International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 2, 2026
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Machine Learning-Based Gas Risk Prediction System Using Random Forest and Django Framework

CHIRAPA ANUSHA

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

B. Suryanarayana Murthy

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

Environmental monitoring has become increasingly important due to rising pollution levels and the growing need for safety in industrial and urban environments. Hazardous gases such as methane (CH_4), carbon dioxide (CO_2), and ammonia (NH_3) pose significant risks to human health and safety when present in high concentrations. This project presents a machine learning-based gas risk prediction system developed using the Django web framework and the Random Forest algorithm.

The system is designed to predict risk levels based on gas concentration data provided through uploaded datasets or manual input. It integrates a user-friendly web interface with a backend machine learning model, enabling users to train models dynamically and perform real-time predictions. The application consists of several modules, including user authentication, dataset upload, model training, and prediction.Users can register and log in securely using Django's built-in authentication system. Once authenticated, users can upload CSV datasets containing gas concentration values such as methane, carbon dioxide, and ammonia, along with corresponding risk labels. The system processes the dataset using the Pandas library, extracts relevant features, and trains a Random Forest classifier. The trained model is then saved using Joblib for future predictions.

The prediction module allows users to input gas concentration values manually. The system loads the trained model and predicts the associated risk level, which may represent safe, moderate, or hazardous conditions. The prediction results are stored in the database for future analysis and tracking.The Random Forest algorithm is chosen due to its robustness, ability to handle nonlinear data, and high accuracy in classification tasks. It works by constructing multiple decision trees and combining their outputs to improve prediction reliability.This system provides an efficient and scalable solution for gas risk assessment, particularly useful in industries, laboratories, and environmental monitoring systems. Compared to traditional threshold-based systems, the proposed approach leverages data-driven insights for improved accuracy and adaptability.Overall, the project demonstrates the integration of machine learning with web technologies to create an

intelligent and practical solution for real-time environmental risk prediction. Future enhancements may include real-time sensor integration, advanced visualization dashboards, and the use of deep learning techniques for improved performance.

Keywords: Machine Learning, Random Forest, Gas Prediction, Environmental Monitoring, Django Web Application, Data Analysis, Predictive Modeling, Air Quality, Risk Assessment, IoT Data Processing

I. INTRODUCTION

Air pollution and hazardous gas emissions have become major concerns in today's world due to rapid industrialization, urbanization, and increased human activities. Gases such as methane, carbon dioxide, and ammonia are commonly found in industrial environments, agricultural settings, and urban areas. While these gases are naturally occurring, their elevated concentrations can lead to serious environmental and health issues, including respiratory problems, explosions, and climate change.Traditional gas monitoring systems rely on fixed threshold values to determine risk levels. These systems trigger alerts when gas concentrations exceed predefined limits. However, such approaches lack adaptability and fail to consider complex relationships between multiple gases. As a result, they may produce inaccurate or delayed warnings.

With the advancement of machine learning, predictive models can now analyze historical data and identify patterns that are not easily detectable using traditional methods. Machine learning algorithms can learn from data and make predictions based on multiple input features, making them highly suitable for environmental monitoring applications.This project introduces a machine learning-based gas risk prediction system implemented using the Django web framework. The system allows users to upload datasets, train models, and perform predictions through a web interface. It uses the Random Forest algorithm, a powerful ensemble learning technique known for its accuracy and reliability in classification tasks.

The application is designed to be user-friendly and efficient. Users can easily register, log in, and interact with the system without requiring technical expertise. The dataset upload feature enables dynamic model training, allowing the system to adapt to different datasets and environments. The prediction module provides real-time results, helping users make informed decisions.The integration of Django with machine learning libraries such as Scikit-learn and Pandas enables seamless data processing and model deployment. Django's robust architecture ensures scalability, security, and maintainability.

This system has practical applications in industries such as mining, chemical processing, agriculture, and environmental monitoring. It can also be integrated with IoT devices for real-time data collection and analysis.In conclusion, this project aims to provide an intelligent and flexible solution for gas risk prediction by combining machine learning

and web technologies. It addresses the limitations of traditional systems and offers a scalable platform for future enhancements.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

Gas prediction and environmental monitoring have been widely studied using various computational techniques, including statistical methods, machine learning algorithms, and deep learning models. Traditional approaches primarily relied on threshold-based detection systems, where predefined limits are used to trigger alerts. While simple, these systems lack adaptability and fail to capture complex relationships among multiple variables.Machine learning techniques have significantly improved prediction accuracy in environmental monitoring systems. Algorithms such as Decision Trees, Support Vector Machines (SVM), Naïve Bayes, and Random Forest have been widely used for classification tasks. Among these, Random Forest is particularly effective due to its ensemble nature, which reduces overfitting and improves generalization.

Recent studies have shown that Random Forest models outperform traditional methods in predicting air quality and gas-related risks. These models can handle nonlinear relationships and interactions between multiple gases, making them suitable for real-world scenarios. Additionally, they are less sensitive to noise and missing data.Deep learning approaches, such as Artificial Neural Networks (ANN) and Long Short-Term Memory (LSTM) networks, have also been explored for gas prediction. These models are capable of capturing complex temporal patterns in time-series data. However, they require large datasets, high computational power, and longer training times.

Another important aspect of modern systems is the integration of web technologies. Frameworks like Django and Flask are commonly used to deploy machine learning models as web applications. This allows users to interact with models through user-friendly interfaces and enables real-time predictions.Several research works have also focused on IoT-based gas monitoring systems, where sensors collect real-time data and transmit it to cloud-based platforms for analysis. These systems often incorporate machine learning models to improve prediction accuracy and provide early warnings.Despite these advancements, challenges remain in terms of data availability, model interpretability, and real-time processing. Many systems require continuous updates and maintenance to remain effective.

The proposed system addresses these challenges by combining a robust machine learning algorithm (Random Forest) with a scalable web framework (Django). It provides a simple yet effective solution for gas risk prediction and can be easily extended to include advanced features such as real-time data integration and visualization dashboards.

III. EXISTING SYSTEM

Existing gas monitoring systems primarily rely on hardware sensors and threshold-based alert mechanisms. These systems detect gas concentrations and trigger alarms when predefined limits are exceeded. While effective in basic scenarios, they lack intelligence and adaptability.

One major limitation of traditional systems is their inability to analyze multiple gases simultaneously and understand their combined effects. For example, the interaction between methane and ammonia may pose a higher risk than each gas individually, but threshold-based systems fail to capture such relationships.Some modern systems use statistical models or basic machine learning techniques for prediction. However, these systems often require manual configuration and lack user-friendly interfaces. Additionally, they may not support dynamic model training or real-time predictions.

Another limitation is the lack of integration with web technologies. Many existing solutions are standalone applications that do not provide remote access or data management capabilities. This restricts their usability in large-scale or distributed environments.Furthermore, advanced models such as deep learning require significant computational resources and large datasets, making them unsuitable for smaller applications.

The proposed system overcomes these limitations by providing a web-based platform with integrated machine learning capabilities. It allows dynamic dataset upload, model training, and real-time prediction, making it more flexible, scalable, and user-friendly compared to existing systems.

IV. PROPOSED METHOD

The proposed system is a machine learning-based gas risk prediction platform developed using the Django web framework and the Random Forest algorithm. The system is designed to overcome the limitations of traditional threshold-based monitoring systems by providing intelligent, data-driven predictions.The system allows users to upload datasets containing gas concentration values such as methane, carbon dioxide, and ammonia along with corresponding risk levels. Using this dataset, the system dynamically trains a Random Forest model, which learns patterns and relationships between gas concentrations and risk levels. Research shows that ensemble models like Random Forest perform well for structured environmental datasets due to their ability to handle nonlinear relationships and reduce overfitting .

Once trained, the model is stored and used for real-time prediction. Users can input gas values manually, and the system predicts the associated risk level instantly. The prediction results are stored in the database, enabling future analysis and tracking.The system also includes authentication features such as user registration and login to ensure secure access. A dashboard provides a centralized interface for uploading datasets, training models, and making predictions.

Unlike traditional systems, the proposed solution adapts to different datasets and environments, making it flexible and scalable. It can be extended to integrate IoT sensors for real-time gas monitoring and automated predictions.Overall, the system provides an efficient, accurate, and user-friendly solution for gas risk prediction, making it suitable for industrial safety, environmental monitoring, and smart city applications.

V. IMPLEMENTATION

The implementation of the gas risk prediction system is carried out using the Django web framework combined with machine learning libraries such as Scikit-learn, Pandas, and Joblib.The system follows a modular approach, where each functionality is implemented as a separate component. The user authentication module is implemented using Django's built-in authentication system, which provides secure registration, login, and logout features. This ensures that only authorized users can access the system.The dataset upload module allows users to upload CSV files containing gas concentration data. The uploaded file is processed using the Pandas library, which reads the dataset and extracts relevant features such as methane, carbon dioxide, and ammonia levels. The target variable, representing risk levels, is also extracted.

The machine learning module uses the Random Forest classifier from Scikit-learn. The model is trained using the extracted dataset, where input features (gas concentrations) are mapped to output labels (risk levels). Random Forest is particularly effective because it builds multiple decision trees and combines their outputs to improve prediction accuracy and reduce overfitting .After training, the model is saved using Joblib, allowing it to be reused without retraining. This improves efficiency and reduces computational overhead.

The prediction module enables users to input gas values manually through a web form. The system loads the trained model and predicts the risk level based on the input values. The prediction is displayed to the user and stored in the database using Django models.The application uses Django templates for the front-end interface, providing a simple and user-friendly experience. Messages and notifications are displayed using Django's messaging framework.

The system also ensures proper data handling and validation to prevent errors during model training and prediction. Overall, the implementation is efficient, scalable, and easy to maintain.

VI. ALGORITHMS

Random Forest Training Algorithm

- Input: Dataset with features (methane, CO_2, ammonia) and target (risk)
- Steps:
    1. Load dataset using Pandas
    2. Split dataset into features (X) and labels (y)
    3. Initialize Random Forest classifier

4. Train model using X and y
5. Save model using Joblib

- Output: Trained model

Random Forest works by creating multiple decision trees and aggregating their results, improving accuracy and reducing variance.

2. Prediction Algorithm

- Input: Gas values (methane, CO_2, ammonia)
- Steps:

1. Load trained model
2. Pass input values to model
3. Predict risk level
4. Store result in database

- Output: Predicted risk

3. Data Processing Algorithm

- Input: CSV file
- Steps:

1. Read file using Pandas
2. Extract required columns
3. Clean missing or invalid values
4. Prepare dataset for training

- Output: Clean dataset

4. Authentication Algorithm

- Input: User credentials
- Steps:

1. Validate user data
2. Authenticate using Django
3. Redirect to dashboard

- Output: Access granted/denied

These algorithms ensure efficient data processing and accurate predictions. Studies confirm that Random Forest performs well in environmental prediction tasks due to its robustness and interpretability .

VII. SYSTEM DESIGN

The system design follows the Django Model-View-Template (MVT) architecture, ensuring modularity, scalability, and maintainability.

1. Architecture Overview

The system consists of three layers:

- **Presentation Layer (Templates):**
  Handles user interaction through web pages such as login, dashboard, upload, and prediction forms.
- **Application Layer (Views):**
  Contains business logic for dataset processing, model training, and prediction.
- **Data Layer (Models):**
  Manages database operations for storing user data, uploaded files, and prediction results.

2. Module Design

a) User Module

Handles registration, login, and session management using Django authentication.

b) Data Upload Module

Allows users to upload datasets for training the model.

c) Machine Learning Module

Trains the Random Forest model using uploaded data.

d) Prediction Module

Generates predictions based on user input.

e) Result Storage Module

Stores prediction results for future reference.

3. Data Flow

1. User logs into system

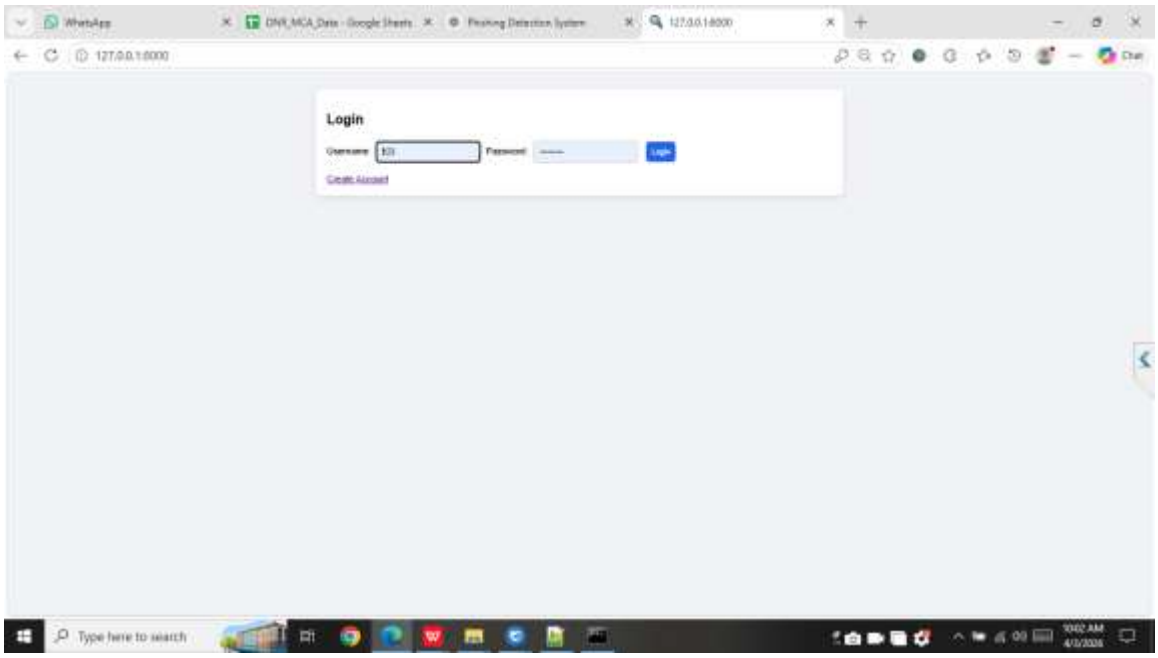
2. Uploads dataset
3. System processes data
4. Model is trained and stored
5. User inputs gas values
6. System predicts risk
7. Results are stored and displayed

4. Design Advantages

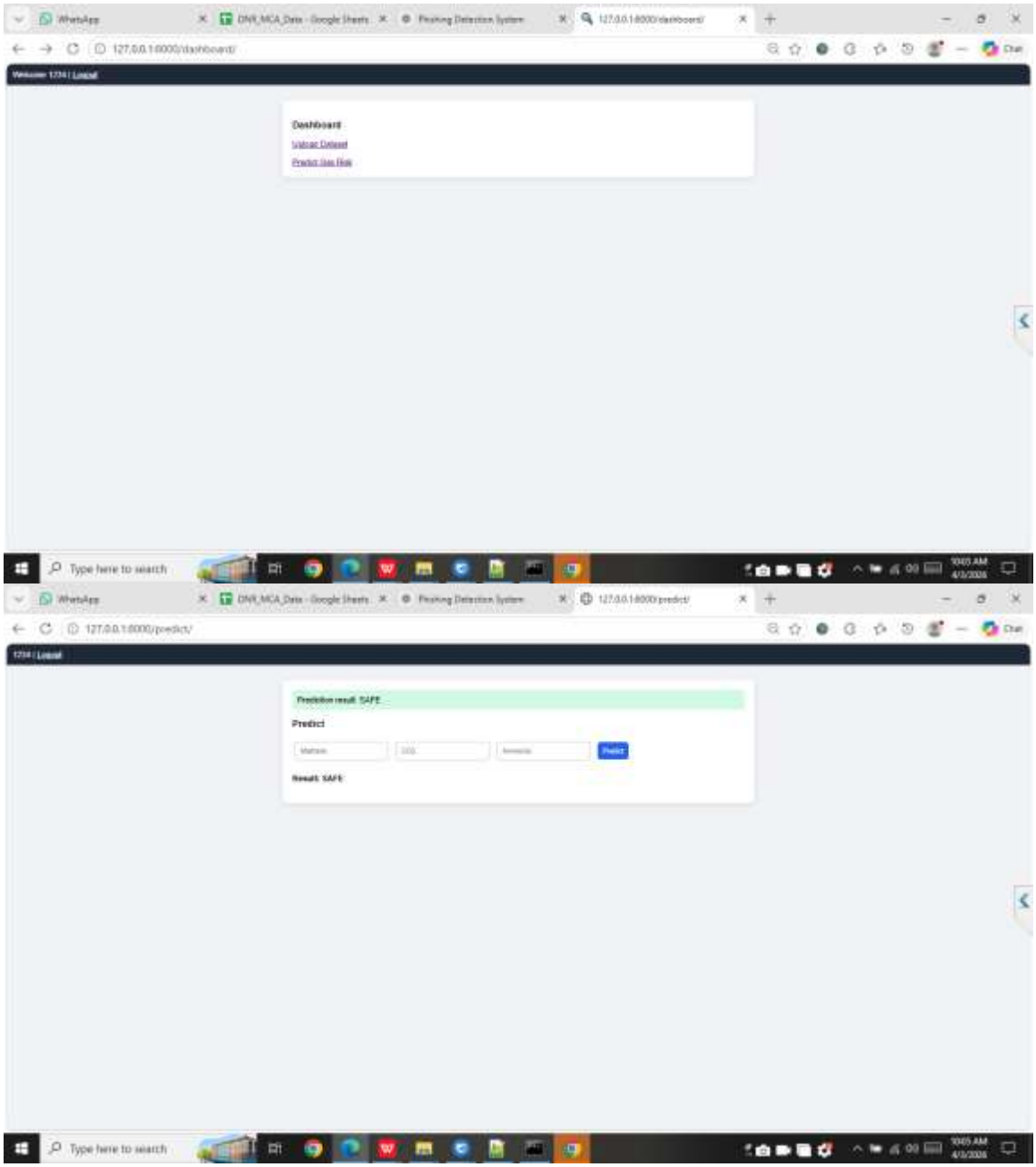
- Modular architecture
- Easy integration with ML models
- Scalable and maintainable
- Supports real-time prediction

Modern systems emphasize integrating machine learning with web frameworks for real-time environmental monitoring and decision-making .

SYSTEM DESIGN IMAGES



International Journal of Engineering Research and Science & Technology



VIII. CONCLUSION

The machine learning-based gas risk prediction system provides an efficient and intelligent solution for monitoring hazardous gas levels. By leveraging the Random Forest algorithm and Django framework, the system enables accurate and real-time risk prediction.The system successfully integrates data processing, model training, and prediction into a single platform. It allows users to upload datasets, train models dynamically, and obtain predictions instantly. This flexibility makes it suitable for various applications, including industrial safety, environmental monitoring, and smart city solutions.

Compared to traditional threshold-based systems, the proposed system offers improved accuracy by analyzing complex relationships between multiple gas variables. Research highlights that machine learning models significantly enhance prediction capabilities in air quality monitoring systems .The system is also scalable and can be extended with advanced features such as real-time sensor integration, visualization dashboards, and deep learning models. Additionally, it provides a user-friendly interface, making it accessible to non-technical users.

Future work can focus on improving model performance, integrating IoT devices, and implementing advanced analytics for better decision-making.In conclusion, this project demonstrates the potential of combining machine learning and web technologies to create practical and impactful solutions for environmental safety and risk prediction.

REFERENCES

1. · Chen et al., “Air Pollution Prediction Using Random Forest,” Sustainability, 2024
2. · Rajesh et al., “ML Framework for Real-Time Air Quality Prediction,” Scientific Reports, 2025
3. · Karmoude et al., “ML for Air Quality Prediction Review,” ScienceDirect, 2025
4. · Amman-Zarqa Study, “Random Forest for Air Quality,” 2024
5. · AQI Comparative Study (RF vs ANN vs LSTM), IEEE, 2024
6. · Machine Learning Review on Air Quality, 2025
7. · Hybrid RF + ARIMA AQI Model, 2025
8. · SSRN, “Air Quality Forecasting ML,” 2024
9. · Scientific Reports, “Hybrid Deep Learning Air Quality,” 2025
10. · Méndez et al., “Survey on ML Air Prediction,” 2023
11. · Tang et al., “ML Modeling for Air Quality,” 2024
12. · Özüpak et al., “Ensemble Air Prediction Models,” 2025
13. · Singh et al., “ML for PM2.5 Prediction,” 2025
14. · Anggraini et al., “Global AQI Prediction ML,” 2024
15. · ArXiv, “AQI Prediction Using ML Models,” 2024