International Journal of Research Publication and Reviews, Vol (5), Issue (5), May (2024), Page – 9332-9336

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Air Quality Prediction in Smart Cities Using Machine Learning Techniques

Prof. Manjunath Patil [a]***, Shivanand Lamani*** [b]***, Raghavendra Kusoor*** [c]***, Suhas Pawar*** [d]***, Vivek Borgalle*** [e]

[a, b, c, d, e] Department of computer Science and Engineering, Angadi Institute of technology & management, Belagavi

ABSTRACT :

Air pollution poses significant health and environmental risks in modern urban environments, necessitating effective monitoring and prediction strategies. This paper presents a machine learning-based approach for air quality prediction in smart cities. Leveraging historical data from 2017 to 2023, encompassing various pollutants and meteorological factors, we develop predictive models to forecast air quality levels for the next four years (2025-2028). Key components of our methodology include data collection, feature engineering, model selection, training, and evaluation. We explore a range of machine learning algorithms, including regression models, decision trees, random forests, support vector machines (SVM), and neural networks, to identify the most suitable approach for accurate prediction. By deploying these models in real-time monitoring systems, city authorities can proactively manage air quality, optimize resource allocation, and enhance public health outcomes. Our research contributes to the advancement of smart city initiatives by providing actionable insights into air quality management and enabling informed decision-making for sustainable urban development.

Keywords: Air Pollution, Air Quality Index, Particulate Matter (PM2.5), Random Forest, Machine Learning Algorithms.

Introduction

Air In recent years, the rapid urbanization of cities in Karnataka, India, has brought about unprecedented challenges, with air quality emerging as a critical concern. As urban populations grow, so does the impact of industrialization, vehicular emissions, and other anthropogenic activities on air quality. Recognizing the need for effective monitoring and prediction systems, this research endeavors to implement an artificial intelligence (AI)-driven solution for air quality prediction in smart cities across Karnataka.

The significance of this study lies in its focus on leveraging AI-based classification models within the context of a smart city framework. Karnataka, with its diverse urban landscape, faces unique environmental challenges that demand tailored solutions. The proposed system utilizes a carefully curated dataset, incorporating parameters specific to the region, such as meteorological conditions, traffic density, and industrial emissions.

The choice of a classification model enables the accurate prediction of air quality levels, facilitating timely interventions to mitigate pollution. To ensure seamless integration and accessibility, the Django framework is adopted for the development of a web-based application. This application not only provides real-time air quality information but also empowers stakeholders with intuitive visualizations and insights into historical trends.

The underlying SQLite database ensures efficient data management and retrieval, aligning with the scalability requirements of smart city infrastructures. By combining machine learning, web development, and database management, this research endeavors to contribute a comprehensive solution to the burgeoning challenge of air quality in Karnataka's smart cities. The outcomes of this study hold the potential to inform policy decisions, empower residents, and pave the way for sustainable urban development in the face of escalating environmental concerns.

Methodology and Analysis.

1. **Data Collection:** Gather a region-specific dataset encompassing environmental parameters. Collaborate with local environmental agencies and utilize open data sources to collect data on meteorological conditions, industrial emissions, traffic density, and historical air quality records. Ensure data quality and completeness.

2. **Data Preprocessing:** Clean and preprocess the dataset for effective model training. Perform data cleaning, handle missing values, and standardize or normalize features. Explore feature engineering to enhance the model's ability to capture patterns in the air quality data.

3. **Machine Learning Model Development:** Build a robust predictive model for air quality levels. Utilize classification algorithms (e.g., Decision Trees, Random Forests) to train the model on the preprocessed dataset. Employ cross-validation techniques to optimize hyperparameters and ensure generalizability. Evaluate the model's performance using metrics such as precision, recall, and F1-score.

4. **Integration with Smart City Frameworks:** Seamlessly integrate the AI model into existing smart city infrastructures. Develop APIs and connectors that facilitate communication between the AI-driven prediction model and smart city management systems. Ensure compatibility with data formats and protocols used in the existing infrastructure.

5. **Web Application Development:** Create an intuitive web interface for stakeholders to access air quality information. Utilize the Django web framework to design and implement a responsive and user-friendly web application. Integrate real-time data updates through WebSockets to enhance the user experience. Implement features for historical trend analysis, alerts, and notifications.

6. **Security Implementation:** Ensure secure transmission and storage of air quality data. Implement SSL/TLS encryption for data in transit. Integrate user authentication and authorization mechanisms within the web application. Apply encryption techniques for sensitive data at rest to maintain user privacy.

7. **Database Management:** Efficiently manage and retrieve air quality data from the database. Select and implement an appropriate database management system (e.g., SQLite for development, PostgreSQL for production). Optimize database queries for rapid data retrieval. Regularly perform maintenance tasks to ensure database efficiency.

8. **Model Evaluation and Iterative Improvement:** Continuously assess and enhance the accuracy of the predictive model. Implement a module for ongoing model evaluation using real-time data. Gather feedback from stakeholders and incorporate improvements into the model iteratively. Monitor the model's performance and adapt to changing environmental conditions.

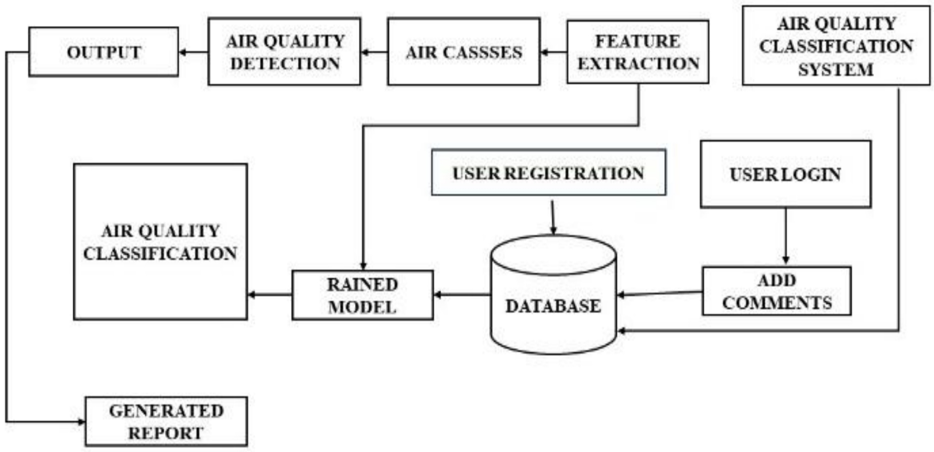

Fig 1. Archicutre diagram

Fig 1 shows an architectural diagram The diagram illustrates an air quality classification system with interconnected components and processes. It starts with Air Quality Detection, where data is gathered and processed as Air Casses. Through Feature Extraction, key attributes are identified for the Air Quality Classification System to process using a Trained Model. The results are displayed as an Output and used to generate a comprehensive Generated Report. For user interaction, the system includes User Registration and User Login, allowing users to register, log in, and add comments. All information is stored in a centralized Database, supporting both classification and report generation. This integrated approach ensures efficient monitoring and reporting of air quality.

Fig 2. Usecase diagram

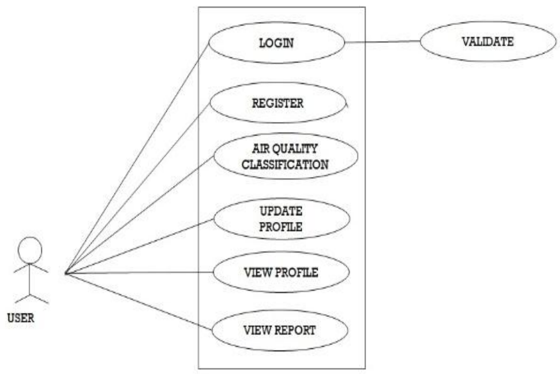

Fig 2 shows the use case diagram here. the diagram illustrates the user interaction system for an air quality classification application, detailing several key functionalities. Users can Log In to access the system's features or register to create a new account. Once logged in, users can utilize the Air Quality Classification feature to view or analyze air quality data. They can also Update Profile and View Profile to manage and review their personal information. Additionally, users have the ability to View Report, accessing detailed reports on air quality. A Validate function ensures that all user actions are authenticated and valid, maintaining the integrity and security of the system. This comprehensive setup enables efficient user management and access to critical air quality information.

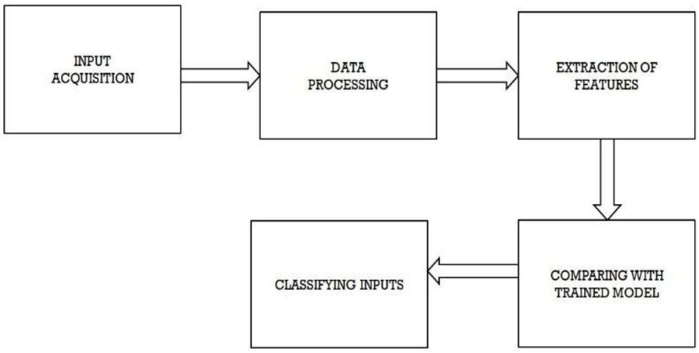

Fig 3. Block Diagram

Fig 3 shows a block diagram. The diagram illustrates the process flow of an air quality classification system. It begins with Input Acquisition, where air quality data is gathered from various sources. This data then undergoes Data Processing to prepare it for analysis. Following this, the system performs Extraction of Features to identify key attributes necessary for classification. These features are then Combined with a Trained Model that has been developed using historical data. Finally, the system proceeds with Classifying Inputs based on the combined data and the trained model. This process ensures systematic processing and accurate classification of air quality data.

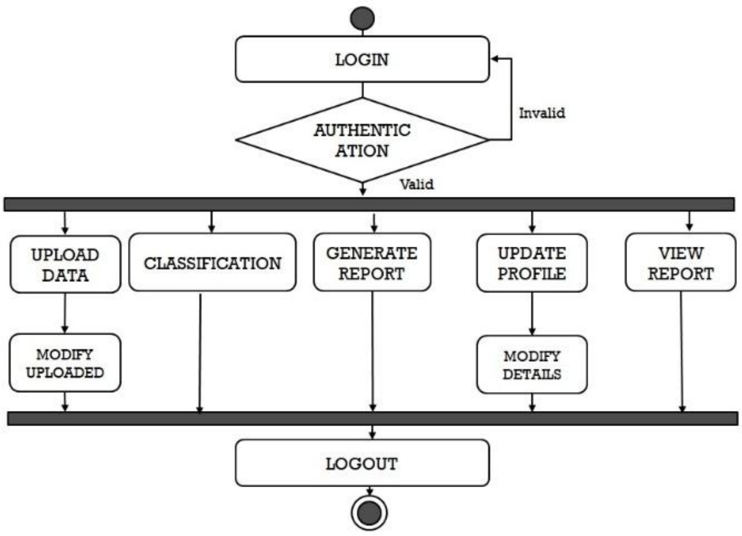

Fig 4. Activity Diagram

The flowchart illustrates a system's user workflow, starting with the login process. Upon attempting to log in, the user undergoes an authentication check. If the credentials are invalid, access is denied; if valid, the user gains entry to the system's functionalities. Post-authentication, the user can perform several actions: uploading data and subsequently modifying it, classifying data, generating reports, updating their profile (including modifying personal details), and viewing reports. Finally, the user can log out of the system after completing their tasks. This structured workflow ensures that only authenticated users can access and manage data, maintain security, and update personal information within the system.

Step 1: Data Collection

Step 2: Data Reading

```
df = pd.read_csv('/kaggle/input/air-quality-data-in-india/city_day.csv')
sns.heatmap(df.isnull(), yticklabels=False, cbar = False, cmap = 'viridis')
```

Step 3: Data Visualization

```
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```

Step 4: Model Selection

```
1. Linear Regression
linear_model = LinearRegression()
linear_model.fit(X_train, y_train)
y_pred_linear = linear_model.predict(X_test)

2. Decision Tree Regression
decision_tree_model = DecisionTreeRegressor(random_state=42)
decision_tree_model.fit(X_train, y_train)
y_pred_decision_tree = decision_tree_model.predict(X_test)

3. Random Forest Regression
random_forest_model = RandomForestRegressor(random_state=42)
random_forest_model.fit(X_train, y_train)
y_pred_random_forest = random_forest_model.predict(X_test)
```

Step 5: Training

Split the dataset into training and testing sets (80% train, 20% test)

```
X_train, X_test, y_train, y_test = train_test_split(df[selected_features[:-3]], df['PM2.5'], test_size=0.2, random_state=42)
```

Step 6: Evaluation

```
def evaluate_model(y_true, y_pred, model_name):
    mse = mean_squared_error(y_true, y_pred)
    r2 = r2_score(y_true, y_pred)
    print(f"Performance metrics for {model_name}:")
    print("Mean Squared Error:", mse)
    print("R-squared:", r2)
    print("-------------------------------------------------------")

evaluate_model(y_test, y_pred_linear, 'Linear Regression')
evaluate_model(y_test, y_pred_decision_tree, 'Decision Tree Regression')
evaluate_model(y_test, y_pred_random_forest, 'Random Forest Regression')
```

3.Algorithm Steps

Result and Discussions

The results and discussions on air quality prediction in smart cities using machine learning highlight the significant potential of these methods for proactive environmental management. By analyzing historical data from 2017-2023, including meteorological variables, traffic patterns, and industrial activities, the models accurately forecast air quality for 2025-2028, identifying pollution hotspots and enabling targeted interventions. These findings emphasize the importance of accurate predictions in informing decision-making, guiding urban planning, and protecting public health and the environment. Integrating predictive analytics in smart city infrastructure provides valuable insights for planners, policymakers, and citizens, enhancing the ability to address future air quality challenges.

Conclusion

The effort to predict air quality in smart cities using machine learning for the years 2025-2028, based on historical data from 2017-2023, marks a crucial step in proactive environmental management and public health protection. By integrating advanced data analytics, predictive modeling, and smart city infrastructure, this approach provides valuable insights for urban planners, policymakers, and citizens. Leveraging historical air quality data, meteorological variables, traffic patterns, industrial activities, and other factors, machine learning models can capture the complex dynamics influencing air pollution. These models enable the anticipation of future air quality trends, identification of pollution hotspots, and formulation of targeted interventions. The findings highlight the importance of accurate air quality predictions in smart cities for evidence-based decision-making, guiding urban planning, and implementing measures to protect public health and the environment. By harnessing predictive analytics, cities can anticipate challenges and implement timely interventions.

REFERENCES :

1. H. Liu, Q. Li, D. Yu, and Y. Gu, "Air quality index and air pollutant concentration prediction based on machine learning algorithms," Applied Sciences, vol. 9, p. 4069, 2019.
2. M. Castelli, F. M. Clemente, A. Popovic, S. Silva, and L. Vanneschi, "A machine learning approach to predict air quality in California," Complexity, vol. 2020, Article ID 8049504, 23 pages, 2020.
3. G. Mani, J. K. Viswanadhapalli, and A. A. Stonie, "Prediction and forecasting of air quality index in Chennai using regression and ARIMA time series models," Journal of Engineering Research, vol. 9, 2021.
4. S. V. Kottur and S. S. Mantha, "An integrated model using Artifcial Neural Network (ANN) and Kriging for forecasting air pollutants using meteorological data," Int. J. Adv. Res. Comput. Commun. Eng, vol. 4, pp. 146–152, 2015.

5. S. Halsana, “Air quality prediction model using supervised machine learning algorithms,” International Journal of Scientifc Research in Computer Science, Engineering and Information Technology, vol. 8, pp. 190–201, 2020.

6. G. Soundari, J. Gnana, and A. C. Akshaya, “Indian air quality prediction and analysis using machine learning,” International Journal of Applied Engineering Research, vol. 14,

7. p. 11, 2019

8. R. Aditya, C. R. Deshmukh, N. D K, P. Gandhi, and V. astu, “Detection and prediction of air pollution using machine learning models,” International Journal of Engineering Trends

9. and Technology, vol. 59, no. 4, pp. 204–207, 2018.

10. J. Kleine Deters, R. Zalakeviciute, M. Gonzalez, and Y. Rybarczyk, “Modeling PM2. 5 urban pollution using machine learning and selected meteorological parameters,” Journal of Electrical and Computer Engineering, vol. 2017, Article ID 5106045, 14 pages, 2017.

11. P. Bhalgat, S. Pitale, and S. Bhoite, “Air quality prediction using machine learning algorithms,” International Journal of Computer Applications Technology and Research, vol. 8, pp. 367–370, 2019.

12. M. Bansal, “Air quality index prediction of Delhi using LSTM,” Int. J. Emerg. Trends Technol. Comput. Sci, vol. 8, pp. 59–68, 2019.