11 0s 12ms/step - accuracy: 1.0000 - loss: 0.0026 - val_accuracy: 0.9752 - val_loss: 0.8083
Epoch 7/50
11/11 0s 17ms/step - accuracy: 1.0000 - loss: 0.0020 - val_accuracy: 0.9752 - val_loss: 0.7555
Epoch 8/50
11/11 0s 18ms/step - accuracy: 1.0000 - loss: 0.0025 - val_accuracy: 0.9752 - val_loss: 0.7031
Epoch 9/50
11/11 0s 17ms/step - accuracy: 1.0000 - loss: 0.0019 - val_accuracy: 0.9752 - val_loss: 0.6493
Epoch 10/50
11/11 0s 16ms/step - accuracy: 1.0000 - loss: 0.0015 - val_accuracy: 0.9752 - val_loss: 0.5929
Epoch 11/50
11/11 0s 11ms/step - accuracy: 1.0000 - loss: 0.0015 - val_accuracy: 0.9752 - val_loss: 0.5437
Epoch 12/50
11/11 0s 10ms/step - accuracy: 1.0000 - loss: 9.9502e-04 - val_accuracy: 0.9752 - val_loss: 0.5025


TRAIN ACC: 100.0
TEST ACC: 97.51552795031056
C:\Users\cyber\OneDrive\Desktop\ps-2\venv\lib\site-packages\sklearn\metrics_class
ned and being set to 0.0 in labels with no predicted samples. Use `zero_division`
_warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
C:\Users\cyber\OneDrive\Desktop\ps-2\venv\lib\site-packages\sklearn\metrics_class
ned and being set to 0.0 in labels with no predicted samples. Use `zero_division`
_warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
C:\Users\cyber\OneDrive\Desktop\ps-2\venv\lib\site-packages\sklearn\metrics_class
ned and being set to 0.0 in labels with no predicted samples. Use `zero_division`
_warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
CLASSIFICATION REPORT
precision recall f1-score support
2 0.98 1.00 0.99 314
3 0.00 0.00 0.00 8
accuracy 0.98 322
macro avg 0.49 0.50 0.49 322
weighted avg 0.95 0.98 0.96 322
PRECISION: 0.95092781914278
RECALL: 0.9751552795031055
F1: 0.9628891753584125


Model saved!
FINAL AQI: 126
CATEGORY: MODERATE
ADVICE: Limit outdoor activity.
Air quality is MODERATE. Limit outdoor activity.

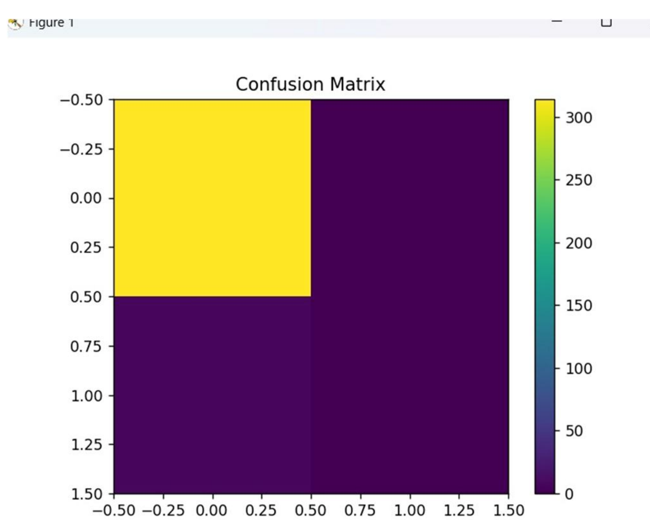
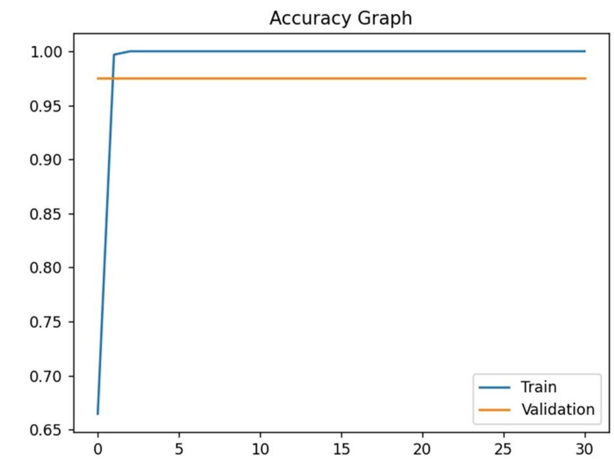
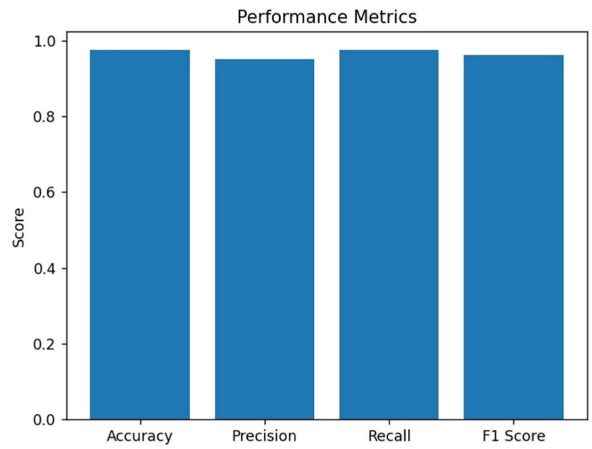


Figure 1
Confusion Matrix
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50
300
250
200
150
100
50
0





C. Real-Time Prediction and Monitoring

System evaluation was carried out real-time by making predictions on real-time data utilizing Streamlit developed user-friendly dashboard. Real-time data was fed into the trained CNN model and AQI predictions, categories, and health advisories were obtained.

We have added a color coded AQI meter as well as a live safety status table which uses CPCB guidelines to show whether each pollutant is safe, moderately safe or unsafe.

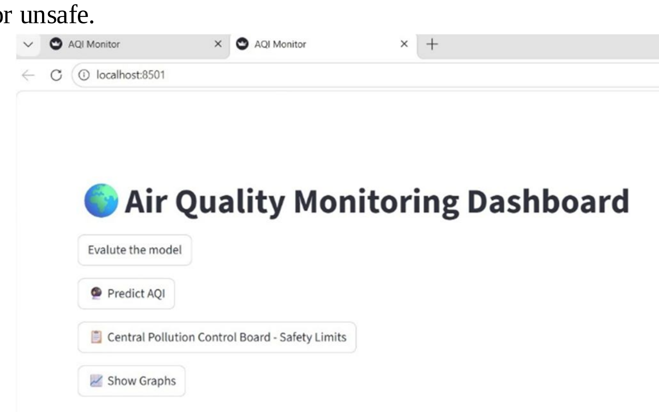


Fig 8. Monitoring dashboard

From Fig 8., the real time monitoring dashboard is shown and display all necessary outputs.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

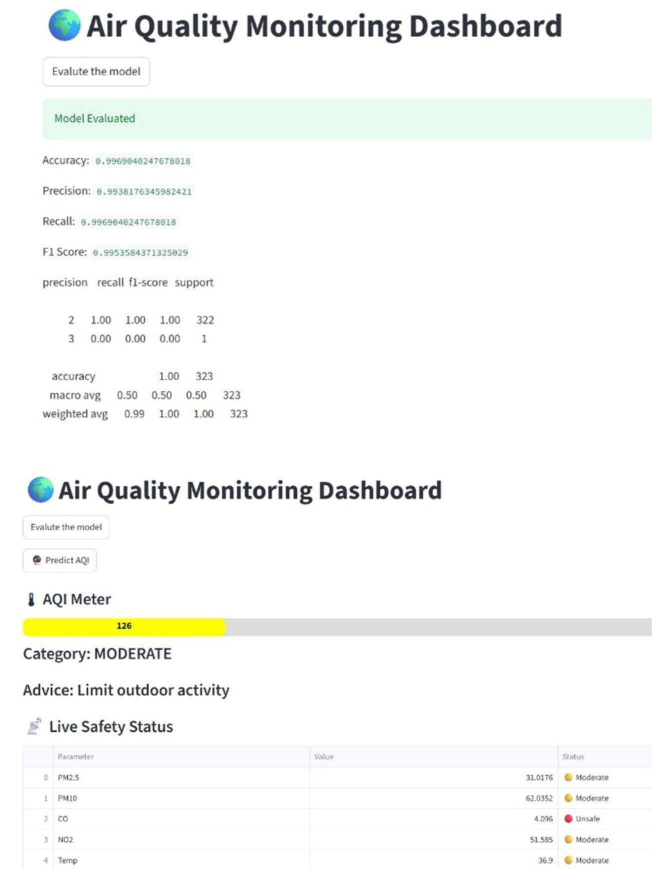


Fig 9. Prediction of AQI

D. Time Series Analysis (AQI on a Minute Basis)

The software aggregates data and timestamps it at a one minute interval in order to calculate the Air Quality Index. If there are multiple values within the minute then the software will calculate the mean of the values within that minute and use that for the Air Quality Index calculation.

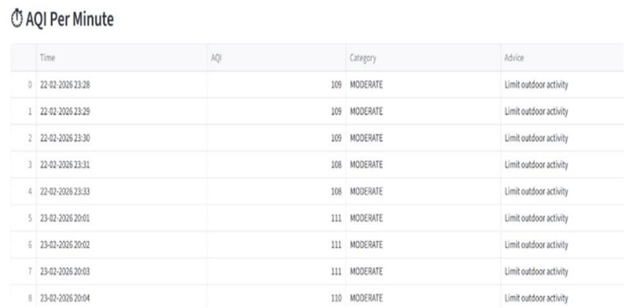


Fig 10. Minute-wise prediction of AQI

From Fig 10., the air quality monitoring dashboard that displays the minute-wise aqi prediction and alerts.

E. Visualization & Graphical Analysis

The model also includes graphical analysis capability to help evaluate the surroundings and assess model performance through visual AQI trends and various metrics plots.

Accuracy, Precision, Recall, and F1-Score Graphs provide a better insight into the accuracy of the CNN model deployed for air quality monitoring. AQI Trend Graph provides users with an overview of the changes in air quality over time.

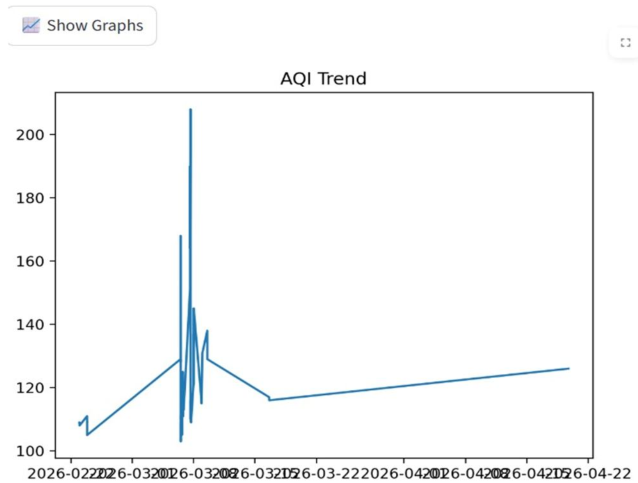
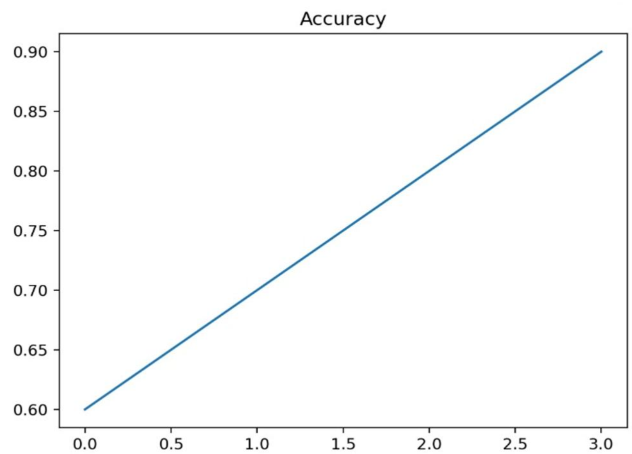
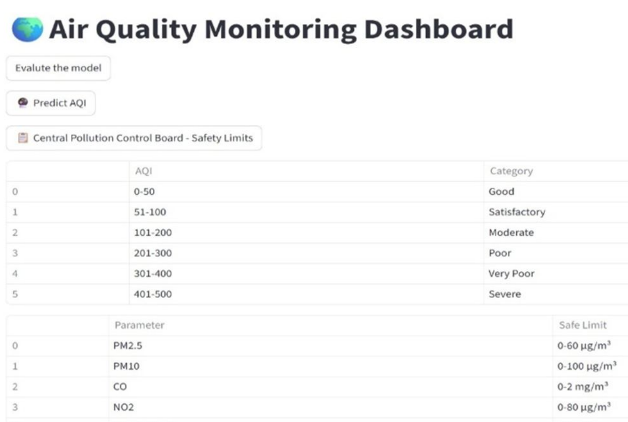


Fig 11. Graph of AQI Trend



F. Safety Limits and Awareness about Environment

In addition to the air quality limits for pollute constituents (PM2.5, PM10, CO, NO_2) current temperature and humidity is displayed according to weather criteria. The values are also compared against safety limits as laid down by the Central Pollution Control Board (CPCB).



	AQI	Category
0	0-50	Good
1	51-100	Satisfactory
2	101-200	Moderate
3	201-300	Poor
4	301-400	Very Poor
5	401-500	Severe

	Parameter	Safe Limit
0	PM2.5	0-60 µg/m³
1	PM10	0-100 µg/m³
2	CO	0-2 mg/m³
3	NO2	0-80 µg/m³
4	Temperature	20-30°C

Fig 12. CPCB safety limits of aqi and gases.

From Fig 12., the safety limits of the gases according to cpcb are shown.

G. Comparison of Models

Linear Regression

It is appropriate for tracking general trends but fails for complicated nonlinearity in air quality information.

Random Forest

GBM gets high accuracy because it is good at learning non-linear relationships between features as well as interactions between features. It's a very good traditional ML method.

KNN

The task of clustering is notoriously difficult for poor results, since it is very sensitive to noise and high dimensional data.

Naive Bayes

Simple and fast, but makes the assumption of feature independence, which is unrealistic for related environmental features.

SVM

This performer is worst for this problem set because it is very poor on sequential data.

CNN

It has the highest accuracy, since it takes temporal dependencies and interdependencies of sensor readings into consideration.

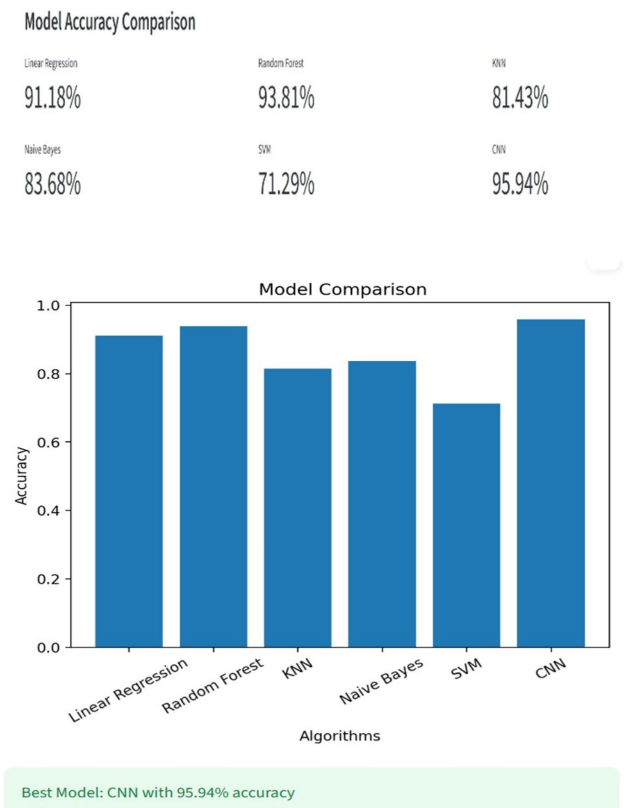


Fig 13. Model Comparison Graph

From Fig 13., the comparison of different models.

VI. CONCLUSION

In this research, smart air quality monitoring and predicting system, based on Internet of Things (IoT) and Deep Learning Techniques, was designed and implemented. The system for air quality monitoring through conventional method has a lot of limitations and problems that can be solved by this smart monitoring system. The smart system for air quality monitoring collects data in continuous basis, process the collected data, and analyze the situation around the environment. Some parameters such as particulate matter, gases, humidity, and temperature around the monitoring location can be detected using the sensors attached in the system.

After we gather all the data we need to first process it in order to calculate the Air Quality Index and then feed it into our CNN model to predict the AQI. I also created a simple interface using Streamlit that displays information about the real time status of individual sensors, current AQI value, safety level and trends in a graphical form.

It supports more data aggregations like minute by minute, by time, with safety state information, with an AQI meter and alerts. All common visual widgets like charts and tables are supported, as well as notifications. Experimental results show that the chosen CNN is more accurate than standard machine learning approaches for this task.

REFERENCES

[1] H.-C. Hsu, G. B. Tarekegn, M. E. Getnet, K.-P. Yu, and L.-C. Tai, "UAV-Based Real-Time Air Quality Monitoring and Prediction Using Machine Learning Approach," IEEE Internet of Things Journal, 2025.

[2] J. Rosa-Bilbao, F. S. Butt, D. Merkl, M. F. Wagner, J. Schäfer, and J. Boubeta-Puig, "IoT- Based Indoor Air Quality Management System for Intelligent Education Environments," IEEE Internet of Things Journal, vol. 12, no. 11, 2025.

[3] E. F. Ladeira and B. M. C. Silva, "Machine Learning-Based Decision Support System for Hazardous Gas Monitoring in Rural Areas," IEEE Access, 2025.

[4] A. M. Ali, A. Al-Qerem, M. R. Hassan, A. Abu-Khadrah, and M. Jarrah, "Traffic Control System for Pollution Control Using Capsule Networks," IEEE Access, 2025.

[5] N. Silva, E. Alves, W. Júnior, A. Moares, N. Silva, and A. Balieiro, "Real-Time Air Quality Monitoring Using FreeRTOS on ESP32-LoRaWAN IoT Devices in Smart Campus Environments," IEEE Latin America Transactions, 2025.

[6] A. Mazinani, D. Antonucci, D. P. Pau, L. Davoli, and G. Ferrari, "Air Quality Prediction via Embedded ML/DL and Quantized Models," IEEE Access, 2025.

[7] S. J. Livingston, S. D. Kanmani, A. S. Ebenezer, D. Sam, and A. Joshi, "Air Quality Prediction Using Ensemble ML Models Combining SVM, RF, ANN, RNN, and Fuzzy Logic," Measurement: Sensors, Elsevier, 2023.

[8] A. Gangwar, S. Singh, R. Mishra, and S. Prakash, "IoT, Big Data, and Machine Learning Techniques for Air Pollution Monitoring and Forecasting: A Comprehensive Review," Wireless Personal Communications, Springer, 2023.

[9] S. Al-Eidi, F. Amsa, O. Darwish, Y. Tashtoush, A. Alqahtani, and Niveshitha, "Comparative Study of ML Algorithms for Air Quality Prediction in Smart Cities," IEEE Access, 2023.

[10] J. Merino, D. Ikeuchi, N. Moretti, A. Mukerjee, L. Li, S. Karatzas, S. Pattinson, and A. Parlikad, "Virtual Sensor Architecture for Indoor Air Quality Monitoring in Manufacturing,"1st Workshop on Low-Cost Digital Solutions for Industrial Automation (LoDiSA,2023),IET, 2023.

[11] J. Mabrouki, M. Azrour, D. Dhiba, Y. Farhaoui, and S. El Hajjaji, "IoT-Based Data Logger for Weather and Air Quality Monitoring," Big Data Mining and Analytics, KeAi / Elsevier, 2021.

[12] C. C. Goh et al., "In-Vehicle Air Quality Monitoring System Using Machine Learning Models," Sensors (MDPI), 2021.

[13] D. Zhang and S. S. Woo, "Hybrid IoT Sensing Network for Air Quality Monitoring Using Fixed and Mobile Sensors," IEEE Access, 2020.

[14] D. Iskandaryan, F. Ramos, and S. Trilles, "Machine Learning Techniques for Air Quality Prediction in Smart Cities: A Comparative Study," Applied Sciences (MDPI), 2020.

[15] B. Wang, W. Kong, H. Guan, and N. N. Xiong, "A Two-Layer Hybrid LSTM–GRU Model for PM2.5 Prediction and Smog Forecasting," IEEE Access, 2019.

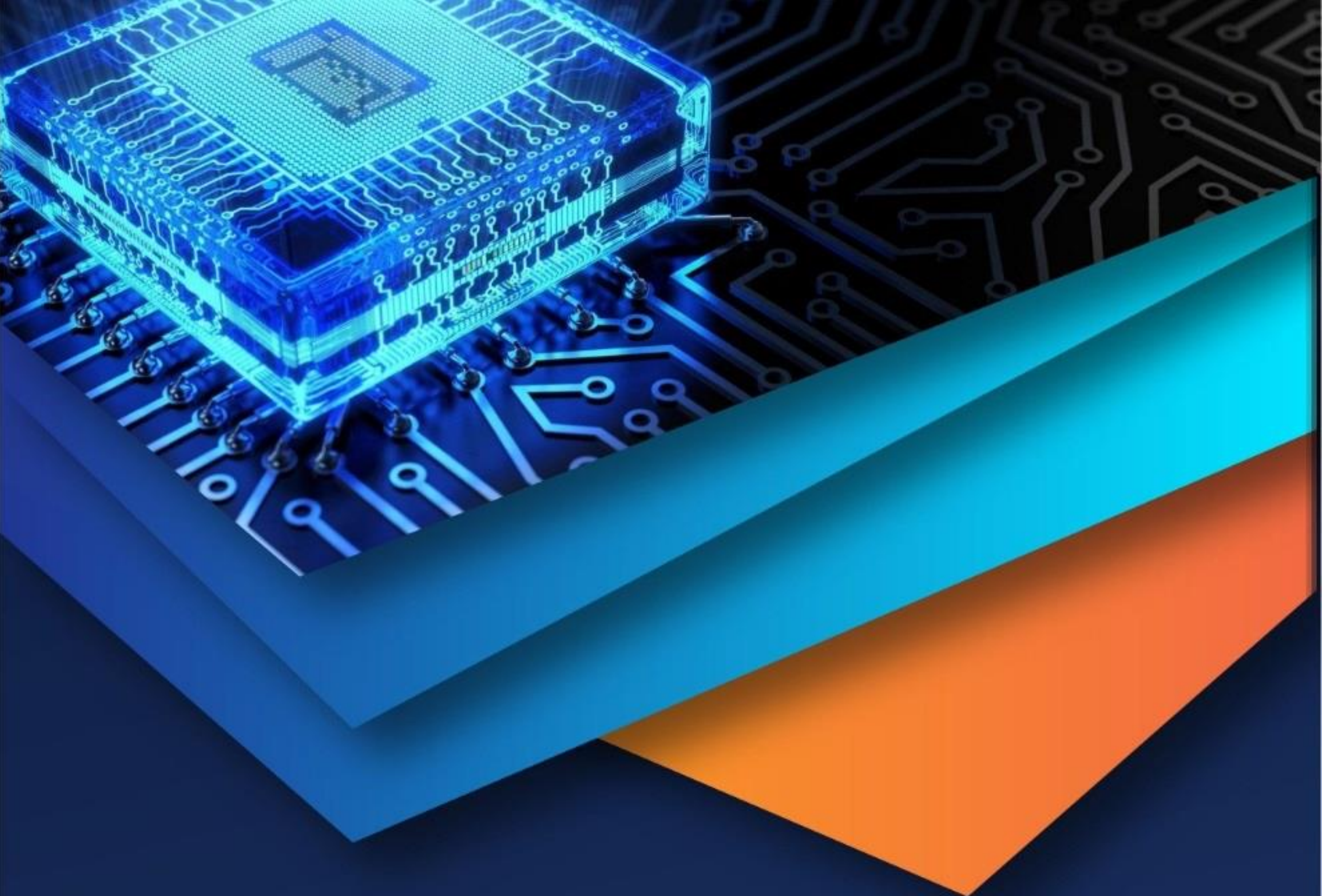
[16] K. B. K. Sai, S. Ramasubbareddy, and A. K. Luhach, "IoT-Based Air Quality Monitoring System Using ThingSpeak and ML Analysis," Scalable Computing: Practice and Experience (SCPE), 2019.

[17] A. Mishra, "IoT-Based Air Pollution Monitoring System Using Machine Learning Models," Proc. 2018 Int. Conf. on Applied Electromagnetics, Signal Processing and Communication (AESPC), IEEE, 2018.

[18] T. W. Ayele and R. Mehta, "IoT-Based Air Pollution Monitoring and Prediction Using LSTM,"Proc. 2nd Int. Conf. on Inventive Communication and Computational Technologies(ICICCT), IEEE, 2018.

[19] G. K. Kang, J. Z. Gao, S. Chiao, S. Lu, and G. Xie, "Big Data and Machine Learning Approaches for Air Quality Prediction," Int. Journal of Environmental Science and Development(IJESD), 2018.

[20] V. H. Khandekar and P. Srinath, "Comparative Analysis of ML Techniques for Air Quality Forecasting," Proc.2017 int. Conf. on Computing, Communication, Control and Automation(ICCUBEA), IEEE, 2017.



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY