Available *online* at : http://jnte.ft.unand.ac.id/

Jurnal Nasional Teknik Elektro

| ISSN (Print) 2302-2949 | ISSN (Online) 2407-7267 |

APD-BayTM: Jakarta Air Quality Index Prediction using Bayesian Optimized LSTM

Raey Faldo [1,2], Satria Mandala [1,2], Mohd Shahrizal Sunar [3], Salim M. Zaki [4]

[1] *Human Centric (HUMIC) Engineering, Telkom University, Bandung, Indonesia*
[2] *School of Computing, Telkom University Bandung, 40257, Indonesia*
[3] *MaGICX (Media and Games Innovation Centre of Excellence), Institute of Human Centered Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia*
[4] *Ministry of Higher Education and Scientific Research, Directorate of Studies, Planning and Follow-up, Department of Follow-up, Iraq, Baghdad*

ARTICLE INFORMATION

Received: January 8, 2024
Revised: February 29, 2024
Accepted: March 13, 2024
Available online: July 31, 2024

KEYWORDS

Deep learning, LSTM, air pollution, air quality index (AQI)

CORRESPONDENCE

Email: satriamandala@telkomuniversity.ac.id

A B S T R A C T

The Air Quality Index (AQI) is a metric for evaluating air quality in a region. Jakarta holds the fifth position globally in terms of air pollution. Several studies have been performed to forecast pollution levels in Jakarta. However, existing studies exhibit limitations such as outdated datasets, lack of data normalization, absence of machine learning parameter setting, neglect of k-fold cross-validation, and a failure to incorporate deep learning algorithms for pollution detection. This study introduces an air quality detection system called APD-BayTM to address these issues. This proposed system leverages Long Short-Term Memory (LSTM) and uses Bayesian Optimization (BO) to enhance the performance of air pollution detection. The methodology used in this research involves four key steps: data preprocessing, LSTM model development, hyperparameter tuning through BO, and performance assessment using 5-fold cross-validation. APD-BayTM exhibits robust performance that is comparable to previous research outcomes. The LSTM model in APD-BayTM on the training dataset achieved average precision, recall, F1 score, and accuracy values of 93.29%, 91.41%, 91.89%, and 95.90%, respectively. These metrics improved on the test dataset, reaching 97.44%, 99.71%, 98.52%, and 99.34%, respectively. These findings show the robustness of APD-BayTM across datasets of varying sizes, encompassing both large and small datasets.

INTRODUCTION

The assessment of air quality in a particular area relies heavily on the Air Quality Index (AQI) [1], [2], which is determined by measuring different air pollutants and evaluating their potential health effects [3]. Jakarta, the capital of Indonesia, has an unenviable record of having the highest global air pollution levels. According to IQAir, a Swiss air filtration company that monitors worldwide air quality, Jakarta ranks fifth in AQI scores [4].

As per the 2019 World Health Organization (WHO) report, around 4.2 million premature deaths occur annually worldwide due to air pollution [5]. Additional research [6] and [7] delineate the primary origins and adverse consequences of air pollution, including impacts on respiratory health and other diseases and its role in contributing to global warming. Commonly identified sources of air pollution involve household combustion tools, motor vehicles, industrial facilities, and occurrences like forest fires [8].

Several studies have been conducted to predict Jakarta's Air Quality Index (AQI). In 2022, Permai et al. [9] conducted research using a Support Vector Machine (SVM) and Multilayer Perceptron (MLP). The results showed impressive accuracy levels of 98% and 92%, respectively. The following year, Muljana et al. [10] conducted a study using a Random Forest Classifier (RFC), achieving an outstanding accuracy rate of 95%. The same year, Rafif et al. [11] conducted research using Decision Tree and SVM, obtaining accuracy rates of 87.86% and 90.56%, respectively.

Although the achieved accuracy in the mentioned studies ([9], [10], [11]) is relatively high, these studies still have some shortcomings. Firstly, these studies did not perform normalization on their datasets. Normalization is a crucial aspect of data processing, aiming to create uniformity and ensure variables have a consistent scale [12], [13]. This precautionary measure is vital to avoid potential issues such as unwanted dominance or bias in experimental results [14], [15].

Secondly, studies [10] and [11] did not include hyperparameter tuning, which is a crucial key to optimizing model performance [16], [17]. On the other hand, the study [9] only performed hyperparameter tuning on the MLP model, and the specific method used was not detailed. Additionally, the [9] research did

https://doi.org/10.25077/jnte.v13n2.1183.2024

not incorporate k-fold cross-validation in evaluating the performance of their models. Using k-fold cross-validation enhances result reliability by assessing the models' performance across multiple subsets of the dataset [18].

In addition to the previously mentioned limitations, there are crucial aspects related to the data used in the studies [9], [10], [11]. All three studies utilized datasets from the same source, namely the official Jakarta Open Data website. Notably, these studies did not incorporate the most recent complete dataset available, specifically the 2021 dataset accessible on Jakarta Open Data. This latest dataset introduces a new parameter, PM2.5, which was absent in the datasets used in studies [9], [10], [11].

Lastly, an important point is that those studies did not explore utilising deep learning algorithms for predicting the Air Quality Index (AQI) in Jakarta. While they achieved commendable accuracy using Support Vector Machine (SVM), Multilayer Perceptron (MLP), Random Forest Classifier, and Decision Tree, the absence of deep learning algorithms in their experiments represents a notable gap. Integrating deep learning methodologies could potentially lead to substantial improvements in AQI prediction results, given the ability of deep learning models to capture intricate patterns and dependencies within complex datasets [19]. Considering deep learning approaches could offer a more sophisticated and nuanced understanding of air quality dynamics in Jakarta.

To address the identified shortcomings in previous studies, we propose an air quality detection system in Jakarta based on the Long Short-Term Memory (LSTM) deep learning algorithm in this study. It leverages the LSTM architecture and enhances its performance through Bayesian Optimization (BO). The proposed system, APD-BayTM, is designed to accurately predict Jakarta's Air Quality Index (AQI).

This study focuses on integrating the Long Short-Term Memory (LSTM) model into the APD-BayTM system. By incorporating Bayesian Optimization for parameter tuning, the study aims to significantly enhance the model's predictive capabilities, specifically in forecasting Jakarta's AQI. Utilizing the latest dataset sourced from the Jakarta Open Data website, which includes crucial parameters like PM2.5, ensures the relevance of the data to current air quality conditions in Jakarta. Additionally, this study emphasizes the normalization of the dataset to maintain consistency and ensure that all variables maintain a uniform scale, thus mitigating potential biases and improving the reliability of experimental outcomes.

The APD-BayTM system's anticipated success is poised to positively impact the comprehension and management of air quality challenges in metropolitan cities. This study will contribute significantly to the existing air quality prediction literature by prioritizing deep learning methods and parameter optimization. Furthermore, the success of this system is likely to inspire the development of more sophisticated and accurate prediction methods in the future, further advancing the field of air quality assessment and management.

RELATED WORK

Numerous studies have been conducted on Air Quality Index (AQI) predictions. In a study by Almaliki et al. [20], various machine learning models, including Fine Decision Tree (FDT), Ensemble Boosted Tree (EBOT), and Ensemble Bagged Tree (EBAT), were proposed for AQI detection, utilizing data from 2016 to 2018. The experimental results demonstrated that the EBOT model outperformed other models, achieving an accuracy of 97.4%.

A study by Saminathan et al. [21] has considered several machine learning algorithms, such as Logistic Regression, SVM, Random Forest, Extreme Gradient Boosting, and Multilayer Perceptron, to classify PM2.5. Furthermore, a SMOTE-based approach was employed to address data imbalance, and the results demonstrated that using Random Forest with balanced data yielded superior accuracy.

In India, Babu et al. [22] used Logistic Regression, Random Forest, K-Nearest Neighbours, Decision Tree, and SVM for air quality prediction. The experimental results showed that, although its accuracy is not explicitly mentioned, the Decision Tree is the most proficient machine learning algorithm for detecting the Air Quality Index (AQI).

A study by Pant et al. [23] focused on predicting AQI based on PM10, PM 2.5, SO2, and NO2 pollutants, with the Decision Tree Classifier demonstrating the highest accuracy (98.63%) among the models developed. Logistic Regression had the lowest accuracy at 91.78%.

In a study [24], Attaallah et al. developed five machine learning models, including a novel SMOTEDNN model for air pollution classification. The SMOTEDNN model achieved the highest accuracy (99.90%) compared to other models.

Additionally, a study by Ragab et al. [25] proposed a forecasting model using a One-Dimensional Deep Convolutional Neural Network (1D-CNN) and Exponential Adaptive Gradients optimization for predicting the Air Pollution Index (API) in Klang. The model exhibited better accuracy than the benchmark model, showcasing various performance metrics.

A study by Toharudin et al. [26] tackled imbalanced data related to PM2.5 concentrations, employing boosting algorithms like AdaBoost, XGBoost, CatBoost, and LightGBM. The boosting methods effectively reduced bias and improved variance reduction, resulting in improved classification performance.

In a study by Permai et al. [9], Support Vector Machine (SVM) and Multilayer Perceptron (MLP) models were employed to classify air quality in Jakarta based on the Air Pollution Standard Index (ISPU) dataset from 2018 to 2021. The SVM model with a Polynomial kernel demonstrated excellent detection performance across various metrics, including accuracy, recall, precision, and F1 score. In their experiments, the accuracy metric reached an impressive 98%. In contrast, the MLP model achieved a slightly lower metric score.

A study by Muljana et al. [10] utilized a Random Forest Classifier (RFC) to predict AQI in Jakarta, addressing class imbalance with the SMOTE-tomek technique. Results indicated favourable performance metrics for RFC, with average precision, recall, F1 score, and accuracy reaching 87.00%, 95.00%, 90.00%, and 95.00%, respectively.

Subsequently, a study by Rafif et al. [11] focused on Decision Tree and SVM models for AQI prediction in Jakarta, using the same dataset as [10] and addressing class imbalance with the SMOTE-ENN technique. The Decision Tree and SVM models demonstrated good performance, with various metrics indicating their effectiveness.

Lastly, a study by Vlachou et al.[27] developed a classification model using Bayesian Logistic Regression with Markov Chain Monte Carlo (MCMC) sampling technique, achieving a maximum accuracy of 87.91%. However, it slightly lagged behind traditional Frequency Logistic Regression regarding accuracy and Area Under the Curve (AUC) score.

Table 1 summarises the methodologies, use cases, and results from recent air quality index prediction or classification studies.

Table 1. Related Work Summary

Reference	Methodology	Use Case	Result
Almaliki et al. [20]	FDT, EBOT, EBAT	AQI prediction	EBOT: 97.4% Accuracy
Saminathan et al. [21]	Logistic Regression, SVM, Random Forest, XGBoost, MLP	PM2.5 Classification	Random Forest (Balanced): Superior Accuracy
Babu et al. [22]	Logistic Regression, Random Forest, KNN, Decision Tree, SVM	Air Quality Prediction (India)	Decision Tree: Proficient AQI Detection
Pant et al. [23]	Decision Tree Classifier	AQI Prediction based on PM10, PM2.5, SO2, NO2	Decision Tree: 98.63% Accuracy
Attaallah et al. [24]	SMOTEDNN, various ML models	Air Pollution Classification	SMOTEDNN: 99.90% Accuracy
Ragab et al. [25]	1D-CNN, Exponential Adaptive Gradients	API Prediction (Klang)	Outperformed Benchmark Model
Toharudin et al. [26]	Boosting Algorithms (AdaBoost, XGBoost, CatBoost, LightGBM)	Imbalanced PM2.5 Data	Improved Classification Performance
Permai et al. [9]	SVM, MLP	AQI Prediction (Jakarta)	SVM (Polynomial Kernel): 98% Accuracy
Muljana et al. [10]	RFC (SMOTE-tomek for Class Imbalance)	AQI Prediction (Jakarta)	RFC: 87.00% Precision, 95.00% Recall, 90.00% F1, 95.00% Accuracy
Rafif et al. [11]	Decision Tree, SVM (SMOTE-ENN for Class Imbalance)	AQI Prediction (Jakarta)	Good Performance Across Metrics
Vlachou et al. [27]	Bayesian Logistic Regression (MCMC Sampling)	Air Quality Classification	Maximum Accuracy: 87.91%

PROPOSED AIR POLLUTION DETECTION USING BAYESIAN LSTM (APD-BAYTM)

The APD-BayTM system is designed to predict air quality levels categorized as GOOD, MODERATE, and UNHEALTHY. As illustrated in Figure 1, the system consists of two primary stages: preprocessing and classification. Tasks such as addressing missing values, normalization, and handling categorical data are performed in the preprocessing step. Following this, the classification process employs LSTM models, with model parameters optimized using Bayesian Optimization. The implementation of the LSTM model utilizes version 2 of the TensorFlow library.

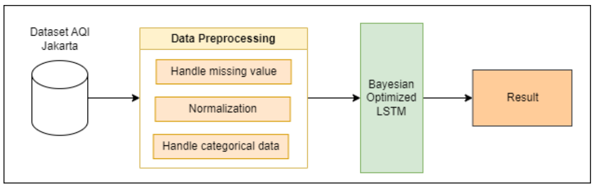

Figure 1. The Proposed APD-BayTM

Long Short-Term Memory (LSTM)

The Long Short-Term Memory (LSTM) is a model or architecture specifically developed to overcome the shortcomings of conventional Recurrent Neural Networks (RNNs), which exhibit limited short-term memory and necessitate extended training periods [28].

In contrast to traditional RNNs, LSTMs employ an auxiliary memory unit to assess the importance of information through a more sophisticated stochastic model. Its typical architecture features a "long-time memory function," enabling it to address long-term nonlinear sequential prediction challenges by incorporating a gating cell. Moreover, LSTMs tackle the issues of

https://doi.org/10.25077/jnte.v13n2.1183.2024

gradient vanishing and explosion during prolonged training sequences, a drawback often encountered in standard RNNs [29].

Unlike regular RNNs, LSTMs replace neurons in the hidden layer with memory cells equipped with gating mechanisms. The output gate adjusts the value transferred to the output from the current state, and the input gate enhances the current input before adding it to the next state. The forget gate selectively determines which elements of the current state should be carried forward.

The structure of the LSTM model developed in this study is depicted in Figure 2. This model comprises three layers arranged in sequence. The initial layer is an LSTM layer with 57 units. The choice of 57 units in this layer results from hyperparameter tuning using Bayesian Optimization (BO). It was observed that this configuration struck a balance between model complexity and computational efficiency. The aim is to ensure the network's capacity to recognize essential temporal features while avoiding unnecessary complexity that may lead to overfitting or computational overhead.

The second layer is another LSTM layer with 67 units. Building upon the insights gained from the first layer, the second LSTM layer is introduced to enhance the model's capability to capture more intricate temporal relationships. This augmentation helps in accommodating potential complexities that may not be adequately captured by the first layer alone. The specific choice of 67 units results from hyperparameter tuning using BO, aiming for a nuanced representation of temporal dependencies without overwhelming the model with unnecessary parameters.

Finally, the third layer is a dense layer featuring three units, aligning with the number of classes intended for prediction (Good, Moderate, Unhealthy). This layer utilizes the softmax activation function to produce a probability distribution of classes, serving as the model's classification output. The decision to use softmax activation, as opposed to other activation methods, stems from its suitability for multiclass classification problems. Softmax ensures that the predicted class probabilities sum to one, providing a transparent and interpretable output for classification tasks.

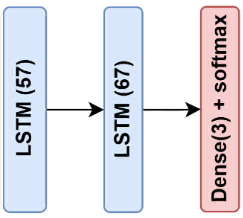

Figure 2. LSTM Model Architecture

BAYESIAN OPTIMIZATION

Iteratively adjusting hyperparameters through trial and error can be laborious and often unproductive [30]. Therefore, efficient tuning methods become crucial, mainly when the objective of optimization is to identify the optimal value at a specific sample point for an unknown function, as outlined in equation (1) [31].

$$p^{+} = arg\ max_{p \in \emptyset} \vartheta(p) \tag{1}$$

Here, p represents the sampling point, ϕ signifies the search region around p, ϑ denotes the unknown objective function, and $p+$ denotes the location where the unidentified objective function is to be maximized.

Bayesian optimization (BO) emerges as a more efficient technique for hyperparameter optimization compared to traditional grid search (GS) and random search (RS) methods. In both GS and RS iterations, evaluations are conducted independently of prior assessments, resulting in inefficient allocation of time evaluating poorly performing regions within the hyperparameter search space. BO addresses this challenge by incorporating prior information about the unknown objective function ϑ and sample locations to estimate the distribution of the objective function using the Bayesian theorem. This posterior information is then used to seek the globally optimal value [32].

The Bayesian optimization process comprises two key phases [33]. In the first phase, the BO algorithm randomly selects data points to construct a surrogate function across the unknown objective function ϑ. A Gaussian process (GP) is employed to update the surrogate function and generate the posterior distribution over ϑ due to its versatility, robustness, accuracy, and analytical convenience.

An acquisition function is formulated in the second phase, utilizing the posterior distribution generated in phase one. This function aims to explore new areas within the search space while exploiting existing regions with the best results. The exploration and exploitation processes continue until a predefined stopping threshold is reached, during which the surrogate model is continuously updated with fresh results. The objective is to maximize the acquisition function to identify the next sampling point.

METHODS

This section outlines the materials and methods employed in the present study. The materials include the dataset and matrices utilized to assess the performance of the APD-BayTM model. Meanwhile, the method includes a step-by-step procedure for developing the APD-BayTM system.

Materials

Dataset

The dataset utilized in this study comprises Standard Air Pollution Index (ISPU) data obtained from five air quality monitoring stations (SPKU) in Jakarta Province, Indonesia, for the year 2021 [34]. Data collection occurred from January 2021 to December 2021, resulting in 1809 data points. The dataset is accessible on the Open Data Jakarta website [34]. For a comprehensive understanding of the dataset attributes, refer to Table 2.

The AQI categories within the dataset adhere to the Regulation of the Minister of Environment and Forestry of the Republic of Indonesia Number P.14/MENLHK/SETJEN/KUM.1/7/2020, outlining the Standard Air Pollution Index [35]. This regulation states that ISPU is categorized into five levels: Good, Moderate, Unhealthy, Very Unhealthy, and Dangerous. ISPU incorporates parameters such as particulate matter 10 (PM10), particulate matter 2.5 (PM2.5), carbon monoxide (CO), nitrogen dioxide (NO2), sulfur dioxide (SO2), ozone (O3), and hydrocarbons (HC). Equation (2) is utilized to calculate the ISPU value.

Table 2. Dataset Attribute

Column	Type	Description
Date	Date	Date of air quality measurement
Station	String	Location of data collection
PM_{10}	Int	Particulate matter with a diameter of less than 10 micrometres
$PM_{2.5}$	Int	Particulate matter with a diameter of less than 2.5 micrometres
SO_2	Int	Sulphide (in the form of SO2)
CO	Int	Carbon Monoxide
O_3	Int	Ozone
NO_2	Int	Nitrogen Dioxide
Max	Int	The highest measured value among all parameters at the same time
Critical	String	Parameter with the highest measurement result: PM_{10}, $PM_{2.5}$, SO_2, CO, O_3, NO_2
Category	String	Category of AQI calculation results: GOOD, MODERATE, UNHEALTHY

$$I = \frac{I_{High} - I_{Low}}{X_{High} - X_{Low}} (X_x - X_{Low}) + I_{Low} \quad (2)$$

Here, I represent the AQI sub-index value for air quality pollutants (PM10, PM2.5, CO, NO2, SO2, O3, HC). I_{High} and I_{Low} are AQI's upper and lower limits for a specific pollutant. X_{High} and X_{Low} are the upper and lower limits of the pollutant concentration. X_x is the concentration of the pollutant. After computing all AQI sub-indices, the sub-index with the highest value is selected as the AQI. The air quality category, determined based on the calculated AQI, along with explanations for each category, is detailed in Table 3.

Table 3. AQI Category

Range	Category	Explanation
1-50	Good	Excellent air quality poses no adverse effects on humans, animals, and plants.
51-100	Moderate	Air quality is still acceptable for human, animal, and plant health.
101-200	Unhealthy	Air quality that is harmful to humans, animals, and plants.
201-300	Very Unhealthy	Air quality can increase health risks for specific exposed segments of the population.
301+	Hazardous	Air quality can cause severe health damage to the population and requires prompt intervention.

Despite government regulations stipulating five categories (Good, Moderate, Unhealthy, Very Unhealthy, and Hazardous) and seven pollutants (PM10, PM2.5, CO, NO2, SO2, O3, HC), this study focuses on three categories (Good, Moderate, Unhealthy) and six pollutants (PM10, PM2.5, CO, NO2, SO2, O3). The decision is based on the limitations of the available dataset, which does not include the specified categories and pollutants.

Metrics

The performance evaluation of the developed model will rely on accuracy, precision, recall, and F1-score. Accuracy represents the ratio of correct predictions (both positive and negative) to the total number of predictions. Precision is the ratio of true positive predictions (TP) to the total positive predictions performed (true positive and false positive). Recall is the ratio of true positive predictions (TP) to the total actual positive class instances (TP and false negative (FN)). F1-Score serves as a combined measure of precision and recall. The accuracy, precision, recall, and F1-score equations are provided in Equation (3) through (6).

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ Number\ of\ Sentiment}\ x\ 100\% \quad (3)$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}\ x\ 100\% \quad (4)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}\ x\ 100\% \quad (5)$$

$$F1\ score = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall}\ x\ 100 \quad (6)$$

Methods To Develop AQI Prediction LSTM Model

This study aims to develop an air quality detection system for Jakarta using the LSTM deep learning algorithm optimized through Bayesian Optimization (APD-BayTM). The detection classes are categorized into GOOD, MODERATE, and UNHEALTHY. The APD-BayTM model is developed in two scenarios.

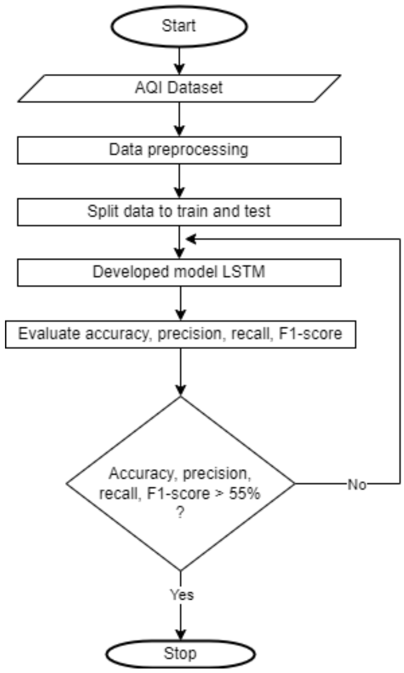

Figure 3. Workflow Method to Develop Model with The Default Parameters

In the first scenario, the model is built with default parameters, while in the second scenario, hyperparameter tuning is performed

 https://doi.org/10.25077/jnte.v13n2.1183.2024

using Bayesian optimization. The steps involved in developing the model for the first scenario are illustrated in Figure 3.

Figure 4 depicts the steps in developing the second scenario's model. The procedures in both scenarios closely mirror each other. Initially, preprocessing is done on the dataset, dividing the data into training and testing sets. Subsequently, the LSTM model is constructed. In the subsequent step, in the first scenario, the performance of the developed models is directly assessed. Conversely, hyperparameter tuning is implemented in the second scenario before evaluating the model's performance.

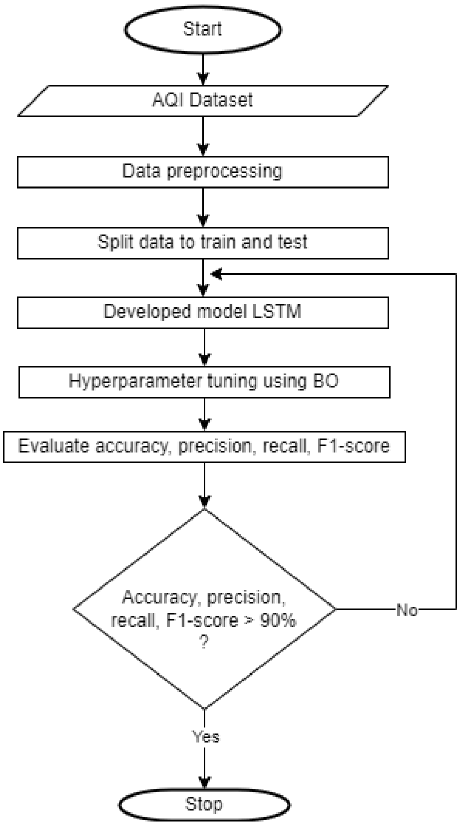

Figure 4. Workflow Method to Develop Model with Hyperparameter Tuning

Data Preprocessing

Figure 5 presents the visual representation of the data preprocessing steps that were conducted. Initially, the attributes date, station, max, and critical were eliminated as they are not utilized in the training process. Subsequently, missing values, represented by "---" in this dataset, were addressed by replacing them with "NaN" (Not-a-Number) using the pandas library. Following this, rows containing NaN values were removed to ensure the use of complete data.

After handling missing data, the next step involved feature normalization, a process of scaling all features in the dataset to have a consistent range. This step is crucial to prevent any single feature from exerting a disproportionate influence on our model. For feature normalization, the Min-Max Scaler from the scikit-learn library was employed.

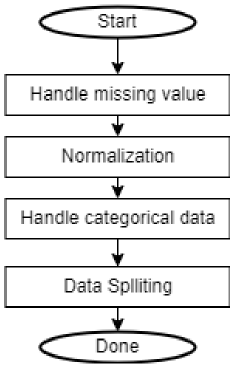

Figure 5. Flowchart Data Preprocessing

The LabelEncoder function from the Keras library is utilized to transform string values in the category output column into numeric values to predict air pollution data. This conversion assigns the label "Good" to 0, "Moderate" to 1, and "Unhealthy" to 2. Subsequently, the data is divided, with 90% allocated for training and validation and the remaining 10% reserved for evaluating the performance of the trained and validated models. The data splitting process is carried out using the train_test_split function.

Experiment Scenario

Two scenarios are employed in developing the APD-BayTM model in this research.

Default Parameters

In this scenario, the models are developed using the default parameters. The LSTM models are constructed using TensorFlow v2 [36]. Table 4 displays the default parameters employed in developing the LSTM model in the first scenario.

Table 4. Default Parameters for Developing LSTM Model

Parameter	Range Value
units	32 (on the first LSTM layer), 3 (on the second LSTM layer), 3 (on the Dense layer)
optimizer	Adam
learning_rate	0.001
activation	softmax
batch_size	128
epoch	50

Hyperparameter Tuning

In the second scenario, the LSTM model is developed through hyperparameter tuning using the Bayesian Optimization technique, with detailed parameter ranges in Table 5. Subsequently, the results of hyperparameter tuning are presented in Table 6.

Table 5. Range of Hyperparameters Used for Bayesian Optimization

Parameter	Range Value
units	16 – 128 (on the first LSTM layer)
	16 – 128 (on the second LSTM layer)
learning_rate	0.001 - 0.1
batch_size	32 – 128
epoch	50 – 100

Table 6. Hyperparameter Tuning for Developing LSTM Model

Parameter	Range Value
units	57 (on the first LSTM layer), 67 (on the second LSTM layer)
optimizer	Adam
learning_rate	0.0299223202049866
activation	softmax
batch_size	91

Comparison of The Performance of APD-BayTM with State Of The Arts Research In Detecting AQI

After the successful development of APD-BayTM, the next step is to evaluate its performance and compare it with models from previous research. These models include the Support Vector Machine (SVM) developed by Permai et al. [9], with an accuracy of 98%, the Random Forest Class developed by Muljana et al. [10], with an accuracy of 95%, and the SVM model developed by Rafif et al. [11], with an accuracy of 90.56%. This comparison will provide valuable insights into the strengths and limitations of the APD-BayTM model in the field of AQI detection in Jakarta, contributing significantly to the advancement and understanding of methodologies in the literature.

RESULTS AND DISCUSSION

Results

APD-BayTM with Default Parameters

In Table 7, the results of the 5-fold cross-validation demonstrated the average performance of the APD-BayTM model with the default parameters. The average performance metrics for the APD-BayTM model in 5-fold cross-validation show moderate precision, recall, F1 Score, and accuracy.

Table 7. Average Performance Metrics for APD- BayTM in 5-Fold Cross-Validation

Precision	Recall	F1 Score	Accuracy
59.66%	56.26%	55.03%	80.29%

Table 8 provided detailed performance metrics from the 5-fold cross-validation for the APD-BayTM model. The model's performance varied across each fold, with precision ranging from 50.66% to 80.79%, Recall from 38.89% to 71.49%, F1 Score from 38.53% to 75.18%, and Accuracy from 74.73% to 85.35%. Fold 2 stands out with higher precision, recall, F1 Score, and accuracy, suggesting the model's performance depends on the specific data subset.

Table 8. Performance Metrics for APD- BayTM in 5-Fold Cross-Validation

Fold	Precision	Recall	F1 Score	Accuracy
1	59.01%	38.89%	38.53%	77.29%
2	**80.79%**	**71.49%**	**75.18%**	**85.35%**
3	54.56%	54.16%	53.95%	82.78%
4	53.28%	51.17%	50.48%	74.73%
5	50.66%	65.57%	57.01%	81.32%

Table 9 illustrates the performance in each class for every fold of the APD-BayTM model. The results showed varying precision, recall, and F1 scores across the different folds and classes. Notably, the model struggled in specific folds, particularly in distinguishing the "Unhealthy" class, where precision, recall, and F1 scores were consistently at 0.00%. The "Good" class exhibited fluctuating performance, with notable improvements in precision in some folds but lower recall.

Table 9. Performance Metrics for APD-BayTM Across Every Class in 5-Fold Cross-Validation

Fold	Class	Precision	Recall	F1 Score	Accuracy
1	Good	100.00%	16.67%	28.57%	77.29%
	Moderate	77.04%	100.00%	87.03%	
	Unhealthy	0.00%	0.00%	0.00%	
2	Good	73.68%	63.64%	68.29%	85.35%
	Moderate	86.88%	94.58%	90.57%	
	Unhealthy	81.82%	56.25%	66.67%	
3	Good	0.00%	0.00%	0.00%	82.78%
	Moderate	83.12%	96.57%	89.34%	
	Unhealthy	80.56%	65.91%	72.50%	
4	Good	85.71%	54.55%	66.67%	74.73%
	Moderate	74.13%	98.97%	84.77%	
	Unhealthy	0.00%	0.00%	0.00%	
5	Good	69.57%	100.00%	82.05%	81.32%
	Moderate	82.40%	96.71%	88.98%	
	Unhealthy	0.00%	0.00%	0.00%	

The confusion matrix for the fold with the highest accuracy for the APD-BayTM model is presented in Figure 6. The model correctly identified 14 Good class instances, with eight misclassified as Moderate. The model achieved 192 true positives for the Moderate class while misclassifying five instances as Good and six as Unhealthy. The model correctly identified 27 instances in the Unhealthy class, with 21 instances misclassified as Moderate. It indicates that the model performed well in predicting Moderate air quality but faced challenges in accurately classifying instances in the Good and Unhealthy categories during this specific fold

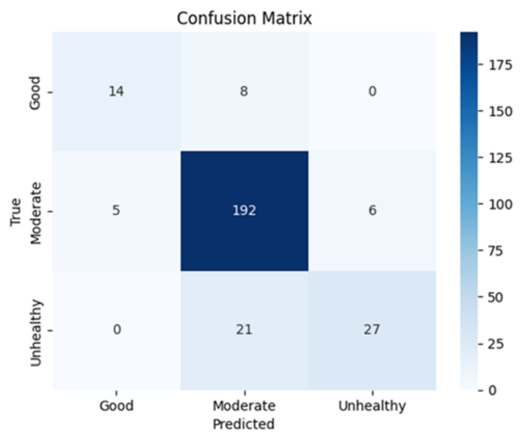

Figure 6. Confusion Matrix of APD-BayTM on Fold 2

The loss and accuracy graphs during training and validation on the fold with the highest accuracy for the APD-BayTM model can be found in Figure 7 and Figure 8, respectively.

https://doi.org/10.25077/jnte.v13n2.1183.2024

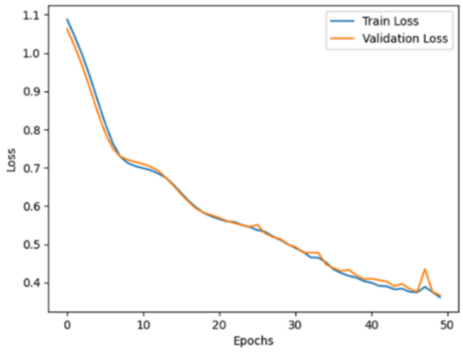

Figure 7. The Loss Graph of Training and Validation for APD-BayTM on Fold 2

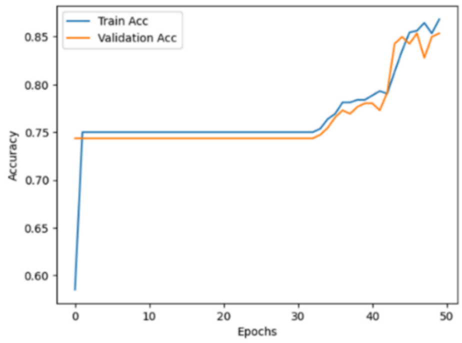

Figure 8. The Accuracy of Training and Validation for APD-BayTM on Fold 2

Table 10 summarises the overall performance of the APD-BayTM model, specifically generated from fold 2 during cross-validation when evaluated on the test set. In Table 11, a more detailed analysis was provided for each class. The "Moderate" class exhibits high precision, recall, and F1 Score, whereas the "Unhealthy" class faces challenges.

Table 10. Average Performance Metrics for The APD-BayTM Model from Fold 2 on Test Set Data

Precision	Recall	F1 Score	Accuracy
84.10%	77.00%	79.02%	87.50%

Table 11. Performance Metrics for The APD-BayTM Model from Fold 2 on Test Set in Detail

Class	Precision	Recall	F1 Score	Accuracy
Good	76.92%	83.33%	80.00%	
Moderate	88.71%	95.65%	92.05%	87.50%
Unhealthy	86.67%	52.00%	65.00%	

Figure 9 illustrates the confusion matrix for the APD-BayTM model, specifically generated from fold 2 during cross-validation when evaluated on the test set data. The model correctly identified ten Good class instances, with two misclassified as Moderate. The model achieved 110 true positives for the Moderate class while misclassifying three instances as Good and two as Unhealthy. In the Unhealthy class, the model correctly identified 13 instances, but there was a misclassification of 12 instances as Moderate. Overall, the model displayed commendable accuracy in predicting the Good and Moderate classes on the test set, with notable areas for improvement in minimizing false positives in the Unhealthy class.

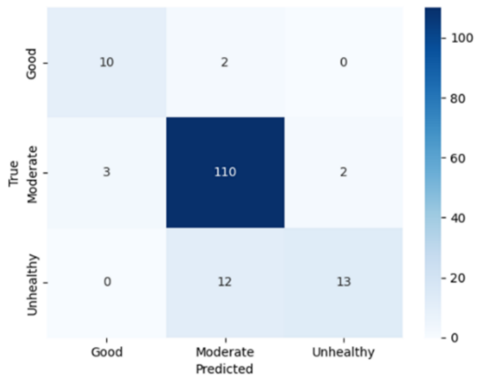

Figure 9. Confusion Matrix of APD-BayTM Model from Fold 2 on Test Set

APD-BayTM with Hyperparameter Tuning

In Table 12, the results of 5-fold cross-validation demonstrate the average performance of the APD-BayTM model with hyperparameter tuning. After hyperparameter tuning, there is a significant improvement in average precision, recall, F1 Score, and accuracy compared to the default model. This improvement suggests the effectiveness of hyperparameter tuning in enhancing the model's overall performance. This improvement is attributed to the model's adaptability achieved through hyperparameter tuning, allowing it to accommodate the unique characteristics of different datasets [37].

Table 12. Average Performance Metrics for APD-BayTM with Hyperparameter Tuning in 5-Fold Cross-Validation

Precision	Recall	F1 Score	Accuracy
93.29%	91.41%	91.89%	95.90%

Table 13 provided detailed performance metrics from the 5-fold cross-validation for the APD-BayTM model with hyperparameter tuning, showing the metrics for each fold. The model consistently exhibited high performance in each fold, with precision ranging from 89.11% to 96.58%, recall from 86.82% to 97.50%, F1 score from 87.36% to 96.23%, and accuracy from 93.04% to 98.17%. These consistent high-performance metrics across folds indicate the robustness and reliability of the APD-BayTM model with hyperparameter tuning in predicting AQI.

Table 13. Performance Metrics for APD-BayTM with Hyperparameter Tuning in 5-Fold Cross-Validation

Fold	Precision	Recall	F1 Score	Accuracy
1	94.00%	94.99%	94.48%	97.07%
2	89.11%	86.87%	87.36%	93.04%
3	**96.58%**	86.82%	90.17%	95.97%
4	95.04%	**97.50%**	**96.23%**	**98.17%**
5	91.72%	90.88%	91.24%	95.24%

Table 14 details the performance in each class for every fold of the APD-BayTM model with hyperparameter tuning. Precision, recall, and F1 score were consistently higher for all classes in each fold, showcasing the positive impact of hyperparameter tuning. The "Unhealthy" class, previously challenging for the LSTM model, exhibited remarkable improvements in precision, recall, and F1 score. Overall, the hyperparameter-tuned LSTM model presented more robust and balanced performance across all folds and classes, underscoring the importance of fine-tuning in optimizing model outcomes.

Table 14. Performance Metrics for Apd-BayTM with Hyperparameter Tuning Across Every Class in 5-Fold Cross-Validation

Fold	Class	Precision	Recall	F1 Score	Accuracy
1	Good	89.47%	89.47%	89.47%	97.07%
	Moderate	98.53%	97.57%	98.05%	
	Unhealthy	94.00%	97.92%	95.92%	
2	Good	88.24%	68.18%	76.92%	93.04%
	Moderate	96.06%	94.66%	95.35%	
	Unhealthy	83.02%	97.78%	89.80%	
3	Good	100.00%	61.90%	76.47%	95.97%
	Moderate	96.26%	98.56%	97.40%	
	Unhealthy	93.48%	100.00%	96.63%	
4	Good	89.47%	94.44%	91.89%	98.17%
	Moderate	99.50%	98.05%	98.77%	
	Unhealthy	96.15%	100.00%	98.04%	
5	Good	79.17%	82.61%	80.85%	95.24%
	Moderate	95.98%	97.45%	96.71%	
	Unhealthy	100.00%	92.59%	96.15%	

The confusion matrix for the fold with the highest accuracy for the APD-BayTM model can be observed in Figure 10. The model correctly identified 17 instances in the Good class, with only one instance misclassified as Moderate. The model achieved 201 true positives for the Moderate class and misclassified two instances, one each as Good and Unhealthy. Regarding the Unhealthy class, the model perfectly identified all 50 instances, with none misclassified, indicating robust performance in distinguishing the Unhealthy class. Overall, the model showcased strong predictive capabilities across all classes in Fold 4, with minimal misclassifications.

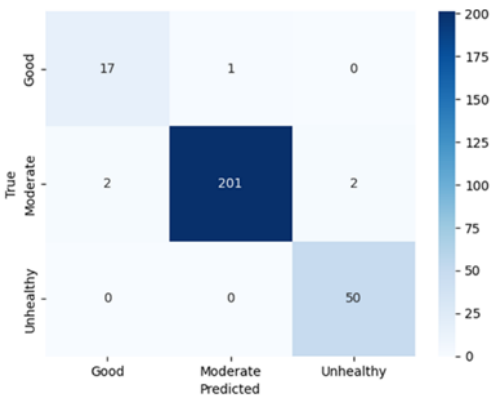

Figure 10. Confusion Matrix of APD-BayTM with Hyperparameter Tuning on Fold 4

The loss and accuracy graphs during training and validation for the fold with the highest accuracy for the APD-BayTM model with hyperparameter tuning can be found in Figure 11 and Figure 12, respectively.

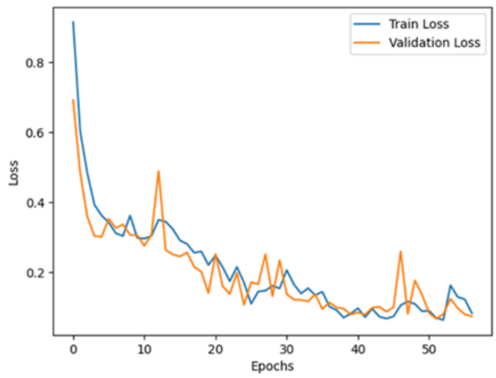

Figure 11. The Loss Graph of Training and Validation for APD-BayTM with Hyperparameter Tuning on Fold 4

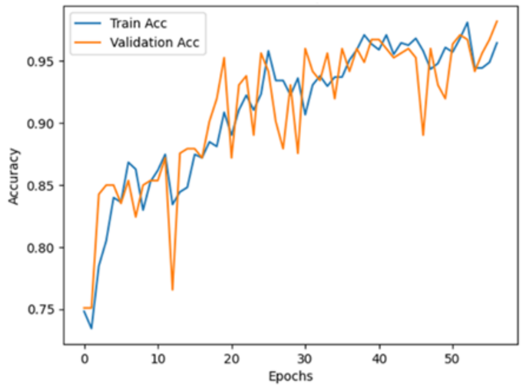

Figure 12. The Accuracy Graph of Training and Validation for APD-BayTM with Hyperparameter Tuning on Fold 4

Table 15 summarises the overall performance of the APD-BayTM model with hyperparameter tuning, specifically generated from fold 4 during cross-validation when evaluated on the test set.

Table 15. Average Performance Metrics For Apd-BayTM With Hyperparameter Tuning From Fold 4 On Test Set Data

Precision	Recall	F1 Score	Accuracy
97.44%	99.71%	98.52%	99.34%

Table 16 presented a detailed analysis of the performance metrics for the tuned APD-BayTM model generated from fold 4 and evaluated on the test set data across different classes. The APD-BayTM model with hyperparameter tuning from fold 4 exhibits robust and impressive performance across all classes, showcasing its ability to accurately predict instances in the "Good," "Moderate," and "Unhealthy" classes.

Table 16. Performance Metrics for Apd-BayTM With Hyperparameter Tuning From Fold 4 on Test Set Data in Detail

Class	Precision	Recall	F1 Score	Accuracy
Good	92.31%	100.00%	96.00%	99.34%
Moderate	100.00%	99.13%	99.56%	
Unhealthy	100.00%	100.00%	100.00%	

Figure 13 illustrates the confusion matrix for the APD-BayTM model with hyperparameter tuning, specifically generated from fold 4 during cross-validation when evaluated on the test set. The model accurately identified all 12 instances in the Good class, resulting in no misclassifications. The model achieved 114 true

https://doi.org/10.25077/jnte.v13n2.1183.2024

positives for the Moderate class and misclassified one instance as Good. The model correctly identified all 25 instances in the Unhealthy class with no misclassifications. The model showcased strong predictive capabilities across all classes on the test set after hyperparameter tuning, with minimal misclassifications.

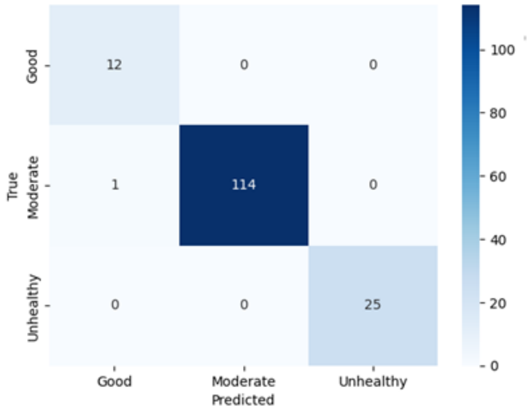

Figure 13. Confusion Matrix of APD-BayTM Model with Hyperparameter Tuning from Fold 4 on Test Set

Discussion

Comparison of APD-BayTM Performance In Both Training Set and Test Set

In developing the APD-BayTM system, we partitioned the dataset into two sets (a training set and a test set) with a distribution ratio of 90:10. The training set was used in the 5-fold cross-validation process. Following this, we assessed the model's performance across folds and identified the fold with the highest performance. This top-performing model was then further evaluated using the independent test set.

Table 17 compares the average performance metrics for the APD-BayTM model from Fold 2 in both the training and test sets, utilizing default parameters. On the training set, the model demonstrated a precision of 80.79%, recall of 71.49%, F1 score of 75.18%, and an accuracy of 85.35%. Upon evaluation of the test set, the model's performance improved, with a precision of 84.10%, recall of 77.00%, F1 score of 79.02%, and an accuracy of 87.50%. This comparison indicates the model's ability to generalize well to unseen data, as reflected in the improved metrics on the test set.

Table 18 extends the analysis by comparing average performance metrics for the APD-BayTM model from Fold 4 in both the training and test sets, incorporating hyperparameter tuning. On the training set, the hyperparameter-tuned model exhibited enhanced performance with a precision of 95.04%, recall of 97.50%, F1 score of 96.23%, and an accuracy of 98.17%. The superior performance of the tuned model was further affirmed on the test set, with a precision of 97.44%, a recall of 99.71%, an F1 score of 98.52%, and an accuracy of 99.34%.

This analysis underscores the efficacy of hyperparameter tuning in optimizing the APD-BayTM model's performance. The substantial improvements in precision, recall, F1 score, and accuracy on both the training and test sets highlight the effectiveness of tuning in enhancing the model's predictive capabilities. The improved performance consistency across different evaluation sets suggests that the hyperparameter-tuned LSTM model is robust and capable of delivering reliable predictions in diverse scenarios.

Table 17. Comparison of Average Performance Metrics for APD-BayTM from Fold 2 in Both Training and Test Sets with Default Parameters

Data	Precision	Recall	F1 Score	Accuracy
Train set	80.79%	71.49%	75.18%	85.35%
Test set	84.10%	77.00%	79.02%	87.50%

Table 18. Comparison of Average Performance Metrics for APD-BayTM from Fold 4 in Both Training and Test Sets with Hyperparameter Tuning

Data	Precision	Recall	F1 Score	Accuracy
Train set	95.04%	97.50%	96.23%	98.17%
Test set	97.44%	99.71%	98.52%	99.34%

Comparison of The Performance of Apd-BayTM With State Of The Arts Previous Research

In Table 19, a comparative analysis is presented between the APD-BayTM model and models from three previous studies that employed different methodologies: Random Forest Classifier (RFC) by Muljana et al. [10], [11], Support Vector Machine (SVM) by Rafif et al. [11], and SVM by Permai et al. [9]. The evaluation metrics include precision, recall, F1 score, and accuracy.

Table 19. Comparative analysis with previous studies

Research Work	Method	Precision	Recall	F1 Score	Accuracy
Muljana et al. [10]	RFC	87.00%	95.00 %	90.00 %	95.00%
Rafif et al. [11]	SVM	90.00%	92.00 %	90.00 %	90.56%
Permai et al. [9]	SVM	99.00%	98.00 %	98.58 %	98.00%
Proposed Model	LSTM	93.29%	91.41 %	91.89 %	95.90%

The proposed model, APD-BayTM, is superior in precision when compared to RFC Muljana and SVM Rafif, but lags behind SVM Permai. Although it performs better in accuracy than RFC Muljana and SVM Rafif, it still lags behind SVM Permai. The recall of this proposed model falls below that of all three previous studies, thus demonstrating potential improvements in capturing all relevant examples. Although the F1 score exceeds RFC Muljana and SVM Rafif, it is still below SVM Permai. These results show that there is still room for improvement in achieving a balance between precision and recall.

It is crucial to consider that the differences in dataset characteristics [38], [39] and the absence of k-fold cross-validation in some previous studies could contribute to variations in the observed performance metrics. Nonetheless, the proposed LSTM model proves to be a competitive and balanced solution, showcasing its effectiveness in addressing the classification task compared to the referenced methodologies.

CONCLUSIONS

This study successfully developed the APD-BayTM system, utilizing LSTM models optimized with Bayesian techniques, to predict the Air Quality Index (AQI) in Jakarta. The model demonstrated significant improvements in predictive performance, with precision increasing from 59.66% to 93.29%, recall from 56.26% to 91.41%, F1-Score from 55.03% to 91.89%, and accuracy from 80.29% to 95.90%. While our results are competitive with those of previous studies, the reliance on a single dataset from 2021 presents a limitation, potentially affecting the model's generalizability.

To enhance the robustness and adaptability of the APD-BayTM system, future research should incorporate more extensive datasets spanning multiple years and regions. Additionally, addressing class imbalance remains a critical area for further development, which could lead to a more equitable and generalizable model. This work lays a promising foundation for using advanced machine learning techniques to inform public and governmental decision-making in air pollution management.

ACKNOWLEDGMENT

This research was supported by the Department of Informatics, School of Computing, Telkom University, Indonesia and Human Centric (HUMIC) Engineering, School of Computing, Telkom University, Bandung, Indonesia.

REFERENCES

[1] P. K and P. Kumar, “A critical evaluation of air quality index models (1960–2021),” *Environ Monit Assess*, vol. 194, no. 5, p. 324, May 2022, doi: 10.1007/s10661-022-09896-8.

[2] M. F. Sanfia, S. Mandala, and E. Ariyanto, “Air Pollution Detection Based on Adaboost Ensemble Learning,” in *2023 International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, Aug. 2023, pp. 203–208. doi: 10.1109/ICoDSA58501.2023.10277254.

[3] M. Hardini, R. A. Sunarjo, M. Asfi, M. H. Riza Chakim, and Y. P. Ayu Sanjaya, “Predicting Air Quality Index using Ensemble Machine Learning,” *ADI Journal on Recent Innovation (AJRI)*, vol. 5, no. 1Sp, pp. 78–86, Aug. 2023, doi: 10.34306/ajri.v5i1Sp.981.

[4] IQAir, “Live most polluted major city ranking.” Accessed: Oct. 25, 2023. [Online]. Available: https://www.iqair.com/world-air-quality-ranking

[5] WHO, “Ambient (outdoor) air pollution,” 2022, Accessed: Jun. 14, 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health

[6] A. Singh, H. Joshi, A. Srivastava, R. Kumar, and N. Hasteer, “An Analysis of Polluted Air Consumption and Hazards on Human Health: A Study Towards System Design,” in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, IEEE, Jan. 2020, pp. 532–539. doi: 10.1109/Confluence47617.2020.9057848.

[7] S. Priya R.M. and P. Sathya, “Statistical Analysis of Air Pollutants in Ambient Air, Reality of Sensors and Corrective Measures in India,” in *2019 Innovations in Power and Advanced Computing Technologies (i-PACT)*, IEEE, Mar. 2019, pp. 1–6. doi: 10.1109/i-PACT44901.2019.8960010.

[8] “World Health Organization.” Accessed: Oct. 08, 2023. [Online]. Available: https://www.who.int/health-topics/air-pollution

[9] S. D. Permai, N. T. M. Sagala, A. Y. Zakiyyah, H. Tanty, and J. Harefa, “Multiclass Classification for Air Quality In Jakarta Using Support Vector Machine and Multi-Layer Perceptron Classifier,” in *2022 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS)*, IEEE, Sep. 2022, pp. 198–202. doi: 10.1109/AiDAS56890.2022.9918697.

[10] R. Muljana, L. D. Ayuningtyas, R. P. Daksa, S. F. Djamhari, M. A. Fiezayyan, and N. T. M. Sagala, “Air Pollution Prediction using Random Forest Classifier: A Case Study of DKI Jakarta,” in *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, IEEE, Feb. 2023, pp. 428–433. doi: 10.1109/ICCoSITE57641.2023.10127759.

[11] M. A. Rafif, G. Sanjaya Indrajaya, M. K. Al-Ghazi, J. Johnny, and N. T. M. Sagala, “Comparison of Decision Tree and Support Vector Machine for Predicting Jakarta Air Quality Index,” in *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, IEEE, Feb. 2023, pp. 381–385. doi: 10.1109/ICCoSITE57641.2023.10127855.

[12] N. H. Syafiuddin, S. Mandala, and N. D. W. Cahyani, “Detection Syn Flood and UDP Lag Attacks Based on Machine Learning Using AdaBoost,” in *2023 International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, Aug. 2023, pp. 36–41. doi: 10.1109/ICoDSA58501.2023.10276638.

[13] S. Fitriani, S. Mandala, and M. A. Murti, “Review of semi-supervised method for Intrusion Detection System,” in *2016 Asia Pacific Conference on Multimedia and Broadcasting (APMediaCast)*, IEEE, Nov. 2016, pp. 36–41. doi: 10.1109/APMediaCast.2016.7878168.

[14] D. Septyadi and S. Mandala, “Analysis of Home Security System Design Based on 4 PIR Sensors Using Deep Learning Method,” in *2023 International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, Aug. 2023, pp. 181–186. doi: 10.1109/ICoDSA58501.2023.10277453.

[15] B. N. Gerald Ergi and S. Mandala, “PIR Sensor-Based Intelligent Home Security System Design Analysis Using Machine Learning Methods,” in *2023 International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, Aug. 2023, pp. 175–180. doi: 10.1109/ICoDSA58501.2023.10277399.

[16] S. Rachmadi, S. Mandala, and D. Oktaria, “Detection of DoS Attack using AdaBoost Algorithm on IoT System,” in *2021 International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, Oct. 2021, pp. 28–33. doi: 10.1109/ICoDSA53588.2021.9617545.

[17] A. Syafiq Muhammad, S. Mandala, and P. H. Gunawan, "IOT-Based Pest Detection in Maize Plants Using Machine Learning," in *2023 International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, Aug. 2023, pp. 254–258. doi: 10.1109/ICoDSA58501.2023.10277633.

[18] S. Mandala *et al.*, "Enhanced Myocardial Infarction Identification in Phonocardiogram Signals Using Segmented Feature Extraction and Transfer Learning-Based Classification," *IEEE Access*, vol. 11, pp. 136654–136665, 2023, doi: 10.1109/ACCESS.2023.3338853.

[19] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[20] A. H. Almaliki, A. Derdour, and E. Ali, "Air Quality Index (AQI) Prediction in Holy Makkah Based on Machine Learning Methods," *Sustainability*, vol. 15, no. 17, p. 13168, Sep. 2023, doi: 10.3390/su151713168.

[21] S. Saminathan and C. Malathy, "Ensemble-based classification approach for PM2.5 concentration forecasting using meteorological data," *Front Big Data*, vol. 6, Jun. 2023, doi: 10.3389/fdata.2023.1175259.

[22] K. M. Babu and J. R. Beulah, "Air Quality Prediction based on Supervised Machine Learning Methods," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 9S4, pp. 206–212, Oct. 2019, doi: 10.35940/ijitee.I1132.0789S419.

[23] A. Pant, S. Sharma, M. Bansal, and M. Narang, "Comparative Analysis of Supervised Machine Learning Techniques for AQI Prediction," in *2022 International Conference on Advanced Computing Technologies and Applications, ICACTA 2022*, 2022. doi: 10.1109/ICACTA54488.2022.9753636.

[24] A. Attaallah and R. Ahmad Khan, "SMOTEDNN: A Novel Model for Air Pollution Forecasting and AQI Classification," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1403–1425, 2022, doi: 10.32604/cmc.2022.021968.

[25] M. G. Ragab *et al.*, "A Novel One-Dimensional CNN with Exponential Adaptive Gradients for Air Pollution Index Prediction," *Sustainability*, vol. 12, no. 23, p. 10090, Dec. 2020, doi: 10.3390/su122310090.

[26] T. Toharudin *et al.*, "Boosting Algorithm to Handle Unbalanced Classification of PM $_{2.5}$ Concentration Levels by Observing Meteorological Parameters in Jakarta-Indonesia Using AdaBoost, XGBoost, CatBoost, and LightGBM," *IEEE Access*, vol. 11, pp. 35680–35696, 2023, doi: 10.1109/ACCESS.2023.3265019.

[27] E. Vlachou, C. Karras, A. Karras, D. Tsolis, and S. Sioutas, "EVCA Classifier: A MCMC-Based Classifier for Analyzing High-Dimensional Big Data," *Information*, vol. 14, no. 8, p. 451, Aug. 2023, doi: 10.3390/info14080451.

[28] G. V. S. S. N. S. Sarma, B. R. Reddy, P. M. Nirgude, and P. V. Naidu, "Performance Assessment of Customized LSTM based Deep Learning Model for Predictive Maintenance of Transformer," *International Journal of Electrical and Electronics Research*, vol. 11, no. 2, pp. 389–400, Jun. 2023, doi: 10.37391/ijeer.110220.

[29] G. Sun, C. Jiang, X. Wang, and X. Yang, "Short-term building load forecast based on a data-mining feature selection and LSTM-RNN method," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 15, no. 7, pp. 1002–1010, Jul. 2020, doi: 10.1002/tee.23144.

[30] M. Massaoudi, S. S. Refaat, I. Chihi, M. Trabelsi, F. S. Oueslati, and H. Abu-Rub, "A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for Short-Term Load Forecasting," *Energy*, vol. 214, p. 118874, Jan. 2021, doi: 10.1016/j.energy.2020.118874.

[31] R. Shi, X. Xu, J. Li, and Y. Li, "Prediction and analysis of train arrival delay based on XGBoost and Bayesian optimization," *Appl Soft Comput*, vol. 109, p. 107538, Sep. 2021, doi: 10.1016/j.asoc.2021.107538.

[32] H. V. Nguyen and H. Byeon, "Prediction of Out-of-Hospital Cardiac Arrest Survival Outcomes Using a Hybrid Agnostic Explanation TabNet Model," *Mathematics*, vol. 11, no. 9, p. 2030, Apr. 2023, doi: 10.3390/math11092030.

[33] A. Kulshrestha, V. Krishnaswamy, and M. Sharma, "Bayesian BILSTM approach for tourism demand forecasting," *Ann Tour Res*, vol. 83, p. 102925, Jul. 2020, doi: 10.1016/j.annals.2020.102925.

[34] "Indeks Standar Pencemaran Udara (ISPU) Tahun 2021." Accessed: Oct. 07, 2023. [Online]. Available: https://data.jakarta.go.id/dataset/indeks-standar-pencemaran-udara-ispu-tahun-2021

[35] D. Chaniago, A. Zahara, and I. S. Ramadhani, "INDEKS STANDAR PENCEMAR UDARA (ISPU) SEBAGAI INFORMASI MUTU UDARA AMBIEN DI INDONESIA," Direktorat Pengendalian Pencemaran Udara, Direktorat Jenderal Pengendalian Pencemaran dan Kerusakan Lingkungan, Kementerian Lingkungan Hidup dan Kehutanan.

[36] "Tensorflow." Accessed: Dec. 06, 2023. [Online]. Available: https://www.tensorflow.org/

[37] M. Schmidt, S. Safarani, J. Gastinger, T. Jacobs, S. Nicolas, and A. Schulke, "On the Performance of Differential Evolution for Hyperparameter Tuning," in *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, Jul. 2019, pp. 1–8. doi: 10.1109/IJCNN.2019.8851978.

[38] P. Tsangaratos and I. Ilia, "Comparison of a logistic regression and Naïve Bayes classifier in landslide susceptibility assessments: The influence of models complexity and training dataset size," *Catena (Amst)*, vol. 145, pp. 164–179, Oct. 2016, doi: 10.1016/j.catena.2016.06.004.

[39] B. Kalantar, B. Pradhan, S. A. Naghibi, A. Motevalli, and S. Mansor, "Assessment of the effects of training data selection on the landslide susceptibility mapping: a comparison between support vector machine (SVM), logistic regression (LR) and artificial neural networks (ANN)," *Geomatics, Natural Hazards and Risk*, vol. 9, no. 1, pp. 49–69, Jan. 2018, doi: 10.1080/19475705.2017.1407368.

AUTHORS BIOGRAPHY

Raey Faldo

was born in Lampung, Indonesia, in 1999. He received a diploma in software engineering from Bandung State Polytechnic, Bandung, in 2021. Currently a final year student at the Department of Informatics, School of Computing, Telkom University, Bandung, Indonesia. His research interests include issues related to artificial intelligence, the Internet of Things (IoT), and software engineering.

Satria Mandala

(Member, IEEE) He received his PhD degree in computer science from Universiti Teknologi Malaysia. He is the director of Human Centric Engineering (Humic Eng) and a member of the School of Computing (SoC)-Telkom University, Indonesia. His research interests include issues related to wireless network security, the Internet of Things (IoT), biomedical engineering, and digital signal and image processing.

Mohd Shahrizal Sunar

a Professor at the Faculty of Computing in Universiti Teknologi Malaysia (UTM), specializing in Extended Reality (XR), earned his PhD from the National University of Malaysia. Dr Shahrizal's educational journey includes obtaining an MSc in Computer Graphics and Virtual Environment from The University of Hull, United Kingdom, and a BSc degree in Computer Science, majoring in Computer Graphics, from Universiti Teknologi Malaysia.

He holds the director position at the Institute of Human Centered Engineering (iHumEn). He is the founding Director of the Media and Game Innovation Centre of Excellence (MaGICX), focusing on augmented reality, virtual reality, computer vision, gamification, user interaction/user experience (UI/UX), and creative content technology.

Salim M. Zaki

Salim M Zaki is currently assistant professor in computer science at Ministry of Higher Education and Scientific Research, Directorate of Studies, Planning and Follow-up, Iraq. Received his PhD from University Technology of Malaysia (UTM). His main research interest is related to information technology, information security, intelligent systems and mobile networks.

https://doi.org/10.25077/jnte.v13n2.1183.2024