Multidisciplinary | Peer-Reviewed
Open Access | Research Journal
ISSN: 3139-2954

Article ID: IRJIST-2026-098
Volume 1 | Issue 4 | 2026
DOI: https://doi.org/10.67308/irjist.098

International Research Journal of Innovation in Science and Technology

Real-Time Air Quality Detection and Prediction System

Bhausaheb P. Dane[1], Shivam S. Sonawane[2], Aishwarya M. Sonawane[3], Supesh P. Sonar[4], Sumit P. Suryawanshi[5]

[1-5]Department of Engineering, Sciences and Humanities (DESH), Vishwakarma Institute of Technology, Pune, Maharashtra, India
Correspondence: [2]shivamsonawane1916@gmail.com

Abstract

Air pollution is a growing environmental concern that affects human health and the environment. This paper presents a low-cost Real-Time Air Quality Detection and Prediction System using an Arduino Uno, MQ135 gas sensor, and DHT11 temperature sensor. The system collects and processes sensor readings to generate indicative PPM and AQI-style values. A multiple linear regression model is used to predict air-quality trends for the next ten time steps. A Flask-based web dashboard displays real-time readings, temperature, air-quality categories, graphs, and predicted values. Tests conducted in indoor, outdoor, and smoke-exposed environments demonstrated the system's response to changing environmental conditions. Approximately 70–75% of predicted readings were reported within the study's accuracy tolerance. The proposed system provides a low-cost approach to local air-quality awareness and short-term prediction.

Keywords: Air Quality Monitoring; Air Quality Prediction; Machine Learning; MQ135; Arduino Uno; Data Visualization.

1. Introduction

Air pollution has become a serious environmental concern in cities and towns worldwide. Rapid industrialisation, urbanisation, and the increasing number of automobiles have contributed significantly to the deterioration of air quality. Emissions of particulate matter (PM), carbon monoxide (CO), carbon dioxide (CO_2), nitrogen oxides, and other pollutants are among the major contributors to air pollution. Prolonged exposure to polluted air may increase the risk of respiratory and cardiovascular diseases, including asthma, lung cancer, and other serious health conditions [9], [10]. In addition to its effects on human health, air pollution poses a significant threat to the environment and affects the quality of life of millions of people, particularly in densely populated urban areas.

Air pollution levels vary considerably across time and geographical locations and are influenced by several factors, including the time of day, traffic density, weather conditions, and human activities. Consequently, a single measurement provides only limited information about the prevailing air-quality conditions. Continuous monitoring is therefore important not only for identifying existing pollution levels but also for understanding their variations and predicting possible future trends. Air-quality forecasting can help individuals exposed to elevated pollution levels take timely precautions. For example, people may postpone strenuous outdoor activities until pollution levels decrease, thereby reducing their exposure.

Conventional air-pollution monitoring systems generally rely on networks of reference-grade monitoring stations, often operated by government agencies. In India, the Central Pollution Control Board (CPCB) operates monitoring stations that provide information through the National Air Quality Index (NAQI) [11]. These stations use sophisticated and calibrated instruments to provide reliable measurements. However, their installation and operation involve substantial costs, and their limited geographical distribution may not adequately represent local variations in air quality. Consequently, the pollution levels experienced near a classroom, busy road junction, or residential neighbourhood may differ significantly from those reported by the nearest monitoring station. Furthermore, the availability and interpretation of monitoring data may present challenges for individuals without technical knowledge.

Recent developments in low-cost gas sensors, microcontrollers, and open-source data-analysis tools have created opportunities for developing localised air-quality monitoring systems. Such systems can be deployed closer to affected communities and integrated with data-processing techniques to monitor environmental conditions. Although low-cost sensors generally offer lower measurement accuracy than reference-grade instruments, their affordability makes wider deployment possible. They can therefore serve as useful tools for environmental awareness, educational applications, and preliminary monitoring of local air-quality variations.

Local environmental conditions also provide an important motivation for this study. Pune, like many rapidly developing Indian cities, experiences considerable vehicular traffic, ongoing construction activities, and seasonal variations that influence the dispersion of air pollutants. Pollution levels may increase during peak traffic hours, while temporary increases can also occur during festive periods such as Diwali, when firecracker emissions contribute to local pollution. Considering these conditions, a low-cost monitoring system that can be deployed near roads, residential areas, or educational campuses may help users observe changes in local air quality at regular intervals.

Conventional threshold-based monitoring systems generally compare current sensor readings with predefined limits to indicate the prevailing air-quality condition. However, such systems provide limited information about possible future changes. In contrast, machine-learning-based approaches can analyse previously recorded measurements and estimate future values, potentially supporting more informed decisions. The proposed system combines real-time sensor-based monitoring with an AQI-style classification approach and a regression-based prediction model. It processes live sensor readings and uses historical observations to estimate short-term changes in air quality. A simple and interpretable regression model is adopted so that its operation and coefficients can be examined and understood, while allowing the prediction process to run on an ordinary laptop without requiring a cloud-based backend.

The main contributions of this work are as follows:

- **Low-cost sensing node:**
  Development of an Arduino Uno-based monitoring node integrating an MQ135 gas sensor and a DHT11 temperature sensor, with sensor warm-up, multi-sample averaging, and an indicative PPM-to-AQI conversion process to obtain air-quality readings.
- **Real-time monitoring and visualisation:**
  Integration of the sensing node with a local software application to display air-quality readings, temperature measurements, and their variations over time.
- **Short-term air-quality prediction:**
  Application of a regression-based prediction model to analyse recorded measurements and estimate future air-quality trends.
- **Local and interpretable processing:**
  Implementation of the monitoring and prediction workflow using locally operated software tools, without dependence on an external cloud service.

2. Literature Review

Government agencies establish criteria for measuring and reporting the state of the air. The CPCB National Air Quality Index [11] transforms pollutant concentrations into a single number between 0 and 500 and classifies that into six categories ranging from Good to Severe, while the United States Environmental Protection Agency presents a similar sub-index scheme [12] and the World Health Organization publish recommended concentration limits for the major pollutants [10].

These sources of information are vital references because they identify what a given AQI value means in terms of health. Their measurement networks, however, are comprised of expensive analyzers at a limited number of sites, which limits the spatial detail and renders them ineffective for personal or hyper-local awareness.

Numerous researchers have engaged in the work of air-pollution monitoring because of the increasing deterioration of the air in numerous cities. Earlier platforms were heavily reliant on large monitoring stations, which are expensive and cover few locations, so contemporary papers propose low-cost gas sensors such as the MQ135 connected to an Arduino or NodeMCU for real-time measurement [1], [8].

These systems allow users to monitor the air at any time and notify them when pollution peaks beyond a safe limit. Common findings from these studies are that traffic, industrialization, increasing population, and fuel burning are the major contributors to pollution, and that pollution normally peaks during peak traffic hours and dips slightly at night and the early morning.

The studies also identify limitations with such systems. Low-cost sensors may show minor fluctuations, require frequent calibrations, and consume considerable power, and environmental conditions such as humidity and temperature influence their measurements [13]. Nevertheless, the majority of the reviewed papers conclude that such systems are highly valuable for awareness-based monitoring and general environmental research.

Other papers deal with IoT-based smart monitoring platforms, wherein the sensor data is uploaded to online servers or cloud services such as ThingSpeak, which are stored and processed there, and presented through dashboards or mobile applications [7]. This enables users and authorities to monitor pollution remotely from anywhere and raises public awareness, since they can easily view when the air becomes unhealthy.

The trade-off is a reliance on network connectivity and a third-party service, and the majority of these systems only display the current value without forecasting it.

A second category of work pertains to the use of machine learning to forecast future air quality from historical data. Gupta et al. [2] present a comparative study of several machine-learning models for air- quality prediction, Sitha Ram et al. [3], Kumar et al. [4], and Madan et al. [5] apply different supervised learning algorithms to pollution datasets. , and Sharma et al. [6] use automated machine learning (AutoML) to predict the AQI while reducing manual model selection.

Time-series methods such as ARIMA [14] and recurrent neural networks such as LSTM [15] are also widely used for pollution forecasting, and they can capture non-linear and long-range temporal patterns. Many of these works, however, operate on archived datasets and do not include a live sensing stage, so they do not show how a prediction can be tied to a physical, real-time monitoring node. Comparing the two paradigms shows the shift that the field has made.

A machine-learning system learns the temporal pattern of the data and can issue an early warning, but it introduces dependence on training data quality and, for complex models, reduces interpretability.

Table 1 generalizes the comparison to categories of air-quality systems. Both comparisons are qualitative and are based on the design characteristics described in the cited literature; they are not head-to-head performance benchmarks.

Table 1: The comparison to categories of air-quality systems

Feature	Government monitoring stations	Threshold-based IoT monitors	ML-only offline predictors	Proposed system
Measurement source	Reference-grade analyzers	Low-cost gas sensor	Archived datasets	Low-cost gas and temperature sensors
Cost	High	Low–Moderate	Low (software only)	Low
Spatial coverage	Sparse, fixed sites	Wherever a node is placed	Depends on dataset	Wherever a node is placed
Real-time display	Yes (online portal)	Yes	No	Yes (local web app)
Forecasting	Varies	No	Yes	Yes (short-term)
Measurement accuracy	High (calibrated)	Moderate; drift	Depends on data	Moderate; sensor-dependent
Internet dependence	Yes (to view)	Often (cloud)	No	No (local server)
Transparency	High	High	Low–Moderate	High (readable coefficients)

The comparison shows that reference stations are accurate but sparse and expensive, cloud IoT monitors are cheap but usually only reactive, and ML-only studies predict well on archived data but lack a live sensing front end. The proposed system addresses the intersection of low cost, real-time sensing, short-term forecasting, and transparency Most low-cost monitoring systems show present readings only and do not forecast them [1], [7], [8]. Many studies for prediction rely on archived datasets and are not connected to a live, physically deployed sensing node [2]–[6].

Cloud-dependent designs require continuous connectivity and third-party services, which limits use in classrooms, laboratories, and low-connectivity areas.

Few works document the conversion chain from raw sensor value to PPM to AQI to forecast in a way that can be reproduced by students and small teams.

3. Methodology/Experimental

3.1. Design Principles

The system was developed based on five principles.

(i) **Low-cost :** only low-price, widespread components are used.

(ii) **Real-time :** the data coming from sensors are constantly read, processed, and displayed.

(iii) **Transparency :** each step, from the sensor value to AQI class and forecast, is traceable and explainable.

(iv) **Locality :** the whole pipeline runs on a laptop by connecting a USB to a microcontroller, without relying on any external service.

(v) **Actionability :** the output is not only a number but also a category and a short-trend understandable by a non-expert.

3.2. System Overview

The system is structured into five layers, a sensing layer is connected to the acquisition and pre-processing layer which runs on a microcontroller called Arduino Uno. A layer about AQI computation and classification follows, and then a storage and prediction layer implemented by using the Python framework Flask is connected to a web- based presentation layer.

3.3. Materials

Hardware. An MQ135 gas sensor is used to detect the presence of air pollutants while a DHT11 sensor is used for measuring temperature. An Arduino Uno microcontroller board is the core of the system and a 16×2 LCD module is illustrated in the circuit of Fig. 1 for the local display. Software. The Arduino IDE is used to program the microcontroller, while Python is used to code the back-end part and the machine-learning model. HTML, CSS, and JavaScript are finally used to implement the web-based interface.

3.4. Methodology

Initially, we link our microcontroller to the corresponding sensors (MQ135 and DHT11). From the Arduino program, at regular intervals, AQI-related and temperature values are gathered. These raw data are transformed to concentrations and AQI units and stored temporarily. By ranges specified in the code, the quality of air is decided, and the data are graphed to study trends of the values over time and to compare the readings at different moments and locations.

The stored series is then used to predict. The readings are cleaned, divided into training and test portions, normalized, and used to fit a regression model whose output is projected over the next ten time steps and displayed on the web page next to the live values.

3.5. Data Acquisition and Pre-Processing

When the system is switched on, the sensors are first initialized and given a warm-up period for their output to stabilize; the MQ135 in particular needs time for its heating element to reach a steady operating temperature. After warm-up, the controller keeps reading the sensor data at a fixed time gap. To reduce random error, more than one reading is taken at each instant and the average is used. The averaged value is converted to PPM and then to AQI. On the laptop side, the Flask application receives the temperature and AQI values over the serial connection, appends them to a CSV file together with a timestamp, and removes empty, invalid, duplicate, or physically implausible rows before the data are used for training.

4. Mathematical Formulation

4.1. From Sensor Reading to Gas Concentration

The MQ135 is a metal-oxide semiconductor sensor whose resistance decreases as the concentration of oxidizing and reducing gases in the air increases [13]. It is connected in a voltage divider with a load resistor RL. The Arduino's 10-bit analog-to-digital converter returns a value ADC between 0 and 1023, which corresponds to an output voltage

$$Vout = ADC \times Vc / 1023 \quad (1)$$

where Vc is the supply voltage (5 V). The sensor resistance is then obtained from the divider relation

$$Rs = RL \times (Vc - Vout) / Vout \quad (2)$$

and the ratio Rs/R0, where R0 is the sensor resistance in clean air measured during calibration, is converted to an estimated concentration using a power-law fit of the datasheet sensitivity curve:

$$PPM = a \times (Rs / R0)b \quad (3)$$

Here a and b are curve-fitting constants for the target gas. Because the MQ135 responds to several gases at once, the value obtained from Eq. (3) is an aggregate, equivalent-concentration indicator of air quality rather than the concentration of one specific pollutant, which is why the paper refers to the result as an AQI-style index

4.2. AQI Sub-Index and Classification

The concentration is converted into an index by piecewise-linear interpolation between breakpoints, the method used by the CPCB and the US EPA [11], [12].

For a concentration C that lies between the breakpoints BPLo and BPHi, the index is

$$I = [(IHi - ILo) / (BPHi - BPLo)] \times (C - BPLo) + ILo \quad (4)$$

where ILo and IHi are the index values corresponding to the two breakpoints. In a multi-pollutant network the reported AQI is the maximum of the individual sub- indices; with a single gas sensor the system has one sub-index, which is used directly. The resulting value is grouped into the four classes used by the implementation.

Table 2 shows the six-category CPCB scale [11] that serves as the reference, and Table 3 gives the indicative correspondence with the four classes of this system.

Table 2: *The six-category CPCB scale*

AQI category	AQI range	Associated health impact
Good	0–50	Minimal impact
Satisfactory	51–100	Minor breathing discomfort to sensitive people
Moderate	101–200	Breathing discomfort to people with lung disease, asthma, and heart disease
Poor	201–300	Breathing discomfort to most people on prolonged exposure
Very Poor	301–400	Respiratory illness on prolonged exposure
Severe	401–500	Affects healthy people and seriously impacts those with existing diseases

Table 3: *The indicative correspondence with the four classes*

System class	Indicative CPCB equivalent	Suggested user message
Good	Good – Satisfactory	Air is comparatively clean; normal outdoor activity is fine
Moderate	Moderate	Sensitive people should limit prolonged outdoor exertion
Unhealthy	Poor – Very Poor	Reduce outdoor activity; consider a mask near traffic or smoke
Hazardous	Severe	Avoid exposure; stay indoors with ventilation controlled

4.3. Min-Max Normalization

Prior to training, the AQI values from the training set are normalized to the range [0, 1] to produce a stable learning process:

$$y' = (y - ymin) / (ymax - ymin) \quad (5)$$

where y is the original AQI value, y' is the normalized AQI, and ymin and ymax are the minimum and maximum AQI values in the training set. After prediction, the values are transformed back to the original scale by the transformation

$$y = y^{\wedge\prime} \times (ymax - ymin) + ymin \quad (6)$$

4.4. Air Quality Prediction Model

A multiple linear regression model is used to predict the AQI from both time and temperature:

$$\hat{A}' = a2 \cdot t + b2 \cdot T + c \quad (7)$$

where $\hat{A}'$ is the normalized predicted AQI, t is the time, $\hat{T}$ is the projected temperature at this time, and a2, b2, and c are the coefficients learned during the ordinary least squares training (that is, the minimization of the sum of squared errors between the predicted and the observed values). The normalized prediction is then transformed back to the AQI value using Eq. (6). Linear models are cheap to train, have no hyper-parameters to tune on the small dataset, and the resulting coefficients have an interpretable sign and magnitude: a positive a2 implies a rising AQI over time, and the sign of b2 indicates whether a higher temperature is associated with a higher AQI.

4.5. Future Time-Series Projection

The future timestamps are generated from the last time recorded using

$$tn = tlast + n \cdot \Delta t, n = 1, 2, \ldots, 10 \quad (8)$$

where tlast is the last recorded time and Δt is the fixed interval, which in the implementation is 2 minutes. This produces ten future prediction timestamps, that is, a 20-minute forecast horizon at the current sampling rate. Longer horizons can be obtained by increasing Δt or the number of steps, but the accuracy of a linear model is likely to drop as the horizon grows.

4.6. Evaluation Metrics

The trained model is evaluated on the 20% of the data held out for testing with three standard regression metrics. Given N test samples with observed AQI values yi and predicted AQI values ŷi, the mean absolute error, the root mean square error, and the coefficient of determination are

$$MAE = (1/N)\Sigma\, |yi - \hat{y}i| \quad (9)$$

$$RMSE = \sqrt{[(1/N)\,\Sigma\,(yi - \hat{y}i)2]}\ (10)$$

$$R2 = 1 - \Sigma\,(yi - \hat{y}i)2\ /\ \Sigma\,(yi - \bar{y})2\ (11)$$

where $\bar{y}$ is the mean of the observed values. MAE represents the average error in AQI units, is straightforward to interpret, RMSE penalizes larger errors more, and R2 reflects the amount of variance explained by the model. Additionally, a tolerance-based accuracy could be reported as a percentage of predicted values within a selected band of the observed value (in the sense that approximately 70- 75% of the predicted readings were found to be accurate in this work).

5. Implementation Details

5.1. Hardware

To make the project simple and cost-effective, the minimum but effective hardware was used. The most important part is the Arduino Uno to which all the sensors are connected and through which the readings were collected. The DHT sensor is for the temperature and the MQ135 sensor for the air-quality value. The Arduino is powered by means of its USB connection, and a laptop is used for the backend, to store readings, and to display the readings and predicted values in graphical form. The wiring of the components is shown in Fig. 1 and the components listed in Table 4.

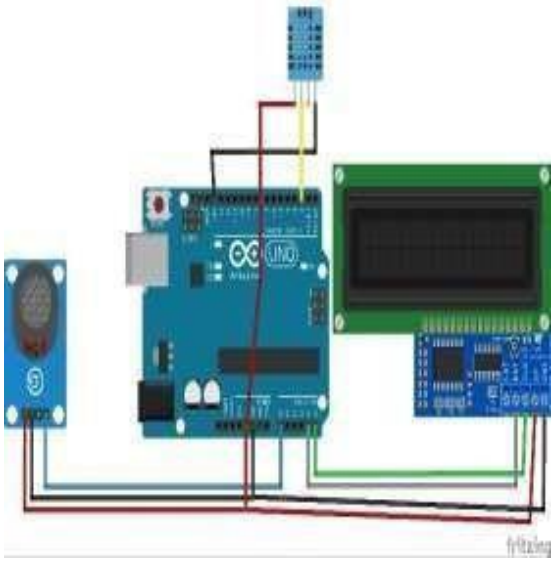


Figure 1: *Schematic diagram*

Table 4: *The list of components*

Component	Role in the system	Relative cost
Arduino Uno	Reads sensors, averages samples, computes PPM and AQI, sends data over serial	Low
MQ135 gas sensor	Detects a mixture of air pollutants (aggregate air-quality indicator)	Low
DHT11 sensor	Measures ambient temperature (and relative humidity)	Low
16×2 LCD (I2C)	Optional local display of the current reading	Low
USB cable and laptop	Power supply, serial link, backend, storage, and web dashboard	Existing

5.2. Software Libraries

Different Python libraries were used to build the prediction pipeline and to create the website summarised in the following Table 5. Flask allows developing an online application where the web page can communicate with the python code. In particular, Flask-CORS is needed to manage the different origins data request from browser that forbid such access in a default manner. Pandas, NumPy and Matplotlib are used to store, manage, calculate and visualise the data. Instead, Scikit- Learn [16] contains the Linear Regression model and the Min-Max normaliser used in the project. Finally, the graphs shown in the webpage to visualise the comparison between real and predicted values are produced using Matplotlib.

Table 5: *Different Python libraries*

Library	Purpose in the system
Flask	Backend server; connects the web page with the Python prediction code
Flask-CORS	Prevents the browser from blocking data requests between page and server
Pandas	Loading, cleaning, and arranging the CSV dataset
NumPy	Numerical operations on arrays and datasets
Scikit-Learn	Linear Regression model, Min-Max scaling, and evaluation metrics
Matplotlib	Real-time graphs of current and predicted AQI

5.3. Web Dashboard

The web-page acts as the portal for the system-user interface. The page automatically refreshes itself, displaying the current temperature, the current AQI value, the air-quality category, the ten predicted AQI values in text-form, and plots of the measured and predicted AQI over time. Showing the information in two different formats provides ease of access, and allows one to more readily identify trends in the data, such as an increase or decrease of the AQI-value. As the server is run on the same laptop as the Arduino is connected to, one can also open the dashboard on any other device connected to the local network.

5.4. Dataset

The dataset is a CSV file named air.csv in which all sensor readings are added. It includes the time, the temperature, and the AQI value, as given in Table 6, and it is constantly growing to see the pattern changes and study them. Before the training process, the file is cleaned to remove empty and incorrect values. Since it is needed to have a good level of accuracy, about 80% of the data is used for training the model and the rest 20% for testing. Then, the AQI readings are scaled to the 0-1 range by using Min-Max scaling and are returned to their normal values after predicting the results. Although the dataset may seem small, it is enough for learning and showing how the prediction system works, and it can be enriched with more variables like humidity or wind speed to improve accuracy.

Table 6: *Time, temperature, and AQI value*

Field	Type / unit	Description
Time	Timestamp	Time at which the averaged reading was recorded
Temperature	Numeric, °C	Ambient temperature from the DHT sensor
AQI value	Numeric	AQI computed from the averaged MQ135 reading

5.5. Data Flow

A full cycle goes like this: Arduino samples each of sensors multiple times, averages the samples, calculates PPM and AQI, pushes the values over serial to the Flask app, which appends a row to air.csv, then reprocesses the clean data (re-scales and re-fits the regression model), which makes the next ten values, and the server sends back the current reading along with its category, the forecast, and the plot to the web page. It is a continuous loop until the user terminates the application, so the forecast is perpetually updated from the latest values.

6. Performance Evaluation

The experiment tackles three questions: (i) can the node continually gather, log, and present readings reliably and consistently; (ii) do the readings react sensibly to changes in the environment such as location, time of day, or nearby pollution sources; and (iii) does the regression model track the true pattern of the data closely enough to be a meaningful awareness tool. The experiment acknowledges at the beginning that the sensor is a low- cost device, and the experiments test qualitative sensitivity and the ability to perform short-term forecasts; they do not attempt to verify the accuracy of the measurements against a reference instrument.

6.1. Test scenarios

The system was tested in an indoor room and outdoor environment with vehicle traffic movement. The sensor was calibrated for stabilization before any measurement sequence. Measurement sequences consisted of readings in the morning, in the afternoon, and in the evening. The system's functioning was also tested near smoke and incense to observe the system's response to a local point source that is created suddenly. Table 7 represents scenarios and observed results.

Table 7: *Observed Results*

Scenario	Condition	Observed behavior
Outdoor, morning	Low traffic	AQI mostly low; air comparatively cleaner
Outdoor, afternoon	Traffic increasing	AQI rises as vehicle activity increases
Outdoor, evening near crowded or vehicle areas	Peak traffic	AQI sometimes reached the Unhealthy range

Scenario	Condition	Observed behavior
Indoor room	Closed space, normal use	Usually lower than outdoor, but not always in the Good range
Near smoke or incense	Sudden local source	Sharp, quick rise in AQI, showing high sensitivity

6.2. Observed Air-Quality Patterns

During mornings the Air Quality Index (AQI) was generally low, which means that the air was comparatively clean. However, as the day progressed and the traffic on roads intensified, the AQI also increased. It was particularly evident during evenings, in the regions with high traffic, that the AQI reached unhealthy levels. Thus, it can be concluded that human activities, such as transportation and combustion of fossil fuels, are the main contributors to air pollution, which corroborates the findings of the literature [1], [8]. The AQI readings indoors were generally lower than those outdoors, but they were still mostly not in the Good category, which implies that the indoor air is not exempt from pollution. The sensor reacted almost immediately to the presence of incense or smoke by a sudden increase in AQI, which suggests that it is sensitive enough to register small-scale variations in the air-quality.

Fig. 2 shows the weekly AQI readings recorded by the system. The values rise from roughly 80 at the start of the week to a peak of around 165 near the end of the working week, and then fall back to about 100, illustrating the kind of multi-day variation that the dashboard graphs make visible to a user.

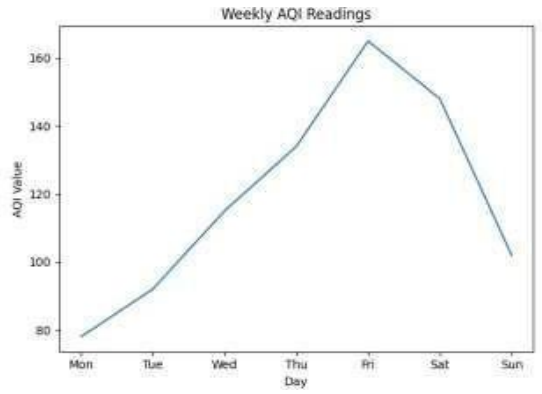


Figure 2: *Weekly AQI readings*

6.3. Prediction Behaviour

Besides estimating the current air quality, we use the stored readings to predict future AQI values. The stored data is used to observe the trend of the pollution over time, and from it, the model estimates how AQI will behave for the next intervals. This allows us to understand not only the present situation but also the expected evolution. We observed that, when the pollution level was already high in the afternoon and evening, the predicted AQI also remained on the higher side, and when readings were low during the early morning, the predicted values also stay in the safer range. This proves that our model is following the real pattern of the pollution reasonably well. Occasionally, an excessive pollution burst or a sudden temperature rise caused errors in the prediction, which is expected from a linear model that cannot anticipate abrupt events.

6.4. Verification Methodology and Results

A couple of verification methods have been used. First, functional verification: we run the program against multiple datasets to find errors in reading, cleaning, or predicting data and also visually check the dashboard to confirm that current values, categories, forecasts, and graphs are shown consistently. Second, a hold-out evaluation: the model is trained with 80% of the cleaned data and tested with the remaining 20%, and Section 4.6 metrics are computed on the test part.

Table 8: *Summary of Evaluation Results.*

Measure	Definition / basis	Result
Tolerance-based prediction accuracy	Share of predictions judged accurate against observed readings	Approx. 70–75%
Continuous operation	System run time without failure during tests	Stable for long durations
MAE (AQI units)	Eq. (9) on the 20% test split	to be recorded
RMSE (AQI units)	Eq. (10) on the 20% test split	to be recorded
R^2	Eq. (11) on the 20% test split	to be recorded

6.5. Factors Affecting Accuracy

The readings accuracy fluctuates sometime due to the sensor condition and its environment. The sensor had been brought close to the smoky fields and exposed to wind where the values had slightly fluctuated.

Multiple reasons could be leading to this inaccuracy. Cross-sensitivity: MQ135 reacts to a mixture of gases making it impossible to identify one pollution source from another. Temperature and humidity: the resistance of metal-oxide sensors varies with temperature and humidity in the air, thus differentiating the reading despite identical air content [13]. Warm-up and drift: taking readings before the heater has stabilized and slow aging process of the sensing element introduce errors unless the sensor has been periodically recalibrated. Airflow: wind and sensor location plays an essential role in determining the rate at which the sensing surface is exposed to new air. Since the design had hardware stability, the system was able to keep running long enough and detected the pollution levels in different locations regardless of the pollution status.

7. Discussion and Advantages

7.1. Core Advantages

Low cost and simplicity. The complete node is made of cheap material that is easily available and programmable; thus, the system is ideal for students, schools and small institutions. Real-time awareness. The readings and predictions are continuously updated ensuring that a user gets real-time information thus they are able to monitor pollution rise instead of waiting for the results.

Early warning through prediction. Because the system projects the measurements forward, individuals can be alerted to imminent threats to their health and organize outdoor activity accordingly. Local operation and privacy. As all the information stays on the local machine, no cloud account is needed, hence the system is operative offline and no location or usage data is shared with external entities. Transparency. The linear model and the conversion equations can be examined and explained, which is useful for learning and for trusting the outcomes.

7.2. Complementing Reference Stations

The proposed node is not a replacement to the calibrated stations that form the backbone of the official AQI reporting. On the contrary, its value is complementary: a network of inexpensive nodes would highlight local variations, such as between a busy junction and a residential lane or an indoor and an outdoor space, invisible to sparsely placed fixed stations. Regular co-location of a node with a reference station would calibrate the low-cost measurements to trusted values, which is the standard approach for enhancing such sensors' reliability.

7.3. Limitations

Sensor accuracy. The MQ135 provides an aggregate, temperature- and humidity-sensitive reading and is not a substitute for reference instruments; it does not measure PM2.5 or PM10 directly.

Simple prediction model. A linear model cannot capture non-linear dynamics or sudden events and has reduced accuracy for longer horizons; about 70–75% of readings were found accurate.

Small dataset. The dataset is limited in size and duration and lacks meteorological variables such as humidity, wind speed, and rainfall.

Local storage. The data is stored in a local CSV file, which limits scalability, long-term history, and access from other devices.

Single node. One node describes one location, so the system does not yet provide area-wide coverage.

7.4. Threats to Validity

The results should be interpreted with the following in mind. The evaluation was performed in a small number of locations over a limited period, so the observed trends may not hold for other seasons or cities. The AQI is calculated from a single aggregate sensor rather than from the several pollutants used in official

indices. The forecasting accuracy figure relies on the tolerance band used to determine a prediction as accurate. Finally, no simultaneous reference- grade measurement was available for comparison, so the absolute correctness of the AQI values could not be established.

7.5. Ethical and Social Considerations

The system is envisioned as an awareness and education tool. Messages should refrain from diagnostic or alarming language and indicate clearly that the values are indicative.

Wrongly high readings could cause unwarranted anxiety and wrongly low readings could give false reassurance, so the accuracy limits should be communicated to the users. On the plus side, cheap monitoring could democratize air-quality information for communities without a nearby official station and incentivize more environmentally conscious behavior.

8. Future Scope and Extensions

There's a good scope ahead for enhancing this project. PM2.5 and PM10 sensing. Fine particulate matter is among the pollutants most tightly associated with respiratory and cardiovascular harm and is the primary driver of AQI values in many Indian cities, despite the MQ135's inability to measure it. A laser-scattering dust sensor like the PMS5003 or the SDS011 class of sensors could directly measure PM2.5 and PM10, align it more closely the CPCB definition of AQI, and enable the calculating of a true maximum-of-sub-indices AQI. It would also make it possible to use the MQ135 to its fullest potential, namely as a gas pollutant indicator.

Cloud storage and dashboards. Replacing the local CSV file with a cloud service like Firebase or ThingSpeak could offer durable storage, remote access from any device, and a longer history for training. A hybrid design could preserve the local mode as a fallback to ensure that the system still works even without internet access.

Mobile application and push notifications. A mobile app could let people check the pollution level at any time, and push notifications could notify users of an emergency (e.g., when the forecast crosses the Unhealthy or Hazardous threshold), enabling them to take precautions when conditions worsen.

Multi-node deployment. Multiple identical nodes placed at different locations, reporting to a central dashboard, could offer coverage at a city-wide or campus-wide scale, enable spatial comparison and mapping of pollution levels, and support public- awareness and safety campaigns.

Advanced predictive models. LSTM networks [15] and ARIMA models [14] are the natural evolution beyond linear regression: the latter captures the autocorrelation and seasonality in the series, while the former can learn non-linear, long-range dependencies and can take multiple input variables together. The linear model can be preserved as a transparent baseline against which the improvement of these models can be measured with the metrics of Section 4.6.

Richer inputs and better calibration. Considering humidity, wind speed, and rainfall as model features, applying temperature and humidity compensation to the MQ135 readings, and occasionally co-locating the node with a reference station for calibration would increase accuracy and reduce drift. Utilizing a larger set of data gathered throughout seasons, including festivals, would also allow the model to better understand recurring patterns.

9. Conclusion

This paper presented the Real-Time Air Quality Detection and Prediction System, developed using simple programming techniques and low-cost hardware. An Arduino Uno paired with an MQ135 gas sensor and a DHT11 temperature sensor collects the data, which are converted into estimated PPM and AQI-style values, classified, logged in a CSV file, normalized, and processed using a multiple linear regression model to predict AQI values for the next ten time steps. The current readings, air-quality categories, forecasts, and graphical representations are displayed through a Flask-based web dashboard. Tests conducted in indoor, outdoor, and smoke-exposed environments demonstrated that the

sensor readings respond to variations in environmental conditions and local pollution sources. Approximately 70–75% of the predicted readings were reported to fall within the study's accuracy tolerance, indicating the system's potential for awareness-oriented monitoring and basic short-term prediction. However, this result should be interpreted in the context of the limited dataset and the tolerance-based evaluation method.

The main advantages of the proposed system include its low cost, real-time monitoring capability, short-term prediction, local data processing, and transparency. Its main limitations include the aggregate and environment-sensitive response of the MQ135 sensor, the simplicity of the linear regression model, the limited size of the local dataset, and the use of a single monitoring node. These limitations indicate several directions for future development, including the integration of PM2.5 and PM10 sensors, cloud-based data storage, mobile notifications, multi-node deployment, and advanced prediction models such as LSTM and ARIMA. With these improvements, the platform could serve as an affordable tool for increasing public awareness of air pollution and supporting more informed decisions about environmental conditions.

References

[1] Kgoputjo Simon Elvis Phala, Anuj Kumar, and Gerhard P. Hancke, "Air Quality Monitoring System," *IEEE Sensors Journal*, vol. 16, no. 12, June 15, 2016.

[2] Shikha Gupta, Armaan Moledina, Soham Athavale, Sarang Gajare, and Mihir Kate, "Air Quality Prediction Using Machine Learning: A Comparative Study," *IEEE*, March 4, 2024.

[3] M. Sitha Ram, Chintamreddy Reshmasri, Shaik Shahila, and Juluru Venkata Pavan Saketh, "Air Quality Prediction using Machine Learning Algorithm," *IEEE*, April 25, 2023.

[4] Vidit Kumar, Sparsh Singh, Zaid Ahmed, and Nikita Verma, "Air Pollution Prediction using Machine Learning Algorithms," vol. 11, no. 12, December 2022.

[5] Tanisha Madan, Shrddha Sagar, and Deepali Virmani, "Air Quality Prediction using Machine Learning Algorithms," *International Conference on Advances in Computing, Communication Control and Networking*.

[6] Meghna Sharma, Eishita Gupta, and D. Viji, "Air Quality Index (AQI) Prediction using Automated Machine Learning," *IEEE*, May 30, 2023.

[7] N. Anitha, S. Anne Caroline, P. Shalini, and V. Trisha, "Air Quality Monitoring System," *IEEE*, March 19, 2024.

[8] Mayank Veerani, Shivam Dwivedi, and D. Gnana Jeeba Das, "Air Quality Monitoring System," *International Conference on Advancement in Electronics & Communication Engineering (AECE 2022)*, July 14–15, 2022.

[9] World Health Organization, *WHO Global Air Quality Guidelines: Particulate Matter (PM2.5 and PM10), Ozone, Nitrogen Dioxide, Sulfur Dioxide and Carbon Monoxide*, Geneva, Switzerland, 2021.

[10] World Health Organization, *Ambient (Outdoor) Air Pollution*, WHO Fact Sheet, Geneva, Switzerland.

[11] Central Pollution Control Board, *National Air Quality Index*, Ministry of Environment, Forest and Climate Change, Government of India, New Delhi, 2014.

[12] U.S. Environmental Protection Agency, *Technical Assistance Document for the Reporting of Daily Air Quality – the Air Quality Index (AQI)*, Research Triangle Park, NC, USA.

[13] Hanwei Electronics Co., Ltd., *MQ-135 Gas Sensor Technical Data Sheet*.

[14] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*, 4th ed., Wiley, 2008.

[15] S. Hochreiter and J. Schmidhuber, "Long Short- Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

Publisher's Note & Copyright

IRJIST Journals remains neutral regarding jurisdictional claims in published maps and institutional affiliations; the views expressed are solely those of the authors.

© 2026 by the authors. Open access under the CC BY 4.0 license.