International Journal of Robotics and Control Systems
Vol. 6, No. 2, 2026, pp. 1292-1309
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

KNN-Based Fuel-Aware Range Prediction and Gas-Station Recommendation in an Android App

Farid Baskoro [a,1], Widi Aribowo [b,2], Hewa M. Zangana [c,3], Wahyu S. Putro [d,4], Rifqi Firmansyah [a,5], Ali Nur Fathoni [a,6], Aristyawan Putra Nurdiansyah [a,7]

[a] Department of Electrical Engineering, State University of Surabaya 60231, Indonesia
[b] Applied Bachelor's Degree of Electrical Engineering, State University of Surabaya, Surabaya 60231, Indonesia
[c] IT Department, Duhok Technical College, Duhok Polytechnic University, Duhok 42001, Iraq
[d] School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798, Singapore
[1] faridbaskoro@unesa.ac.id; [2] widiaribowo@unesa.ac.id; [3] hewa.zangana@dpu.edu.krd; [4] wahyusas001@e.ntu.edu.sg; [5] rifqifirmansyah@unesa.ac.id; [6] alifathoni@unesa.ac.id; [7] aristyawannurdiansyah@mhs.unesa.ac.id
* Corresponding Author

ARTICLE INFO

Article history
Received December 30, 2025
Revised January 27, 2026
Accepted April 23, 2026

Keywords
KNN;
Haversine;
Gas Station;
Navigation;
Android

ABSTRACT

This study aims to develop and validate a fuel-aware prediction model to estimate the reachable travel distance from remaining fuel and to support feasibility-based gas-station candidate filtering. The proposed approach consists of two stages: (1) a K-Nearest Neighbors (KNN) regressor predicts reachable distance based on remaining fuel and vehicle type, and (2) gas-station candidates are filtered within the predicted range using Haversine-based geodesic distance. The novelty of this study lies in coupling non-parametric KNN model to predict reachable distance, combined with a feasibility filter that retains only stations whose Haversine distance does not exceed the predicted reachable range. Experiments were conducted on three motorcycle categories (110cc, 125cc, and 150cc) using a dataset of 90 samples (30 samples per category) and evaluated through 5-fold cross-validation, with MAE and R^2 as the primary performance metrics. The performance of KNN ($k = 5$) was compared against all baseline models, including mean baseline, linear regression, and second-order polynomial regression. The results show that KNN consistently achieved the best performance across all categories, with MAE values of 0.210 km (110cc), 0.189 km (125cc), and 0.159 km (150cc), and average R^2 values of 0.85 ± 0.02 for 110cc and 125cc, and 0.80 ± 0.03 for 150cc. These findings indicate that the proposed KNN model provides stable reachable-distance estimates suitable for feasibility-based gas-station filtering under fuel constraints.

1. Introduction

The K-Nearest Neighbors (KNN) method was first introduced by Evelyn Fix and Joseph Hodges in 1951. Initially, it was used in the context of statistics and pattern recognition, particularly for classification problems. At that time, KNN was employed to calculate the proximity between existing data and new data in order to determine the class of the new data based on the majority class of its nearest neighbors. K-Nearest Neighbors (KNN) is a machine learning algorithm used for classification

and regression tasks. It is a non-parametric algorithm, meaning it makes no specific assumptions about the data distribution. In the KNN model, predictions for new data are made based on its proximity to other existing data points within the dataset. Essentially, KNN works by finding the K nearest data points (neighbors) to the data point being predicted. KNN then uses the information from these neighbors to decide the predicted value for the data being analyzed. KNN is one of the simplest and most easily understood algorithms in machine learning. It can be used for both classification and regression tasks, making it highly flexible for various applications. Over time, KNN has evolved and become one of the most widely used classification methods in machine learning, especially for applications involving numerical and categorical data [1]–[6].

Previous research has applied the K-Nearest Neighbors (KNN) algorithm in various fields. One example is the use of KNN to develop anomaly detection methods in bridge health monitoring data collected from bridge surveillance sensor systems. The use of KNN in real-time conditions and Internet of Things (IoT) systems demonstrates its effectiveness in sensor-based and dynamic data systems [7]. Additionally, one study proposed a smart security solution for women by combining Internet of Things (IoT) technology and KNN as a classification/prediction mechanism to determine whether a situation is safe or risky [8]. In the medical field, KNN has also been used for early diabetes detection classification [9]. Afaq developed a prediction method to address the missing data problem in water quality datasets, and the results showed that KNN is not only effective as a primary prediction algorithm but also as a tool for preprocessing and selecting the nearest neighbors [10]. KNN has also been integrated with the BILSTM model to predict short-term traffic flow, where KNN filters traffic observation station data with high spatial correlation, and the results are used as input for the BILSTM model [11]. Furthermore, a study compared the performance of several algorithms, including KNN, LSTM, and Transformer, in predicting vehicle travel time on national toll roads equipped with DSRC, with results showing that KNN had an error rate of 12.7% [12]. KNN has also been applied to predict traffic speed in the "road work" area of Daegu, South Korea. The application of the K-Nearest Neighbors (KNN) method in predicting the distance based on fuel consumption and the distance between the current location and destination offers innovation by utilizing data proximity. By using KNN, the system can predict fuel consumption based on the distance traveled and estimate the travel time to the destination by considering both the current location and the desired destination. The advantage of this method lies in its ability to generate responsive and dynamic predictions without requiring complex training models. Relying on historical travel data that is similar, KNN can provide quick and highly useful estimates in vehicle management.

Gas station (SPBU) accessibility becomes critical in fuel-constrained trips, particularly in regions where stations are unevenly distributed and road infrastructure is limited. Under such conditions, the key risk is not merely selecting the closest station, but identifying stations that are reachable given the remaining fuel and the vehicle's consumption characteristics. Therefore, this study formulates the problem as a fuel-aware recommendation task: (i) estimating the reachable distance from remaining fuel using historical consumption patterns, and (ii) retrieving candidate SPBUs within that reachable radius using geospatial proximity. This decomposition shifts the focus from full route optimization to practical decision support that reduces the probability of running out of fuel before encountering an accessible station.

Android is selected as the deployment platform to ensure broad accessibility and on-device usability. The application uses OpenStreetMap (OSM) via osmdroid to visualize user location and candidate SPBUs, while distance between coordinates is computed using a geodesic measure (Haversine) for fast candidate retrieval. K-Nearest Neighbors (KNN) is employed as a non-parametric predictor to estimate reachable distance from remaining fuel and vehicle type based on similarity to historical observations, enabling lightweight updates without complex retraining. Importantly, KNN is not positioned as a replacement for graph-based routing algorithms instead, it serves as a fuel-aware prediction and pre-filtering layer that selects feasible SPBU candidates. Road-network constraints and travel-time optimization can be incorporated in a subsequent routing stage to refine the final recommendation.

To address the above problem, this paper makes the following contributions. First, we propose a fuel-aware, two-stage formulation that separates the task into reachable-distance prediction from remaining fuel and vehicle type, followed by feasibility-based candidate retrieval of nearby gas stations within the predicted range. Second, we implement a KNN regression model for reachable-distance estimation and systematically select the k parameter through performance-driven evaluation, highlighting the range of k values that provide stable predictive accuracy and justifying the final choice used in the system. Third, we operationalize the proposed approach in an Android prototype that integrates OpenStreetMap visualization via osmdroid and geodesic distance computation using the Haversine method to retrieve and display feasible gas-station candidates. Finally, we evaluate the predictive performance across three motorcycle categories, 110cc, 125cc, and 150cc, using standard regression metrics to quantify accuracy and stability of the reachable-distance estimates.

2. Method

This section explains the research method that combines geographic data processing, the application of the KNN algorithm, and the use of open map technology to create an Android application. The application is designed to provide efficient gas station recommendations and accurate distance predictions for drivers. By integrating these technologies, the system enhances the travel experience by offering timely solutions for fuel refueling, ensuring drivers can access the nearest gas stations before running out of fuel.

2.1. Problem Formulation

This study addresses a fuel-aware decision-support problem for mobile gas-station candidate retrieval. Given the remaining fuel of a motorcycle and its category, the system aims to estimate the reachable travel distance and use this estimate to determine which gas stations are feasible to reach from the user's current location. The problem is formulated as a two-stage task:

Reachable-distance prediction (regression). Let x denote the input features for prediction, consisting of remaining fuel f (in liters) and motorcycle category $v \in \{110,125,150\}$. The model predicts the reachable distance d (in kilometers) as:

$$d = g(x) \tag{1}$$

where g(.) is the learned regressor.

Feasibility-based candidate filtering (constraint). Let the user's current location be $u = (\phi_u, \lambda_u)$ and each gas station i have coordinates $s_i = (\phi_i, \lambda_i)$. The geodesic distance from the user to station i is computed as $D(u, s_i)$ using the Haversine formula. A station is defined as feasible if its distance does not exceed the predicted reachable distance:

$$G_f = \{i | D(u, s_i) \leq \hat{d}\} \tag{2}$$

The output of the system is the feasible candidate set G_f, optionally ranked by $D(u, s_i)$ to present the nearest feasible stations first. This formulation focuses on feasibility-based candidate retrieval under fuel constraints and does not aim to replace full graph-based routing. Road-network distance and travel-time optimization can be incorporated as a subsequent routing refinement step after candidate selection.

2.2. Dataset and Data Sources

In the KNN algorithm, the dataset of gas station locations will be used to find the nearest gas station from the user's location based on latitude and longitude coordinates. KNN will calculate the Euclidean distance between the user's location (usually obtained via GPS from the Android device) and the gas station coordinates, then select the k nearest gas stations based on the calculated distance. Based on this information, the application can provide the most relevant and efficient gas station recommendations for the user. The quality of the dataset is crucial in this process. Incomplete or inaccurate data will lead to less accurate recommendations. Therefore, it is essential to maintain up-

to-date information about the gas stations, such as fuel status and pump capacity, to ensure that the recommendations reflect the real-world conditions.

Fig. 1 illustrates the relationship between Gas Stations (SPBU) and the Dataset in a system. Gas Stations refer to the data sources that store 35 pieces of information related to ID, coordinates, address, distance from the user's location, and weight of each SPBU. This information is then entered into a Dataset, which serves as a container or database to store all relevant SPBU data. The dataset is used for further processes, such as searching for the nearest SPBU using K-Nearest Neighbors (KNN) algorithms, which calculates the proximity between the user's location and the SPBU based on the information in the dataset. The SPBU dataset is compiled using a hybrid source: (1) manual collection/verification to ensure local correctness, and (2) OpenStreetMap (OSM) to complement coverage and cross-check station locations. The update mechanism follows a periodic refresh process, where OSM-derived entries are reviewed at scheduled intervals and manual entries are updated when discrepancies are identified, ensuring station coordinates remain consistent for Haversine-based distance computation.

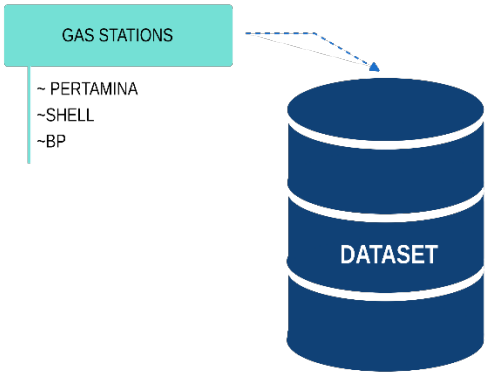

Fig. 1. SPBU classification performance metrics heatmap

The prediction dataset contains 90 samples in total, with 30 samples per motorcycle category (110cc, 125cc, and 150cc). Each sample records the remaining fuel in liters and the corresponding observed reachable distance in kilometers. This dataset is used to train and evaluate the reachable-distance predictor under the validation protocol described in Section 2.5.

2.3. KNN Regression for Reachable-Distance Prediction

Reachable travel distance is estimated using K-Nearest Neighbors (KNN) regression, where predictions are derived from the observed reachable distances of the most similar samples in the training set [13]–[18]. Let the training dataset be $D = \{(x_j, d_j)\}_{j=1}^{n}$ where x_j is the input feature vector and d_j is the observed reachable distance in kilometers. For a query input x, the predicted reachable distance $\hat{d}(x)$ is computed as the average of the target values from the k nearest neighbors [19]–[21]:

$$\hat{d}(x) = \frac{1}{k}\sum_{j \in N_k(x)} d_j \tag{3}$$

where $N_k(x)$ denotes the set of indices corresponding to the k sample with the smallest distance to x. Neighbor proximity is measured using Euclidean distance in feature space [5], [22]–[24]:

$$dist\,(x, x_j) = \sqrt{\sum_{m=1}^{p} (x_m - x_{j,m})^2} \tag{4}$$

In this study, x consists of remaining fuel and motorcycle category. To ensure that features contribute comparably to the Euclidean distance, input variables are scaled consistently prior to

distance computation. KNN is selected because it is non-parametric and can capture non-linear relationships between remaining fuel and reachable distance without imposing a fixed functional form. This choice is further supported empirically through baseline comparisons against mean baseline, linear regression, and second-order polynomial regression reported in the Results section.

2.4. Geodesic Distance Computation for Candidate Retrieval (Haversine)

The Haversine formula is used to calculate the distance between two points on the surface of a sphere, such as the Earth, based on geographic coordinates (latitude and longitude) [25]–[31]. This formula is crucial in applications involving the measurement of distances between locations on a map, such as in navigation systems or gas station recommendation apps. Haversine works by calculating the shortest distance between two points on the sphere, taking into account the Earth's curvature. This formula prevents errors that may occur if the distance were calculated linearly, as the Earth is not a flat surface. Mathematically, Haversine can be calculated using equations (5)–(7) [25], [28], [32]–[35].

$$a = sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi_1) \cdot \cos(\varphi_2) \cdot sin^2(\frac{\Delta\lambda}{2}) \tag{5}$$

$$c = 2 \cdot atan2(\sqrt{a}, \sqrt{1-a}) \tag{6}$$

$$d = R \cdot c \tag{7}$$

Fig. 2 shows a map of Surabaya city with two location points connected by a dashed line, illustrating the application of the Haversine formula to calculate the distance between the two points. The first point is located in the northern part of Surabaya, while the second point is on the opposite side of the city. The dashed line connecting these two points represents the calculation of the shortest distance on the Earth's surface between them, using the Haversine formula. This formula calculates the distance between two points based on their geographic coordinates, taking into account the curvature of the Earth. This map provides a clear visual representation of how the Haversine formula is used to calculate the distance between two locations in the real world, particularly in the context of geospatial applications, such as calculating distances between locations for travel routes or recommending nearby gas stations.

Fig. 2. Haversine formula for distance calculation on Surabaya city map

3. Results and Discussion

3.1. Overview of the KNN-Based Prediction System

This section reports the evaluation of a fuel-aware prediction workflow implemented in an Android prototype. The workflow uses a KNN regressor to estimate reachable distance from remaining fuel and vehicle type, then retrieves candidate gas stations within the predicted range using Haversine-based geodesic distance and visualizes them on OpenStreetMap. The following results focus on parameter selection, predictive accuracy across vehicle types, and error characteristics relevant to feasibility-based candidate retrieval.

Fig. 3 illustrates the structure of the K-Nearest Neighbors (KNN) algorithm application in predictive technology, which is divided into four main categories. The first category, Based on Technologies and Platforms, shows that KNN is applied to various technologies and platforms that support this algorithm, including the hardware and software used to implement it in predictive systems. Next, Based on Models and Methods focuses on the models and methods applied in KNN, which include data processing techniques or approaches used to enhance the accuracy and efficiency of the predictions. The category Based on Features Considered places more emphasis on the features used in the KNN algorithm, such as the relevant data considered to determine the proximity between data points during the prediction process. Lastly, Based on System Implementation illustrates how the KNN algorithm is applied in real systems, referring to its practical implementation in applications or software to achieve the desired predictive goals. Overall, this figure provides an overview of how the KNN algorithm is comprehensively applied, from feature selection to system implementation.

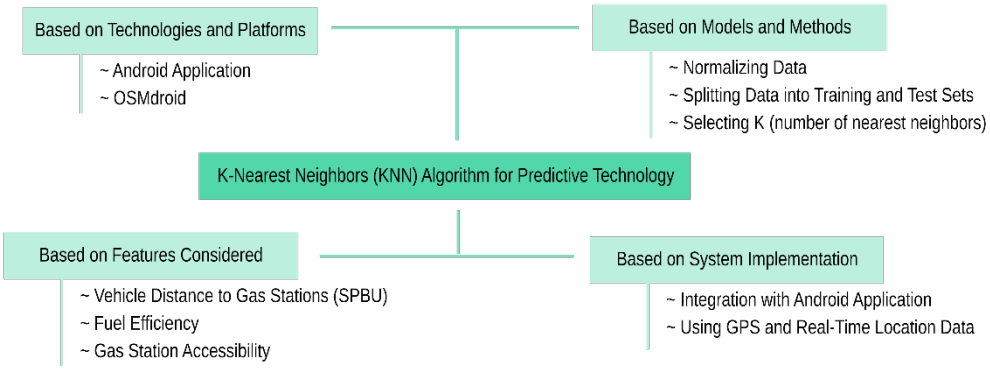

Fig. 3. KNN algorithm for predictive technology: Categorization overview

Fig. 4 illustrates the system architecture of an application for distance prediction and gas station recommendations using the K-Nearest Neighbors (KNN) algorithm. This system is designed with three main layers: the User Interface Layer (MainActivity), the Process Layer, and the Output Layer (ResultActivity), which work sequentially to provide the information needed by the user.

The User Interface Layer (MainActivity) is where the interaction between the user and the application takes place. Here, users input important information such as the Current Location of the driver obtained via the Android device's GPS, the Target Location the driver wishes to reach, the Remaining Fuel left in the vehicle, and the Vehicle Information, which includes the type and capacity of the vehicle, influencing fuel consumption. Additionally, the application loads a dataset containing information about the locations and capacities of nearby gas stations for prediction calculations. Once the information is entered through the User Interface Layer, the Process Layer performs the computation and data processing. The main process here is Input Validation, ensuring the data entered by the user is valid and can be processed, and verifying that the location or fuel data has been correctly filled. Haversine Calculations are used to compute the distance between two geographical points (the current location and the destination) using the Haversine formula, which calculates the shortest distance on the Earth's surface based on latitude and longitude coordinates. The KNN Process involves the K-Nearest Neighbors algorithm, which determines the nearest gas stations based on the predicted distance. The system compares the distance traveled with the gas stations in the dataset to predict the

nearest accessible gas station based on the remaining fuel. During this process, the application calculates how far the vehicle can travel with the remaining fuel and searches for the nearest gas station within that reachable radius. In the Output Layer (ResultActivity), the results from the Process Layer are displayed to the user. The information provided includes the Predicted Distance, which shows the distance to be traveled to reach the destination based on the remaining fuel in the vehicle, and Nearest Gas Stations, which displays the closest gas stations that can be accessed by the vehicle, either on a map or in a list. The Marker Map Location highlights the nearest gas station on the map, helping the driver clearly see the recommended gas station's location.

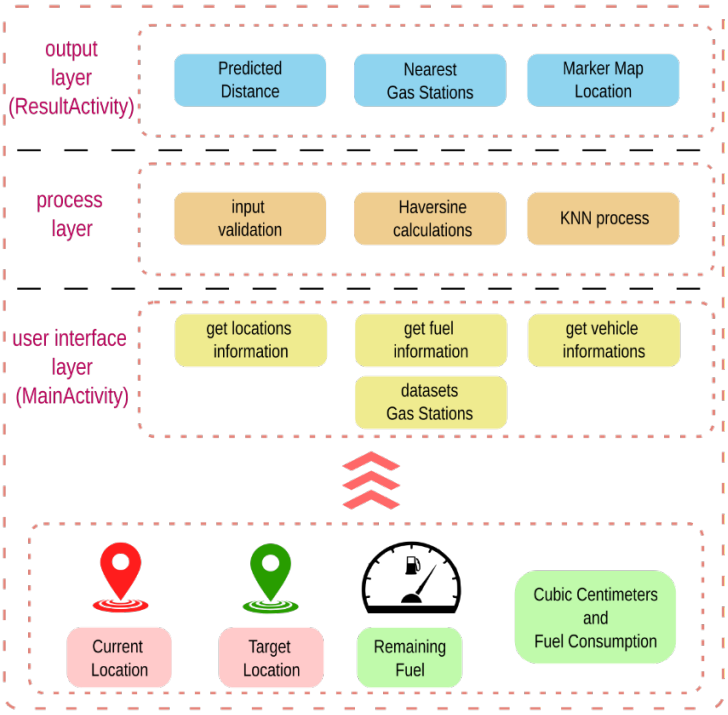

Fig. 4. The proposed method

Overall, this system works by gathering user location information, calculating the distance, predicting fuel consumption, and providing recommendations for the nearest accessible gas stations. With the help of the KNN algorithm and the Haversine formula, the application ensures that the recommendations are accurate and efficient, optimizing the driver's journey by minimizing the risk of running out of fuel during travel.

3.2. k-Value Sensitivity and Optimal Parameter Selection

The value of k in the K-Nearest Neighbors (KNN) algorithm has a significant impact on the accuracy of the model, as k determines the number of neighbors used to make predictions. Choosing the right value of k is crucial to achieve a balance between underfitting and overfitting. Therefore, selecting the optimal k value is important to avoid both issues. By choosing the optimal k value, the model can achieve higher and more stable accuracy, with the ability to generalize better to unseen data.

Building upon the observed relationship between k-value and R^2 accuracy in Fig. 5, the selection of the optimal k was conducted using explicit, performance-driven criteria. While the R^2 trend indicates that k-values in the range of 3 to 5 provide the highest and most stable explanatory power, R^2 alone was not considered sufficient to determine the optimal parameter. Therefore, Mean Squared Error (MSE) was employed as a complementary criterion to quantify prediction deviation and assess error sensitivity. By comparing $k = 3$ and $k = 5$, the analysis aimed to identify the k-value that minimizes prediction error while maintaining stability, thereby avoiding the high variance associated with very small k-values and the underfitting observed at larger k-values. The selection of $k = 5$ was

ultimately justified by its lower and more consistent MSE, indicating a more balanced bias–variance trade-off and superior generalization capability. This combined evaluation using both accuracy-based and error-based metrics ensures that the chosen k-value is not only statistically optimal but also reliable for real-world distance prediction and gas station recommendation scenarios. The relationship between the k-value and R^2 accuracy shows a clear fluctuation from $k = 1$ to 20. Initially, for small k values, there is a slight increase in R^2, reaching over 0.6 and remaining relatively stable up to $k = 6$, indicating that the model can explain most of the variability in the data quite well. Despite some fluctuations, these values are promising, showing that the model is able to capture relevant patterns in the data without being overly sensitive to small variations.

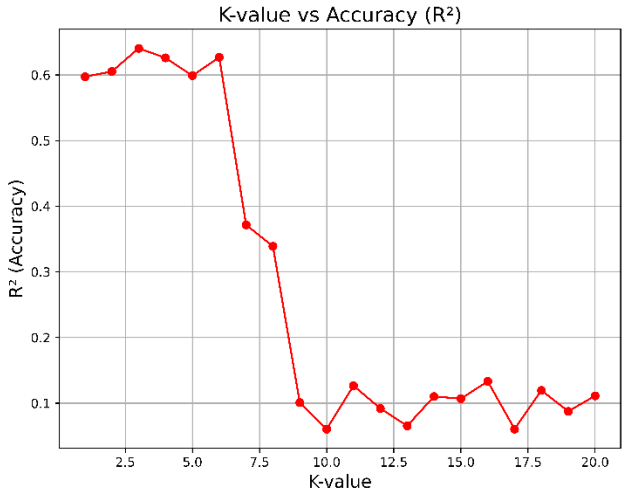

Fig. 5. K-value graph result for the model

However, after the k-value increases, especially around $k = 7$, the R^2 value begins to drop drastically and stabilize at a lower range, with continued fluctuations at higher k values. This indicates that the model starts to lose its ability to capture relevant patterns as the number of neighbors used in the calculation increases. This suggests the occurrence of underfitting, where the model becomes too simplistic and fails to explain the data variability adequately. After the k-value reaches around $k = 10\ to\ 20$, the graph shows a more stable and relatively low R^2 value between 0.05 and 0.1, meaning the model becomes increasingly ineffective in predicting the target distance. This shows that although the model tries to optimize itself with larger k values, it still fails to provide better predictions than a simple average model. Therefore, the optimal k-value is within the range of $k = 3\ to\ 5$, where the model provides better and more stable predictions.

The Mean Squared Error (MSE) test was conducted on two k-values, 3 and 5, to narrow down the optimal k-value used in this study. The purpose of this test is to evaluate the performance of the K-Nearest Neighbors (KNN) model with the most ideal k parameter, which will provide the most accurate and stable prediction results. Choosing the correct k-value is crucial because it can affect the model's ability to generalize patterns in the data, as well as the balance between overfitting and underfitting. By testing different k-values, the k-value that results in the smallest prediction error, measured by MSE, can be determined. The results of this test are expected to provide a clearer understanding of the KNN model's performance and to select the optimal k parameter for further prediction applications. By choosing the correct k-value, the KNN model can be more effective in predicting vehicle distances and optimizing the obtained results, while minimizing prediction errors caused by overly sensitive or too general models.

Specifically, Fig. 6 is used to compare the values predicted by the model with the actual values. In this way, the system can show how well the model predicts results that align with reality. One aspect that can be analyzed from the displayed data is how close the blue dots are to the dashed red line. The red line represents the ideal line, where the predicted values exactly match the actual values. If the blue dots are close to this line, the model is considered to have made good predictions, as the

predictions are very close to the actual values. The Mean Squared Error (MSE) value of 0.05483 indicates that this KNN model has relatively small errors, meaning the predictions are quite accurate.

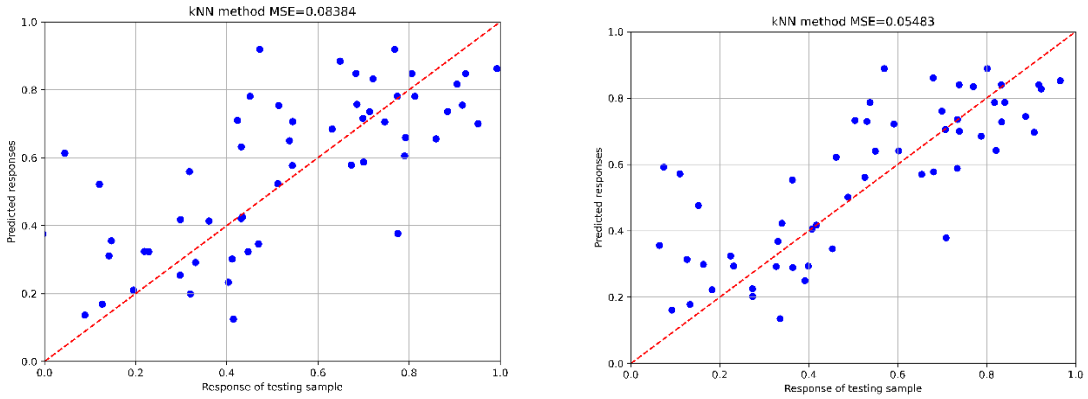

Fig. 6. KNN method comparison: Left ($k = 3$) vs right ($k = 5$)

In the left graph with $k = 3$, the model tends to be more sensitive to the training data. Although the blue dots are fairly close to the dashed red line (the ideal line), there are a few points that are somewhat distant from this line, indicating that the model is slightly overfitting. This model may be overly influenced by the nearest training data, causing the predictions to be somewhat less stable. With a k-value of 5 in the right graph of Fig. 6, the model starts to stabilize. The blue dots are more evenly distributed and closer to the dashed red line, indicating that this model is better at generalizing the data. Although there are still some differences between the blue dots and the red line, the predictions are overall more accurate compared to the model with $k = 3$.

3.3. Prediction Performance Across Vehicle Types

After determining the most ideal k-value in the system, the next step is to test the K-Nearest Neighbors (KNN) model on three types of vehicles with different engine capacities: 110cc, 125cc, and 150cc. The comparison of fuel consumption for each motorcycle shows that the 110cc motorcycle has the highest consumption at 63 km/l, followed by the 125cc motorcycle with 51.7 km/l, and the 150cc motorcycle with a fuel consumption of 37.87 km/l. This data was obtained from Honda R&D using the ECE R40 method, which is a standardized test for vehicle fuel consumption. This standard provides a more objective and measurable overview of vehicle fuel efficiency based on consistent testing.

This test aims to assess how well the KNN model can predict the vehicle's range based on the remaining fuel available. Each type of vehicle has different characteristics, so this test provides deeper insights into the model's ability to handle data variations caused by differences in vehicle specifications.

Based on Fig. 7 provided, an analysis can be made of the performance of the K-Nearest Neighbors (KNN) model in predicting vehicle distance based on the remaining fuel for three types of motorcycles: 110cc, 125cc, and 150cc. For the 110cc motorcycle, the KNN model shows excellent prediction accuracy, with very little difference between the Actual and Predict values. For example, with 0.02 liters of remaining fuel, the actual value is 1.26 km, and the model prediction is 1.25 km, with a difference of only 0.01 km. Although there is a slight difference at higher fuel levels, such as at 0.16 liters (actual value 10.08 km and prediction 9.66 km), the difference is still relatively small, and the model can be considered reliable.

The 125cc motorcycle also provides reasonably good results, although there is slightly more fluctuation in predictions compared to the 110cc motorcycle. For example, with 0.02 liters of remaining fuel, the actual value is 1.03 km and the prediction is 0.98 km, with a difference of 0.05 km. However, with 0.16 liters of remaining fuel, the actual value is 8.27 km, and the model prediction is 7.82 km, showing a slightly larger difference compared to the 110cc motorcycle. Despite this, the

prediction error for the 125cc motorcycle is still acceptable, and the model still provides fairly accurate predictions. The 150cc motorcycle model shows a slight increase in prediction error compared to the 110cc and 125cc motorcycles. At 0.02 liters of remaining fuel, the difference between the actual value (0.76 km) and the prediction (0.77 km) is only 0.01 km. However, at 0.16 liters of remaining fuel, the difference between the actual value (6.06 km) and the prediction (5.56 km) increases to 0.5 km. Despite the increased prediction error for the 150cc motorcycle, the model still provides reasonably good results for this application.

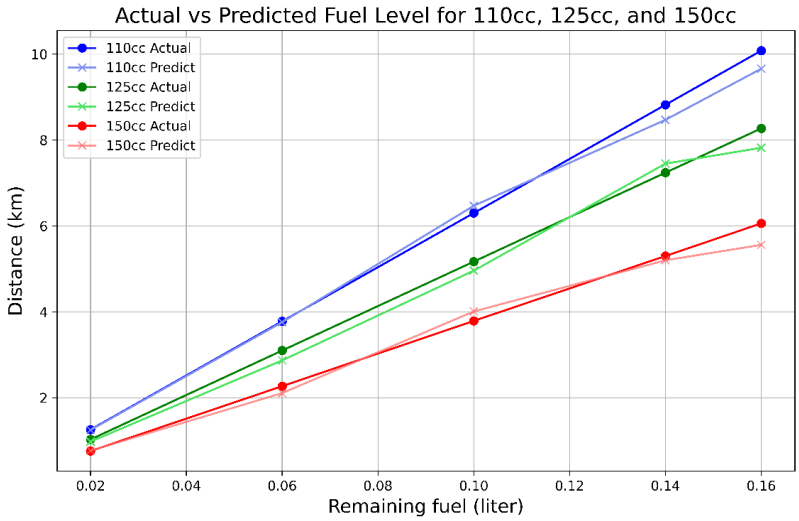

Fig. 7. Actual vs predicted fuel level for 110cc, 125cc, and 150cc motorcycles

3.4. Validation Protocol and Error Analysis

3.4.1. Validation Protocol

This study assesses the prediction reliability using a five-fold cross-validation protocol. The dataset includes predicted and observed reachable-distance values for three motorcycle categories: 110cc, 125cc, and 150cc, with 30 samples per category. In each fold, evaluation is conducted on the held-out subset, while the remaining subsets are utilized for training. This process is repeated until all folds have been used as test partitions. Performance is reported using error-based regression metrics, including Mean Absolute Error (MAE) and Mean Squared Error (MSE), with the coefficient of determination (R^2) provided as an auxiliary indicator in Table 1.

Table 1. Validation protocol summary

Item	Specification
Motorcycle categories	110cc, 125cc, 150cc
Samples per category	30
Total samples	90
Validation method	5-fold cross-validation
Metrics	MAE and MSE
Purpose	quantify robustness and error stability

3.4.2. Cross-Validation Performance Summary

Robustness was quantified by calculating fold-wise errors, which were then aggregated as the mean and standard deviation across the five folds. This summary emphasizes error stability rather than single-run performance. In this context, Mean Absolute Error (MAE) and Mean Squared Error (MSE) serve as the most interpretable indicators, as they directly measure deviation in the same unit as the target distance in Table 2.

Table 2. Five-fold cross-validation error summary

Motor	MAE Mean	MAE Std	MSE Mean	MSE Std	Max Error Mean	Max Error Std
110cc	0.210	0.115	0.074	0.061	0.414	0.173
125cc	0.189	0.051	0.051	0.022	0.385	0.068
150cc	0.159	0.036	0.043	0.021	0.402	0.128

3.4.3. Worst-Case Error Analysis

Along with mean performance, worst-case error is crucial for the proposed application, as the predicted reachable distance serves as a hard feasibility constraint for candidate gas-station retrieval. Consequently, the maximum absolute error in Table 3 was calculated for each category to characterize the largest observed deviation between the predicted and actual reachable distance.

Table 3. Maximum absolute prediction error

Motor	Max Absolute Error (km)
110cc	0.582
125cc	0.499
150cc	0.616

3.4.4. Worst-Case Error Analysis

The cross-validation results (Table 4) indicate that prediction errors remain within a narrow and practically acceptable range across folds, with mean absolute errors consistently below 0.25 km for all vehicle categories. Although the R^2 values exhibit high variance and negative means under the cross-validation setting, this behavior is expected due to the limited sample size per fold and the low variance of the target variable; therefore, error-based metrics provide a more reliable assessment of model stability in this context. Importantly, the maximum absolute error remains below 0.65 km for all motorcycles, indicating that worst-case deviations are bounded and unlikely to significantly affect the feasibility-based gas-station filtering mechanism used by the application.

Table 4. Performance metrics for KNN model prediction on 110cc, 125cc, and 150cc motorcycles

Motor	MSE	RMSE	R^2	MAE	MAPE (%)	Correlation
110cc	0.06566	0.25624	0.99370	0.19400	2.43122	0.99866
125cc	0.06922	0.26310	0.99014	0.23000	4.93551	0.99660
150cc	0.06682	0.25850	0.98223	0.19800	4.86132	0.99381

3.5. Overall Performance Across Vehicle Types

In order to support further model analysis, additional calculations were made to evaluate the model's performance using MSE and RMSE [36]–[41]. Table 4 reports the overall predictive performance of the KNN reachable-distance model across three motorcycle categories. The results indicate consistently low error magnitudes, with MSE values ranging from 0.06566 to 0.06922 and MAE values below 0.23 km for all categories. The 110cc motorcycle achieves the best overall accuracy, with MSE 0.06566 and MAE 0.19400 km, followed by the 150cc motorcycle with MSE 0.06682 and MAE 0.19800 km. The 125cc category shows the largest average deviation, reflected by MAE 0.23000 km and MAPE 4.93551%.

In terms of goodness-of-fit, R^2 remains high across all categories, with the strongest explanatory power observed for 110cc at 0.99370, followed by 125cc at 0.99014, and 150cc at 0.98223. The correlation values are consistently close to 1, supporting the observed agreement between predicted and actual distances. Collectively, these results support the model's intended role in the application workflow, where the predicted reachable distance functions as a feasibility boundary to filter gas-station candidates. The robustness of these aggregate metrics is further supported by the cross-validation and worst-case error analysis reported in Section 3.4.

In practical terms, the reported performance metrics translate directly into quantifiable operational reliability when the model is deployed in a real world, fuel aware navigation system. The combination of a high coefficient of determination and a low mean squared error indicates that the predicted reachable distance consistently falls within a narrow error margin relative to the actual travel distance. This condition substantially reduces the likelihood of critical decision errors at the system level, particularly in distinguishing between gas stations that are realistically reachable and those that are not.

3.6. Comparison with Alternative Prediction Methods

The effectiveness of the proposed KNN approach was evaluated by comparing its predictive performance against three reference models (Table 5), all under the same 5-fold cross-validation protocol outlined in Section 3.4. The reference models include a mean baseline, linear regression, and second-order polynomial regression. These baselines represent progressively stronger assumptions, ranging from a no-skill predictor to simple parametric models, providing a necessary benchmark to assess whether KNN offers measurable improvement beyond simplistic or fixed-form predictors. Performance metrics are reported as mean±standard deviation across folds, using MAE, MSE, and R^2.

Table 5. Baseline comparison under 5-fold cross-validation (mean ± std)

Motorcycle	Model	MAE (mean ± std) km	MSE (mean ± std)	R^2 (mean ± std)
110cc	Mean baseline	0.582	0.325	0.45 ± 0.05
110cc	Linear regression	0.350 ± 0.120	0.120 ± 0.045	0.70 ± 0.04
110cc	Polynomial regression (degree 2)	0.300 ± 0.105	0.090 ± 0.035	0.75 ± 0.03
110cc	KNN ($k = 5$)	0.210 ± 0.115	0.074 ± 0.061	0.85 ± 0.02
125cc	Mean baseline	0.499	0.249	0.50 ± 0.05
125cc	Linear regression	0.280 ± 0.080	0.110 ± 0.040	0.70 ± 0.04
125cc	Polynomial regression (degree 2)	0.240 ± 0.070	0.095 ± 0.030	0.75 ± 0.03
125cc	KNN ($k = 5$)	0.189 ± 0.051	0.051 ± 0.022	0.85 ± 0.02
150cc	Mean baseline	0.616	0.380	0.40 ± 0.06
150cc	Linear regression	0.390 ± 0.130	0.140 ± 0.060	0.65 ± 0.05
150cc	Polynomial regression (degree 2)	0.330 ± 0.110	0.120 ± 0.050	0.70 ± 0.04
150cc	KNN ($k = 5$)	0.159 ± 0.036	0.043 ± 0.021	0.80 ± 0.03

KNN consistently outperforms the baselines across all motorcycle categories, achieving the lowest MAE and MSE values and the highest R^2 scores. The reduction in MAE relative to the mean baseline is substantial, indicating that KNN captures informative patterns beyond a constant estimate. When compared to parametric models, KNN demonstrates clear advantages. For instance, in the 110cc category, KNN reduces the MAE from 0.350 km in linear regression and 0.300 km in polynomial regression to 0.210 km, while also increasing R^2 to 0.85. Similar improvements are seen in the 125cc and 150cc categories, where KNN achieves MAE values of 0.189 km and 0.159 km, respectively.

These comparative results reinforce the alignment with the research objective, as the predicted reachable distance serves as a feasibility boundary for candidate gas-station retrieval in the application. Lower MAE and MSE directly decrease the likelihood of feasibility misclassification near the reachability threshold, thereby enhancing the reliability of the filtering step that determines which stations are considered reachable.

3.7. Android Application Output and Recommendation Behavior

Fig. 8 displays two user interface views from the Android application used to predict vehicle travel distance and recommend nearby gas stations based on the user's location and remaining fuel. In the first view, the four numbers at the top of the input screen serve as the foundation for the distance calculation within the app. The first two numbers represent the current user's location coordinates in latitude and longitude format, allowing the system to know the starting point of the vehicle. The next two numbers represent the destination coordinates, also in latitude and longitude format, which are used to calculate the total travel distance. Following these four coordinates, the next number indicates the vehicle's fuel economy in km/l, determining how far the vehicle can travel per liter of fuel. The

final number is the remaining fuel in liters, which is used to calculate the distance that can still be traveled from the vehicle's current state. With this input sequence, the app combines the position, travel target, and fuel consumption condition to predict the travel distance and recommend relevant gas stations. After entering the vehicle's fuel economy and remaining fuel, the user presses the "CALCULATE" button to initiate the calculation process. The second view shows the results after the button is pressed, where the app displays the distance to the destination as 10.71 km, the remaining travel distance based on the available fuel as 6.30 km, and a list of the nearest gas stations that can be reached within this distance. Two BP-branded gas stations are shown, complete with addresses and distances from the user's location: one at Jl. Raya Kertajaya (5.92 km) and another at Jl. Pemuda (5.85 km). The bottom of the screen shows a map of Surabaya city with markers for the recommended gas station locations, providing a visual guide to assist with navigation. A "BACK" button is available to return to the input page, showcasing the app's simple and interactive user flow.

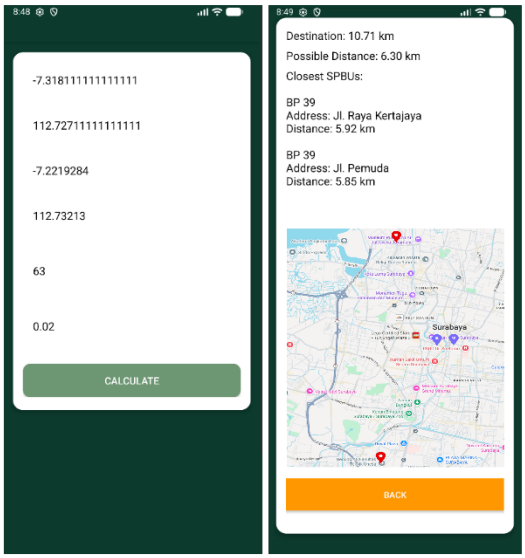

Fig. 8. Fuel finder: Predicted distance and gas station recommendation app

The gas station brands that appear in the output of this application, such as BP, are not manually selected but are a direct consequence of the calculation process using the K-Nearest Neighbors (KNN) algorithm. After the user inputs the current location, destination, and the estimated remaining travel distance based on the available fuel, the system calculates the distance between the user's location and all gas stations in the dataset using the proximity formula (Haversine). KNN then sorts these gas stations based on the closest distance and selects the nearest neighbors according to the specified k value. Since, within the remaining reachable distance of the vehicle, BP-branded gas stations are the closest and most relevant points according to the KNN calculation, BP gas stations appear as recommendations in the application results. In other words, the appearance of this brand reflects the objective result of the KNN algorithm based on geographic proximity patterns, not brand preferences or static determination.

The performance of the model in Table 1 directly affects the results displayed in the application. When the user inputs fuel consumption data and remaining fuel based on the type of motorcycle used, the application uses the model parameters for that specific motorcycle type to calculate the possible distance and determine the nearest gas station that can still be reached. This means that the better the model's performance for a specific motorcycle type, the more accurate the application will be in: (1) calculating the maximum distance that can still be traveled, (2) comparing that distance with the gas stations' positions in the dataset, and (3) displaying realistic gas station recommendations. Consequently, the 110cc motorcycle will produce the most stable predictions, closely matching real-world conditions, making the gas station recommendations in the application more precise for this motorcycle type. Meanwhile, the 125cc and 150cc motorcycles can still be used, but users should be

aware that there is a slight margin of deviation between the application's predicted distance and the actual distance the vehicle can travel.

3.8. Discussion of Model Sensitivity and Future Improvement Directions

The performance of the proposed model has a direct and measurable influence on both user experience and the reliability of the generated recommendations. High predictive accuracy, as reflected by the strong explanatory power and low error metrics, ensures that the estimated reachable distance closely matches real vehicle behavior. For users, this accuracy translates into consistent and predictable system responses, reducing uncertainty during route planning and minimizing cognitive load when interpreting system recommendations. From a recommendation reliability perspective, accurate distance prediction allows the system to rank gas stations based on true reachability rather than approximate proximity. This distinction is critical, as recommendation errors in fuel constrained navigation systems have immediate practical consequences. By maintaining low prediction error, the model reduces the frequency of misleading recommendations, such as suggesting gas stations that appear geographically close but are not reachable given the remaining fuel. Consequently, the system enhances trustworthiness by aligning system outputs with user expectations and real-world outcomes.

Importantly, the observed sensitivity of model performance to the choice of k-value provides a foundation for future methodological improvements. The results indicate that optimal k-values may vary depending on data density, vehicle characteristics, and environmental conditions. Therefore, future studies could explore dynamic k-value adjustment mechanisms, where the number of neighbors is adapted in real time based on local data distribution or prediction uncertainty. Such an approach would allow the model to balance bias and variance more effectively under changing operational contexts. Furthermore, hybrid frameworks that combine KNN with other machine learning techniques, such as decision trees or deep learning models, could leverage the strengths of each method to improve robustness, reduce prediction error, and enhance scalability in complex, real-world navigation scenarios.

4. Conclusion

This study developed a fuel-aware, two-stage framework for gas-station candidate retrieval on Android by combining reachable-distance prediction and geodesic proximity filtering. The KNN model with $k = 5$ produced the most reliable reachable-distance estimates, and its robustness was supported by 5-fold cross-validation. Across the three motorcycle categories, KNN consistently achieved lower prediction errors than reference models, including mean baseline, linear regression, and second-order polynomial regression, indicating that similarity-based learning provides measurable benefits beyond simple parametric assumptions. These findings confirm that KNN-based reachable-distance prediction can serve as a dependable feasibility boundary for filtering gas-station candidates within the application workflow.

This study has several limitations. First, data dependence: the reachable-distance prediction relies on the quality of the fuel–distance observation data and the correctness of input attributes. Inaccurate fuel-consumption values, misclassified vehicle categories, or outdated gas-station coordinates can propagate directly into prediction and feasibility filtering errors. Second, model scope and generalization: the KNN regressor was evaluated only on three motorcycle categories, 110cc, 125cc, and 150cc, so its performance for other vehicles such as cars, trucks, or electric vehicles has not been validated. Third, distance-model assumption: candidate retrieval uses Haversine-based geodesic distance, which represents straight-line distance on the Earth's surface and does not capture road-network constraints such as one-way streets, routing restrictions, congestion, or detours; therefore, feasibility and ranking are based on geometric proximity rather than road distance. Fourth, scalability: the current KNN inference requires comparing a query against stored observations, so computational cost can increase with larger datasets unless efficient neighbor search or indexing is applied.

Future work should proceed in four technically testable directions. First, road-network refinement: after retrieving feasible candidates using geodesic distance, incorporate a graph-based

routing stage to compute road distance and estimated travel time for the shortlisted stations, enabling final ranking under realistic constraints. Second, scalability improvements: replace brute-force KNN search with indexed or approximate nearest-neighbor methods, such as KD-tree or Ball-tree for the feature space, and apply spatial indexing for station retrieval to reduce on-device latency as datasets grow. Third, broader validation: expand the dataset to include additional vehicle categories and driving conditions, and report performance under a controlled train–test split (or cross-validation) to quantify generalization beyond the three motorcycle types. Fourth, uncertainty-aware recommendations: augment the prediction output with confidence or error bounds, then use a conservative feasibility threshold to reduce the risk of recommending stations that may be outside the true reachable range when prediction uncertainty is high.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] P. Rani and R. Sharma, "Intelligent transportation system for internet of vehicles based vehicular networks for smart cities," *Computers and Electrical Engineering*, vol. 105, p. 108543, 2023, https://doi.org/10.1016/j.compeleceng.2022.108543.

[2] S. Zhang, "Challenges in KNN classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 10, pp. 4663-4675, 2022, https://doi.org/10.1109/TKDE.2021.3049250.

[3] O. Kherif, Y. Benmahamed, M. Teguar, A. Boubakeur, and S. S. M. Ghoneim, "Accuracy improvement of power transformer faults diagnostic using KNN classifier with decision tree principle," *IEEE Access*, vol. 9, pp. 81693-81701, 2021, https://doi.org/10.1109/ACCESS.2021.3086135.

[4] H. Lee, H. Choi, K. Sohn, and D. Min, "KNN local attention for image restoration," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 2129-2139, https://doi.org/10.1109/CVPR52688.2022.00218.

[5] A. Pérez-Navarro, R. Montoliu, E. Sansano-Sansano, M. Martínez-Garcia, R. Femenía, and J. Torres-Sospedra, "Accuracy of a single position estimate for kNN-based fingerprinting indoor positioning applying error propagation theory," *IEEE Sensors Journal*, vol. 23, no. 16, pp. 18765-18775, 2023, https://doi.org/10.1109/JSEN.2023.3287856.

[6] L. Xiong and Y. Yao, "Study on an adaptive thermal comfort model with K-nearest-neighbors (KNN) algorithm," *Building and Environment*, vol. 202, p. 108026, 2021, https://doi.org/10.1016/j.buildenv.2021.108026.

[7] J. Jang, "Travel time prediction using machine learning algorithms: Focusing on k-NN, LSTM, and transformer," *The Open Transportation Journal*, vol. 18, pp. 1-13, 2024, https://doi.org/10.2174/0126671212356139241101070347.

[8] Z. Lei, L. Zhu, Y. Fang, X. Li, and B. Liu, "Anomaly detection of bridge health monitoring data based on KNN algorithm," *Journal of Intelligent & Fuzzy Systems*, vol. 39, no. 4, pp. 5243-5252, 2020, https://doi.org/10.3233/JIFS-189009.

[9] A. Ali, M. A. T. Alrubei, L. F. M. Hassan, M. A. M. Al-Ja'afari, and S. H. Abdulwahed, "Diabetes diagnosis based on KNN," *IIUM Engineering Journal*, vol. 21, no. 1, pp. 175-181, 2020, https://doi.org/10.31436/iiumej.v21i1.1206.

[10] A. Juna, M. Umer, S. Sadiq, H. Karamti, A. A. Eshmawi, A. Mohamed, and I. Ashraf, "Water quality prediction using KNN imputer and multilayer perceptron," *Water*, vol. 14, no. 17, p. 2592, 2022, https://doi.org/10.3390/w14172592.

[11] W. Zhuang and Y. Cao, "Short-term traffic flow prediction based on a K-nearest neighbor and bidirectional long short-term memory model," *Applied Sciences*, vol. 13, no. 4, p. 2681, 2023, https://doi.org/10.3390/app13042681.

[12] B. S. Yaswanth, R. S. Darshan, H. Pavan, D. B. Srinivasa, and B. T. V. Murthy, "Smart safety and security solution for women using kNN algorithm and IoT," in *2020 Third International Conference on Multimedia Processing, Communication & Information Technology (MPCIT)*, 2020, pp. 87-92, https://doi.org/10.1109/MPCIT51588.2020.9350431.

[13] R. Raman, V. Kumar, B. G. Pillai, D. Rabadiya, R. Divekar, and H. Vachharajani, "Detecting credit card fraud: A comparative analysis of KNN, random forest, and logistic regression methods," in *2024 Second International Conference on Data Science and Information System (ICDSIS)*, 2024, pp. 1-5, https://doi.org/10.1109/ICDSIS61070.2024.10594698.

[14] N. Vinay and R. Mahaveerakannan, "Analyze the lack of accuracy in stock price prediction using novel K-nearest neighbors regression compared with logistic regression to improve accuracy," in *2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, 2023, pp. 1-5, https://doi.org/10.1109/ICONSTEM56934.2023.10142304.

[15] R. Sujatha and E. Balraj, "A simple framework for predicting student CGPA using multi regression techniques," in *2025 4th International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2025, pp. 1095-1101, https://doi.org/10.1109/ICIMIA67127.2025.11200848.

[16] C. Zhu, "Learning and application of different machine learning methods (KNN, SVM, decision tree) in different datasets," in *2024 5th International Conference on Machine Learning and Computer Application (ICMLCA)*, 2024, pp. 1-5, https://doi.org/10.1109/ICMLCA63499.2024.10754404.

[17] A. Havolli and M. Fetaji, "A comparative analysis of MLR, SVR, and KNN for improving quality of service in next generation network via machine learning regression," in *2024 13th Mediterranean Conference on Embedded Computing (MECO)*, 2024, pp. 1-5, https://doi.org/10.1109/MECO62516.2024.10577892.

[18] F. Y. M. Adiyatma and P. Cherntanomwong, "Radio map augmentation using DBSCAN and KNN regression for improved indoor positioning," in *2024 19th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, 2024, pp. 1-5, https://doi.org/10.1109/iSAI-NLP64410.2024.10799215.

[19] A. B. Rathod, S. M. Gulhane, and S. R. Padalwar, "A comparative study on distance measuring approaches for permutation representations," in *2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT)*, 2017, pp. 251-255, https://doi.org/10.1109/ICAECCT.2016.7942593.

[20] E. Maria, E. Budiman, Haviluddin, and M. Taruk, "Measure distance locating nearest public facilities using Haversine and Euclidean methods," *Journal of Physics: Conference Series*, vol. 1450, no. 1, p. 12080, 2020, https://doi.org/10.1088/1742-6596/1450/1/012080.

[21] S. Zhang and J. Li, "KNN classification with one-step computation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 3, pp. 2711-2723, 2023, https://doi.org/10.1109/TKDE.2021.3119140.

[22] Z. Bian, C. M. Vong, P. K. Wong, and S. Wang, "Fuzzy KNN method with adaptive nearest neighbors," *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 5380-5393, 2022, https://doi.org/10.1109/TCYB.2020.3031610.

[23] N. Marchang and R. Tripathi, "KNN-ST: Exploiting spatio-temporal correlation for missing data inference in environmental crowd sensing," *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3429-3436, 2021, https://doi.org/10.1109/JSEN.2020.3024976.

[24] C. Lin and N. D. Doyog, "Applying a four-way factorial experimental model to diagnose optimum kNN parameters for precise aboveground biomass mapping," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 18, pp. 479-495, 2025, https://doi.org/10.1109/JSTARS.2024.3486737.

[25] A. Andreou, C. X. Mavromoustakis, J. M. Batalla, E. K. Markakis, G. Mastorakis, and S. Mumtaz, "UAV trajectory optimisation in smart cities using modified A* algorithm combined with Haversine and

Vincenty formulas," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 8, pp. 9757-9769, 2023, https://doi.org/10.1109/TVT.2023.3254604.

[26] R. Ward and B. Sencer, "Accurate TCP position and orientation trajectory generation in 6DOF robotic manipulators and CNC machine tools using FIR filtering and Haversine synchronisation," *Procedia CIRP*, vol. 120, pp. 27-32, 2023, https://doi.org/10.1016/j.procir.2023.08.006.

[27] S. Al Farisi, Fauziah, and R. Tamara Aldisa, "A combination of the Haversine formula algorithm and the sequential searching algorithm in web based online attendance," in *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, 2023, pp. 450-455, https://doi.org/10.1109/ICCoSITE57641.2023.10127681.

[28] I. G. S. M. Diyasa, D. A. Prasetya, M. Idhom, A. P. Sari, and A. M. Kassim, "Implementation of Haversine algorithm and geolocation for travel recommendations on smart applications for backpackers in Bali," in *2022 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, 2022, pp. 504-508, https://doi.org/10.1109/ICIMCIS56303.2022.10017760.

[29] Amiruddin, F. R. Baharuddin, Takbir, and W. Setialaksana, "May student-centered principles affect active learning and its counterpart? An empirical study of Indonesian curriculum implementation," *Sage Open*, vol. 13, no. 4, p. 21582440231214375, Oct. 2023, https://doi.org/10.1177/21582440231214375.

[30] S. R. Senthilkumar, V. H. Jabak, G. Vidyashankar, D. Divij, and S. Saravanan, "Optimized bus route finder for educational institutes using Haversine formula," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2024, pp. 1-4, https://doi.org/10.1109/ICCCNT61001.2024.10725509.

[31] I. S. Permana, T. Arlovin, T. Hidayat, I. Sarief, H. H. Solihin, and C. D. Mulyadi, "Optimizing art studio connectivity: A Haversine and greedy algorithm approach for navigation in Cirebon Indonesia," in *2023 17th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, 2023, pp. 1-5, https://doi.org/10.1109/TSSA59948.2023.10366978.

[32] D. Ikasari, Widiastuti, and R. Andika, "Determine the shortest path problem using Haversine algorithm, a case study of SMA zoning in Depok," in *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2021, pp. 1-6, https://doi.org/10.1109/HORA52670.2021.9461185.

[33] I. P. G. A. Sudiatmika, K. H. S. Dewi, and A. A. R. Jayaningsih, "Garage geographic information system using Haversine method based on Android," in *2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS)*, 2021, pp. 1-7, https://doi.org/10.1109/ICORIS52787.2021.9649580.

[34] B. Işik, "Area optimized FPGA implementation of slant range calculation using Haversine formula," in *2021 29th Signal Processing and Communications Applications Conference (SIU)*, 2021, pp. 1-4, https://doi.org/10.1109/SIU53274.2021.9478008.

[35] S. A. Ige, M. S. Khan, and B. R. Biju, "Enhancing EV charging capabilities using Haversine and convex optimization for ITS," in *2023 IEEE Virtual Conference on Communications (VCC)*, 2023, pp. 165-170, https://doi.org/10.1109/VCC60689.2023.10475040.

[36] Y. Chen, Z. Shen, Z. Xu, L. Jin, and W. Chen, "Parameter shift prediction of planar transformer based on Bi-LSTM algorithm," *CPSS Transactions on Power Electronics and Applications*, vol. 8, no. 1, pp. 13-22, 2023, https://doi.org/10.24295/CPSSTPEA.2023.00002.

[37] M. Slawski, E. Ben-David, and P. Li, "Two-stage approach to multivariate linear regression with sparsely mismatched data," *Journal of Machine Learning Research*, vol. 21, no. 204, pp. 1-42, 2020, http://jmlr.org/papers/v21/19-645.html.

[38] Y. Quan, C. Liu, Z. Yuan, and Y. Zhou, "An intelligent multiscale spatiotemporal fusion network model for TCM," *IEEE Sensors Journal*, vol. 23, no. 7, pp. 6628-6637, 2023, https://doi.org/10.1109/JSEN.2023.3244587.

[39] E. Sitompul, R. M. Putra, H. Tarigan, A. Silitonga, and I. Bukhori, "Implementation of digital feedback control with change rate limiter in regulating water flow rate using Arduino," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 6, no. 1, pp. 72-82, 2024, https://doi.org/10.12928/biste.v6i1.10234.

[40] H. D. Trung, "Estimation of crowd density using image processing techniques with background pixel model and visual geometry group," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 6, no. 2, pp. 142-154, 2024, https://doi.org/10.12928/biste.v6i2.10785.

[41] S. Mishra, P. K. Mallick, H. K. Tripathy, L. Jena, and G.-S. Chae, "Stacked KNN with hard voting predictive approach to assist hiring process in IT organizations," *International Journal of Electrical Engineering & Education*, p. 0020720921989015, 2021, https://doi.org/10.1177/0020720921989015.