JADIMI:
JOURNAL OF AI-DRIVEN INFORMATICS AND MANAGEMENT INFORMATION
Vol. 1 No. 1, August 2026, pp 39-44

Daily ISPU Classification in DKI Jakarta Based on Gas Pollutant Parameters using Machine Learning

Andien Swesty Dewantari [1*], Nayla Raissa Putri[1], M. Rafi Naufal Hilmy[1] , Vicky Prasetya[1], M. Pramudya Miftah[1]

[1] *Software Engineering Study Program, Politeknik Astra, Indonesia*

[*] *Corresponding author: 0920240049@polytechnic.astra.ac.id*

Abstract*—Air pollution is a persistent environmental problem in urban areas, causing serious impacts on public health and environmental quality. Growing transportation activity, industrial output, and population density lead to daily fluctuations in pollutant concentrations. The Air Pollution Standard Index (ISPU) is used as an indicator of air quality based on several pollutant gas parameters, such as PM2.5, PM10, SO_2, CO, O_3, and NO_2. Machine learning approaches can help classify air pollution levels from historical data patterns. This study examines and compares four machine learning algorithms, namely Logistic Regression, Decision Tree, Random Forest, and XGBoost, to classify the daily air pollution level in DKI Jakarta based on ISPU categories. The dataset used is DKI Jakarta air quality data consisting of 5,538 observations, evaluated under two data-splitting scenarios (70:15:15 and 80:10:10) to test model consistency. Model performance was evaluated using accuracy, precision, recall, and F1-score metrics. The test results show that the 80:10:10 scenario delivers better performance, with XGBoost achieving the highest performance and a test accuracy of 93.68%, outperforming Random Forest, Decision Tree, and Logistic Regression. Based on feature importance analysis, the O_3 and PM10 parameters contribute the most to the classification process. These findings indicate that ensemble-learning-based algorithms, particularly XGBoost with a larger training-data proportion, can effectively capture the relationships among pollutant parameters. This approach can support the development of data-driven air quality classification systems, with future research recommended to incorporate weather variables, real-time data streams, further hyperparameter tuning, and deep learning approaches to improve classification accuracy.

Keywords—air quality; ISPU; air pollution; machine learning; XGBoost; classification

Manuscript received 30 Jul. 2026; revised 30 Aug. 2026; accepted 30 Aug. 2026. Date of publication 31 Aug. 2026.

I. INTRODUCTION

Air pollution is one of the main environmental problems in large cities, including DKI Jakarta, caused by high transportation activity, industry, and population density. Long-term exposure to air pollutants can cause negative impacts on public health, ranging from respiratory disorders to cardiovascular disease [1][2]. The government uses the Air Pollution Standard Index (ISPU) as a reference to communicate air quality levels to the public. ISPU is divided into several categories: Good, Moderate, Unhealthy, Very Unhealthy, and Hazardous [4][5]. However, the ISPU classification process still faces challenges because it involves many pollutant parameters, potentially incomplete data, and the need for fast and accurate information. Therefore, an automatic approach based on machine learning is needed to help the classification process become more efficient and consistent [1].

Based on these problems, this study aims to build and compare the performance of four classification algorithms, namely Logistic Regression, Decision Tree, Random Forest, and XGBoost, in classifying the daily air pollution level in DKI Jakarta. To comprehensively test computational stability, the evaluation is carried out on two data-splitting ratio scenarios (70:15:15 and 80:10:10). The contribution of this study is to provide a comparison of the most ideal data-splitting scenario performance and

to identify the pollutant gas parameters that have the greatest influence on ISPU classification.

II. RELATED WORK

Research on air quality classification in DKI Jakarta has been widely conducted due to the high level of air pollution and the importance of ISPU information for the public. Umri, Firdaus, and Primajaya (2021) compared five classification algorithms, namely Neural Network Backpropagation, Support Vector Machine, K-Nearest Neighbors, Naive Bayes, and Decision Tree, to determine the ISPU level in DKI Jakarta [9]. That study used the main pollutant parameters CO, SO2, NO2, O3, and PM10, and evaluated the models using accuracy, kappa, and RMSE metrics. The best result was obtained by Decision Tree with an accuracy of 99.80%, a kappa of 0.996, and an RMSE of 0.039 [3][4].

These results show that algorithm selection greatly affects air quality classification performance. Decision Tree proved superior to Naive Bayes, KNN, SVM, and Neural Network because it was able to handle the complex patterns of ISPU data. This study also used 10-fold cross validation and backward-elimination feature selection to improve model quality. These findings become an important basis that machine learning approaches can be used effectively for ISPU classification in DKI Jakarta [4].

Based on these previous studies, it can be concluded that ISPU classification in DKI Jakarta has been widely studied using various machine learning algorithms. However, a broader model comparison with a more rigorous evaluation approach is still needed. Therefore, this study develops previous studies by comparing conventional models to modern ensemble algorithms under two different data-splitting scenarios to obtain a model that is genuinely resistant to overfitting.

III. MATERIALS AND METHODS

This study was conducted through several main stages: data understanding, data preprocessing (data cleaning, imputation, encoding, and data balancing), data splitting under two ratio scenarios, modeling using four classification algorithms, and model evaluation. The research stages are shown in Fig. 1.

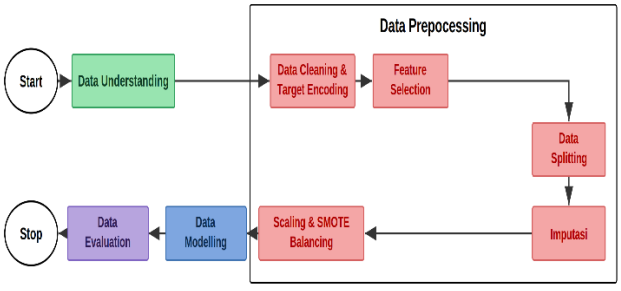


Figure. 1 Research flow diagram

A. Dataset

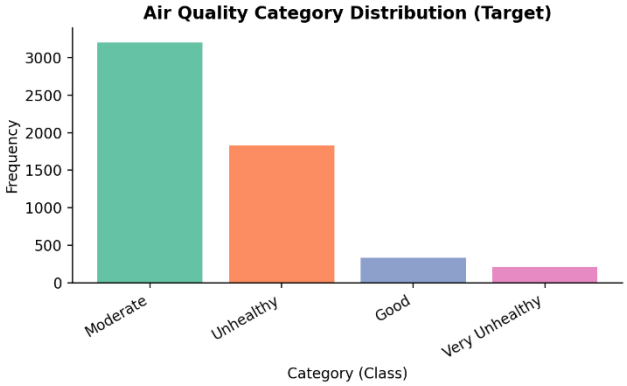
The dataset used in this study is historical daily data of the DKI Jakarta Air Pollution Standard Index (ISPU). In its initial form, this dataset has a dimension of 5,538 rows and 11 columns, consisting of the attributes date, station, six pollutant gas parameters (pm10, pm25, so2, co, o3, no2), max, critical, and category as the target label[5].

During the initial inspection, it was found that the pm25 pollutant gas parameter had an extremely high number of missing values, namely 4,022 rows of data (more than 50% of the total dataset). Therefore, the pm25 column was automatically removed from the analysis. Next, metadata features and features that could potentially cause data leakage, such as "date", "station", "max", and "critical", were also discarded. Finally, the target column was cleaned by removing data labeled "no data", leaving 5,533 rows of clean dataset (consisting of the pollutants PM10, SO2, CO, O3, and NO2) used for modeling. Table I in below dataset feature description.

TABLE I
DATASET FEATURE DESCRIPTION

Feature	Description	Unit
PM10	Particulate < 10 microns	µg/m³
SO2	Sulfur dioxide	µg/m³
CO	Carbon monoxide	µg/m³
O3	Ozone	µg/m³
NO2	Nitrogen dioxide	µg/m³
ISPU Category	Classification target label (Good, Moderate, Unhealthy, Very Unhealthy)	—

The distribution of ISPU categories in the dataset shows a fairly significant class imbalance, with 3,187 data points for the Moderate category, 1,827 for Unhealthy, 316 for Good, and 203 for Very Unhealthy, as shown in Fig 2.



Figure, 2 ISPU category distribution in the dataset

After the clean features were obtained (PM10, SO2, CO, O3, and NO2), an initial statistical validation was carried out in the form of a Spearman correlation analysis between each pure pollutant parameter and the ISPU target category, as shown in Fig. 3. This result is used as a basis for validating that the five remaining features are suitable for use as predictor variables in the modeling stage.

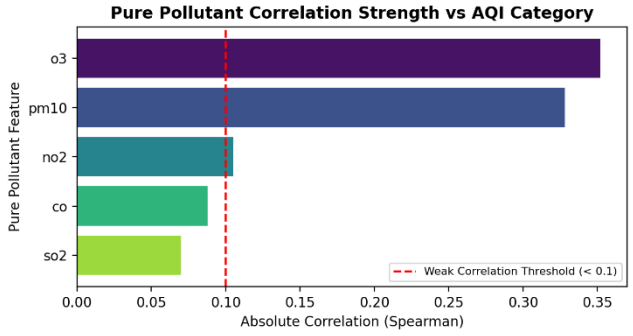


Figure. 3 Spearman correlation strength between pure pollutants and ISPU category

B. Data Preprocessing

The data preprocessing stage begins with numeric data type conversion and missing-value handling. Based on the boxplot visualization on the training data (in Fig. 4), all pollutant gas parameters show a skewed distribution and contain many extreme outliers.

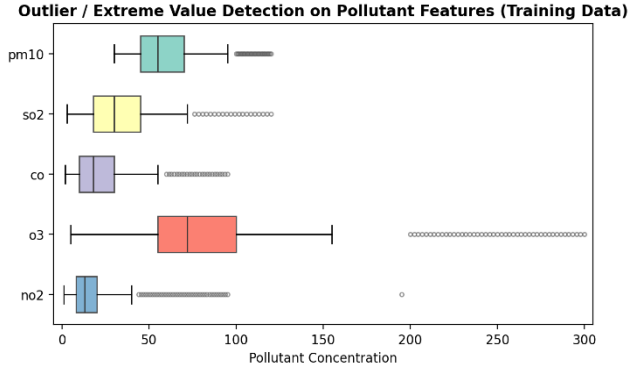


Figure. 4 Extreme outlier detection in pollutant features (training data)

Due to the presence of outliers and the skewed distribution, missing-value imputation was decided using the median value of each feature calculated from the training data, rather than the mean value, so that the imputation result is not biased by extreme values. Before imputation, the number of empty cells in the 70:15:15 scenario was recorded at 488 cells in the training data, 124 cells in the validation data, and 111 cells in the test data; while in the 80:10:10 scenario, 552 cells were recorded in the training data, 85 cells in the validation data, and 86 cells in the test data. The imputation value (median) was calculated specifically from the training data for each scenario, then also applied to the validation data and test data, so that all missing values in all three subsets were fully handled (0 remaining empty values)[6].

Next, encoding was carried out on the ISPU category labels into numeric form using the Label Encoding method, with the class mapping: Good = 0, Very Unhealthy = 1, Moderate = 2, and Unhealthy = 3. The relationship between the pollutant features and the target is visualized through a Spearman correlation heatmap in Fig. 5.

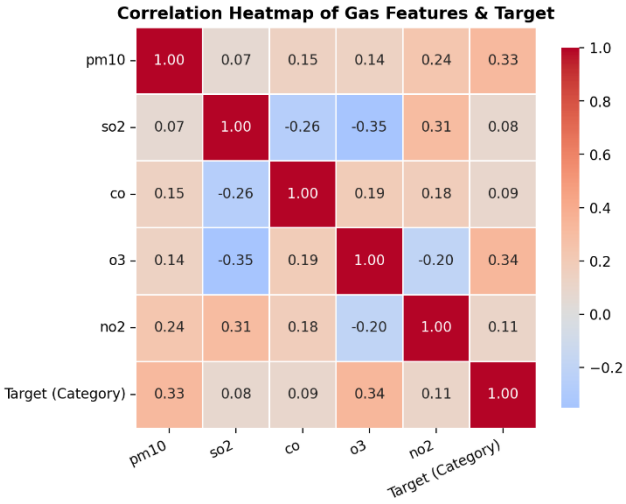


Figure. 5 Spearman correlation heatmap between gas features and target

C. Data Splitting

To comprehensively evaluate model performance and robustness, the dataset of 5,533 rows was divided into training data, validation data, and testing data. This splitting process was carried out using a stratified split technique to ensure that the proportion of each ISPU category (Good, Moderate, Unhealthy, Very Unhealthy) remained maintained in each subset.

In this study, two data-splitting ratio scenarios were compared to identify the configuration that provides the most optimal model performance, namely:

1) Scenario 70:15:15: The dataset is divided into 3,873 training data, 830 validation data, and 830 test data. This scenario provides a larger portion for evaluation, so that model testing against data variation becomes more robust.
2) Scenario 80:10:10: The dataset is divided into 4,426 training data, 553 validation data, and 554 test data. This scenario aims to provide a larger portion of learning data so that the algorithm can recognize patterns more optimally, although the amount of data for evaluation becomes smaller.

The validation data in both scenarios is used as a reference to select the best model and detect indications of overfitting during the training process. Meanwhile, the test data is used as the final evaluation on data that has never been seen by the model at all (unseen data).

Furthermore, given that the training data has a significant class imbalance dominated by the "Moderate" category, the Synthetic Minority Over-sampling Technique (SMOTE) is applied[7]. It should be noted that SMOTE is applied exclusively to the training data in both scenarios so that the number of each category becomes balanced. The validation data and test data are not included in the SMOTE process so that they continue to represent the actual data distribution conditions in the field. The training data class distribution before and after SMOTE in both scenarios is shown in Table II.

TABLE II
TRAINING DATA CLASS DISTRIBUTION BEFORE AND AFTER SMOTE

Scenario	Category	Before	After
70:15:15	Moderate	2231	2231
70:15:15	Unhealthy	1279	2231
70:15:15	Good	221	2231
70:15:15	Very Unhealthy	142	2231
80:10:10	Moderate	2549	2549
80:10:10	Unhealthy	1462	2549
80:10:10	Good	253	2549
80:10:10	Very Unhealthy	162	2549

D. *Classification Algorithms*

1) Logistic Regression: a linear classification algorithm that models the probability of a class using a logistic function, used as a baseline model with the parameters max_iter=1000 and class_weight='balanced' [8].

2) Decision Tree: an algorithm that builds a decision-tree structure based on the selection of the most significant feature in separating classes at each node, with the parameters max_depth=6, min_samples_leaf=4, and class_weight='balanced'.

3) Random Forest: a bagging-based ensemble method that combines many decision trees to improve accuracy and reduce overfitting, with the parameters n_estimators=150, max_depth=8, min_samples_leaf=2, and class_weight='balanced'.

4) XGBoost: a boosting-based ensemble method that builds a model gradually by correcting the errors of the previous model, known for its high performance on structured data, with the parameters n_estimators=100, max_depth=5, and learning_rate=0.1.

E. *Model Evaluation*

Each model was trained using the SMOTE-processed training data (after being standardized using StandardScaler), then evaluated on the validation data to select the best-performing model and detect indications of overfitting through the gap between training and validation accuracy[9]. The model with the highest combined score (average of training and validation accuracy) from both scenarios was then tested on the test data using the accuracy, precision, recall, and F1-score metrics, as well as a confusion matrix to examine the details of misclassification in each ISPU category[2].

IV. RESULTS AND DISCUSSION

A. *Pollutant Parameter Correlation Analysis*

Before evaluating the performance of the machine learning classification models, a Pearson correlation analysis was carried out to understand the level of linear relationship between the pollutant gas parameters and their relationship with the target label (ISPU category). This correlation matrix visualization is shown in Fig. 3 and Fig. 5[10].

Based on Fig. 3, it can be seen that the Particulate (PM10) and Ozone (O_3) parameters have the highest correlation with the target category and are above the weak-correlation threshold (0.1) marked by the dashed line. This statistically indicates that these two pollutants are the strongest early indicators influencing the level of air quality hazard in DKI Jakarta compared to the other parameters (SO_2, CO, and NO_2), which are relatively close to or below that threshold. This initial correlation finding is consistent with the feature importance results from the XGBoost model in the final evaluation stage [11].

B. *Model Training and Validation Results*

The four models were trained using the SMOTE-processed training data and evaluated on both the training data and validation data to detect indications of overfitting. The comparison of training and validation accuracy results is shown in Table III and Fig. 6 and Fig. 7.

TABLE III
COMPARISON OF TRAINING AND VALIDATION ACCURACY FOR THE 7-:15:15 AND 80:10:10 SCENARIOS

Scen.	Model	Train	Val.	Gap
70:15:15	Logistic Regression	0.6915	0.6855	0.0059
70:15:15	Decision Tree	0.8885	0.8759	0.0126
70:15:15	Random Forest	0.9419	0.9181	0.0238
70:15:15	XGBoost	0.9589	0.9253	0.0336
80:10:10	Logistic Regression	0.6945	0.6799	0.0146
80:10:10	Decision Tree	0.8886	0.8626	0.0260
80:10:10	Random Forest	0.9433	0.9060	0.0373
80:10:10	XGBoost	0.9575	0.9295	0.0280

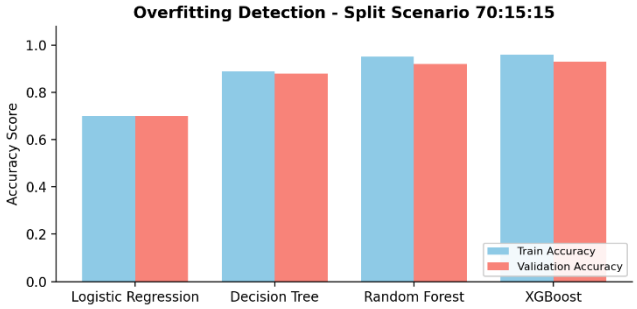


Figure. 6 Training vs. validation accuracy per model (70:15:15)

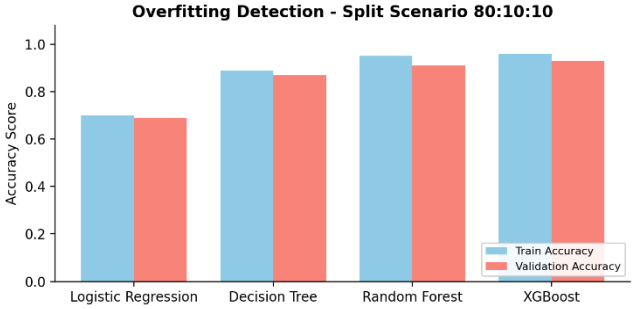


Figure. 7 Training vs. validation accuracy per model (80:10:10)

Based on Table III, the Logistic Regression model shows the lowest accuracy performance in the range of 68–70% in both scenarios, which indicates its limitations

in handling non-linear relationships between pollutant gas parameters (underfitting). In contrast, Decision Tree, Random Forest, and XGBoost show good generalization with a gap below 0.04. In both scenarios, XGBoost consistently recorded the highest validation accuracy compared to the other three algorithms, namely 0.9253 in the 70:15:15 scenario and 0.9295 in the 80:10:10 scenario.

C. Data Splitting Scenario Comparison

To identify the most ideal data-splitting scenario, a thorough comparison was carried out on all eight model-scenario combinations based on the combined score (average of training accuracy and validation accuracy), as summarized in Table IV.

TABLE IV
OVERALL MODEL RANKING BY COMBINED SCORE (SCENARIO COMPARISON)

Rank	Model	Scen.	Train	Val.	Comb.	Gap
1	XGBoost	80:10:10	0.9575	0.9295	0.9435	0.0280
2	XGBoost	70:15:15	0.9589	0.9253	0.9421	0.0336
3	Random Forest	70:15:15	0.9419	0.9181	0.9300	0.0238
4	Random Forest	80:10:10	0.9433	0.9060	0.9246	0.0373
5	Decision Tree	70:15:15	0.8885	0.8759	0.8822	0.0126

The results in Table IV show that the influence of the data-splitting ratio on model performance is model-dependent, so it cannot be concluded that one scenario is absolutely better than the other scenario for all algorithms.

In three of the four algorithms tested, namely Logistic Regression, Decision Tree, and Random Forest, the 70:15:15 scenario consistently produces higher validation accuracy along with a smaller training-validation accuracy gap compared to the 80:10:10 scenario. For example, Random Forest in the 70:15:15 scenario achieves a validation accuracy of 91.81% with a gap of 0.0238, while in the 80:10:10 scenario it only achieves 90.60% with a larger gap of 0.0373. A similar pattern also occurs in Decision Tree (87.59% versus 86.26%) and Logistic Regression (68.55% versus 67.99%). This indicates that the larger proportion of validation data in the 70:15:15 scenario (15% compared to 10%) provides a more stable generalization estimate for these three algorithms, while also reducing the risk of overfitting in models that are relatively more sensitive to the size of the training data.

On the other hand, the XGBoost algorithm shows the opposite pattern. In the 80:10:10 scenario, XGBoost obtains the highest overall validation accuracy of 92.95% with a gap of 0.0280, outperforming its own performance in the 70:15:15 scenario (92.53% with a gap of 0.0336). This shows that the gradual boosting mechanism in XGBoost is able to utilize the additional proportion of training data (80% compared to 70%) more effectively to learn non-linear patterns between pollutant parameters, without sacrificing the model's generalization ability[9][12].

Based on the combined score in Table IV, the combination of XGBoost with the 80:10:10 scenario ranks first overall (combined score of 0.9435), slightly above XGBoost in the 70:15:15 scenario (0.9421). Thus, it can be concluded that the 70:15:15 scenario is generally superior for conventional to bagging-based algorithms (Logistic Regression, Decision Tree, and Random Forest), while the 80:10:10 scenario is the best scenario specifically for the boosting-based algorithm (XGBoost), which is also the model with the highest overall performance in this study. Since the final goal of the study is to obtain the model with the best overall performance, the combination of XGBoost and the 80:10:10 scenario was selected as the final model for further evaluation on the test data[13].

D. Best Model Evaluation on Test Data

The XGBoost model in the 80:10:10 scenario, as the model with the highest combined score, was then tested using test data that had never been used in the training or validation process. This model produced an accuracy of 93.68%. The classification report results are shown in Table V.

TABLE V
OVERALL MODEL RANKING BY COMBINED SCORE (SCENARIO COMPARISON)

Category	Precision	Recall	F1-Score
Good	0.79	0.94	0.86
Very Unhealthy	0.95	0.90	0.92
Moderate	0.96	0.93	0.95
Unhealthy	0.92	0.95	0.94
Weighted Avg.	0.94	0.94	0.94

In general, the model shows very good performance in all categories with a weighted-average F1-score of 0.94. The Good category has the lowest precision (0.79), although it still produces a good F1-score (0.86), which indicates a small number of misclassifications from other categories being predicted as the Good category, likely caused by the relatively small amount of historical data for this category (only 316 out of 5,533 data points) before SMOTE was applied to the training data. The Moderate and Unhealthy categories, which have the largest amount of data, actually show the highest performance with F1-scores of 0.95 and 0.94, respectively[13].

E. Confusion Matrix

The confusion matrix of the XGBoost model on the test data is shown in Fig. 8. The confusion matrix shows that the majority of predictions lie on the main diagonal, which means the model is able to classify most of the data correctly in each category. The most misclassifications occur between adjacent categories, namely Moderate and

Unhealthy, which naturally have closely adjacent pollutant parameter value ranges.

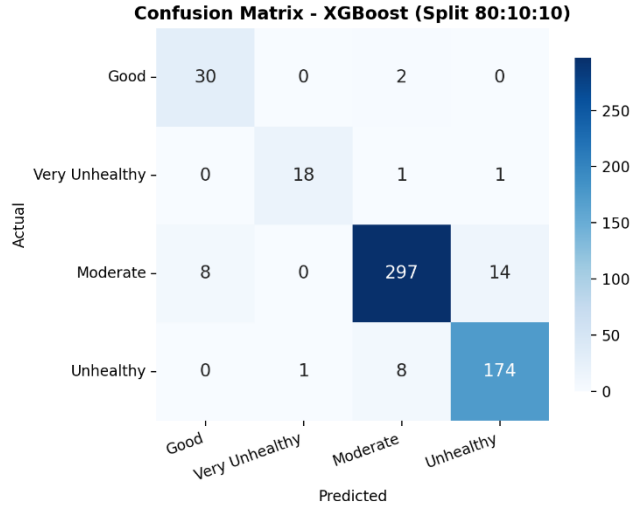


Figure. 8 Confusion matrix of the XGBoost model

F. *Feature Importance*

Based on the feature importance results from the XGBoost model, the pollutant gas parameters that have the greatest influence on the ISPU category classification can be identified, as shown in Fig. 9.

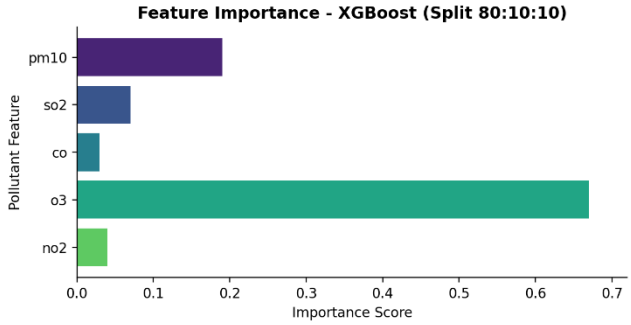


Figure. 9 Feature importance of pollutant gas parameters in the XGBoost model

These results show that the O3 and PM10 parameters provide the highest contribution to the model's classification decisions, consistent with the Spearman correlation analysis results in Fig. 3. Meanwhile, the SO2 and CO parameters provide a relatively lower contribution. This finding shows that not all parameters contribute equally to the classification, so the parameters with the highest importance scores (O3 and PM10) can become the main reference in daily air quality monitoring in DKI Jakarta[14][15].

REFERENCES

[1] F. Hamami and I. A. Dahlan, "Air Quality Classification in Urban Environment using Machine Learning Approach," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 986, no. 1, art. 012004, 2022, doi: 10.1088/1755-1315/986/1/012004.

[2] A. Hadi *et al.*, "Analisis Klasifikasi Kualitas Udara di DKI Jakarta Menggunakan Algoritma Decision Tree," *J. Ilmu Komput. dan Inform.*, vol. 2, no. 3, pp. 90–97, 2026.

[3] R. Firdaus, H. Habibie, and Y. Rizki, "Implementasi Algoritma Random Forest untuk Klasifikasi Pencemaran Udara di Wilayah Jakarta Berdasarkan Jakarta Open Data," *J. FASILKOM*, vol. 14, no. 2, pp. 520–525, 2024, doi: 10.37859/jf.v14i2.7669.

[4] S. S. A. Umri, M. S. Firdaus, and A. Primajaya, "Analysis and Comparison of Classification Algorithm in Air Quality Index," *JIKO (J. Inform. dan Komput.)*, vol. 4, no. 2, pp. 98–104, 2021, doi: 10.33387/jiko.

[5] Kaggle, "Air Quality Index in Jakarta (ispu_dki_all), 2010–2021." [Online]. Available: https://www.kaggle.com/datasets/senadu34/air-quality-index-in-jakarta-2010-2021

[6] G. M. Idroes *et al.*, "Urban Air Quality Classification Using Machine Learning Approach to Enhance Environmental Monitoring," *Leuser J. Environ. Stud.*, vol. 1, no. 2, pp. 62–68, 2023, doi: 10.60084/ljes.v1i2.99.

[7] F. P. Arifianti and A. Salam, "XGBoost and Random Forest Optimization using SMOTE to Classify Air Quality," *Adv. Sustain. Sci. Eng. Technol. (ASSET)*, vol. 6, no. 1, art. 02401025, 2024, doi: 10.26877/asset.v6i1.18136.

[8] A. V. Sophia, F. I. Dinillah, and I. Mahfudzi, "Klasifikasi Indeks Kualitas Udara DKI Jakarta dengan Multinomial Logistic Regression Berbasis Machine Learning," *J. Teknol. Marit.*, vol. 8, no. 2, pp. 43–50, 2025, doi: 10.35991/jtm.v8i2.80.

[9] A. M. Luthfi and F. Fauzi, "Perbandingan Klasifikasi Random Forest, Support Vector Machines, dan LGBM pada Klasifikasi Kualitas Udara di Jakarta," *Justindo (J. Sist. dan Teknol. Inf. Indones.)*, vol. 9, no. 2, pp. 99–108, 2024, doi: 10.32528/justindo.v9i2.1912.

[10] H. Kusniyati, Y. I. Rumahorbo, and R. Yusuf, "Extreme Gradient Boosting Performance on Air Pollution Level Classification in DKI Jakarta," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 11, no. 6, pp. 274–282, 2025, doi: 10.32628/CSEIT2511649.

[11] B. V. Jayadi, M. D. Lauro, Z. Rusdi, and T. Handhayani, "Air Quality Index Classification for Imbalanced Data using Machine Learning Approach," *Sistemasi: J. Sist. Inf.*, vol. 13, no. 3, pp. 951–958, 2024, doi: 10.32520/stmsi.v13i3.3503.

[12] M. Fahmi and I. K. G. Suhartana, "Perbandingan Algoritma Decision Tree dan Support Vector Machine dalam Prediksi Kualitas Udara," *J. Nas. Teknol. Inf. dan Apl. (JNATIA)*, vol. 1, no. 1, pp. 21–30, 2022, doi: 10.24843/JNATIA.2022.v01.i01.p03.

[13] A. A. Nababan, M. Jannah, M. Aulina, and D. Andrian, "Prediksi Kualitas Udara Menggunakan XGBoost dengan Synthetic Minority Oversampling Technique (SMOTE) Berdasarkan Indeks Standar Pencemaran Udara (ISPU)," *JTIK (J. Tek. Inform. Kaputama)*, vol. 7, no. 1, pp. 214–219, 2023, doi: 10.59697/jtik.v7i1.66.

[14] A. R. Kannajmi and D. Saputra, "Penentuan Model Algoritma Klasifikasi Terbaik untuk Klasifikasi Kualitas Udara di Jakarta 2023," *J. Inform. dan Tek. Elektro Terap. (JITET)*, vol. 13, no. 1, 2025, doi: 10.23960/jitet.v13i1.5664.

[15] A. Basuki, B. Nugroho, and Kartini, "Klasifikasi Indeks Kualitas Udara Kota Bandung Menggunakan Algoritma XGBoost: Studi Kasus Data Stasiun Pemantauan Tahun 2024," in *Prosiding Seminar Nasional Informatika Bela Negara (SANTIKA)*, vol. 6, no. 1, pp. 1–6, 2026, doi: 10.33005/santika.v6i1.916.