Design and Implementation of a Machine Learning-Based Network Optimization Recommendation System with Web Performance Evaluation

Ahmad Rifai[1], **Asri Wulandari***[1], **Alfin Hikmaturokhman**[2]
[1]*Department of Electrical Engineering Politeknik Negeri Jakarta*
Jl. Prof. DR. G.A. Siwabessy, Kampus Universitas Indonesia Depok 16425
Depok, Indonesia
[2]*Department of Electrical Engineering Telkom University*
Jl. DI Panjaitan No.128, South Purwokerto
Purwokerto, Indonesia
Email: asri.wulandari@elektro.pnj.ac.id*

ARTICLE INFORMATION

Received 25 July 2025
Revised 23 November 2025
Accepted 01 December 2025

Keywords:
Machine Learning
Network Optimization
Random Forest
Recommendation System Website

ABSTRACT

Network optimization is an essential process to maintain the quality of cellular services. However, manual analysis of drive test data to determine optimization recommendations is time-consuming and inefficient. This study aims to develop a machine learning-based network optimization recommendation system implemented in the form of a website to assist RF Engineers in analyzing drive test data more efficiently. The system uses a Random Forest Classifier to recommend the type of network optimization, achieving an average accuracy, precision, recall, and F1-score of 95.5%, and a Random Forest Regressor to predict network performance parameters after optimization, with an average R^2 of 0.9618, MAE of 0.0178, MSE of 0.00078, and RMSE of 0.0286. The dataset used is obtained from drive tests consisting of longitude, latitude, RSRP, SINR, Downlink Throughput, and Uplink Throughput parameters. The website was developed using the Flask framework and tested using System Usability Scale (SUS), Google Lighthouse, and GTmetrix. SUS testing obtained an average score of 79.16, categorized as "Good," indicating that the website is easy to use and understand. Google Lighthouse testing obtained a performance score of 82, indicating good and responsive loading performance. GTmetrix testing showed an average performance score of 90.5% and structure score of 90.25%, indicating a well-structured website with optimal loading performance across various global server locations. This system can assist RF Engineers in analyzing drive test data and making network optimization decisions more quickly, practically, and efficiently.

1. Introduction

The development of information and communication technology encourages telecommunications operators to improve efficiency in the network optimization process. One of the main challenges faced is the long time required to manually analyze drivetest data, especially when determining the appropriate network optimization recommendations in a short time. This condition can hinder decision-making processes and handling of low-performance sites.

Fourth Generation (4G) or Long Term Evolution (LTE) is a wireless communication standard developed by the Third Generation Partnership Project (3GPP) that focuses on all-IP based packet-switched services. 4G LTE networks provide high data rates with downlink speeds of up to 300 Mbps and uplink speeds of up to 75 Mbps, enabling seamless connectivity for users without service disruption. This technology supports various applications such as voice, data, video, and IPTV services. LTE network infrastructure utilizes fiber optic transmission to ensure stable and fast connectivity (Damayanti et al., 2023).

Machine learning is a potential solution due to its ability to learn patterns from historical data and produce predictions with high accuracy. (Wahid et al., 2022) utilized the Random Forest Regressor algorithm to predict network throughput with high accuracy results. (Didigwu & Anichi., 2024) compared several machine learning algorithms for predicting LTE network performance and concluded that Random Forest Regressor had the best performance among other algorithms.

However, previous studies generally focused only on predicting a single network parameter without integrating optimization recommendation features into an applied system. In fact, the presence of a recommendation system equipped with parameter predictions after optimization will facilitate RF Engineers in conducting analyses and making decisions more quickly and effectively.

This study aims to design and implement a machine learning-based network optimization recommendation system in the form of a website. The system uses a Random Forest Classifier algorithm to determine the type of network optimization suitable for site conditions, as well as a Random Forest Regressor to predict network parameter values after optimization. With this system, it is expected to assist RF Engineers in analyzing drivetest data automatically, efficiently, and support the improvement of cellular network service quality.

2. Literature review

2.1. *Literature Review*

Several previous studies have utilized machine learning algorithms in predicting cellular network performance parameters. (Didigwu & Anichi., 2024) conducted downlink throughput prediction for 4G networks by comparing Random Forest, Linear Regression, Gradient Boosting, Support Vector Regression (SVR), and K-Nearest Neighbors (KNN), and concluded that Random Forest showed the best performance but did not visualize drivetest data results. (Wahid et al., 2022) used Random Forest Regression, Gaussian Process Regression, and KNN to predict downlink throughput in 4G LTE networks, achieving high accuracy but without providing optimization recommendations for practical field implementation.

In addition, (Fauzi et al., 2022) predicted cellular network coverage using RSRP parameters with several supervised learning algorithms such as Linear Regression, Artificial Neural Networks, Support Vector Machine, Regression Trees, Ensembles of Trees, and Gaussian Process Regression. However, the study focused only on RSRP prediction without implementing inference models for direct application. Meanwhile, (Eyceyurt et al., 2022) conducted uplink throughput prediction using Linear Regression, Gradient Descent, Gradient Boosting Regression, Decision Tree Regression, and KNN, but their research was limited to model development without inference implementation or integration into an applied system. A summary of previous related studies can be seen in Table 1.

Table 1. Previous research related to machine learning in cellular networks

Researchers & Year	Title	Method	Research Focus	Strength	Limitation
(Didigwu & Anichi ., 2024)	*Prediction of Mobile Network Performance Using Supervised Machine Learning Models*	*Random Forest*, *Linear Regression*, *Gradient Boosting*, *Support Vector Regression* (SVR), dan *K-Nearest Neighbours* (KNN)	Predicting 4G network parameters (Downlink Throughput) using machine learning algorithms	Comparing several machine learning models in predicting downlink throughput parameters	Only predicts downlink throughput parameters and does not visualize 4G data network parameter data from drive tests
(Fauzi, et al., 2022)	*Mobile Network Coverage*	*Linear Regression* (LR), *Artificial*	Predicting cellular network coverage	Comparing several machine	Only predicts RSRP parameters and does not

Researchers & Year	Title	Method	Research Focus	Strength	Limitation
	Prediction Based on Supervised Machine Learning Algorithms	*Neural Network* (ANN), *Support Vector Machine* (SVM), *Regression Trees* (RT), *Ensembles of Trees* (ET), dan *Gaussian Process Regression* (GPR)	(RSRP) using supervised machine learning algorithms	learning algorithms in predicting cellular network coverage	perform machine learning model inference
(Wahid, et al., 2022)	*Machine Learning Model for Performance Prediction in Mobile Network Management*	*Random Forest Regression*, *Gaussian Process Regression*, dan *K-Nearest Neighbour* (KNN)	Using machine learning to predict 4G LTE network performance (Downlink Throughput)	Comparing several machine learning models in predicting downlink throughput parameters	Only predicts downlink throughput parameters and does not provide any type of network optimization recommendations
(eyceyurt, et al., 2022)	Machine-Learning-Based Uplink Throughput Prediction from Physical Layer Measurements	*Linear Regression*, *Gradient Descent*, *Gradient Boosting Regression*, *Decision Tree Regression*, dan *K-Nearest Neighbour*	Predicting 4G network parameters (Uplink Throughput) using machine learning algorithms	Comparing several machine learning models in predicting uplink throughput parameters	Only creates models to predict uplink throughput parameters and does not perform model inference

Based on Table 1, most previous studies focused only on predicting a single network parameter, such as RSRP or throughput, without providing optimization recommendations or parameter prediction after optimization in an integrated system. Therefore, this study offers novelty by combining a Random Forest Classifier to recommend network optimization types and a Random Forest Regressor to predict performance parameter values after optimization. Both models are implemented in a Flask-based website with interactive visualization to support data analysis and network optimization decision-making more practically, quickly, and efficiently.

2.2. *Machine Learning in Cellular Networks*

Machine learning is a derivative of artificial intelligence (AI). Machine learning is an autonomous machine that can learn independently from data provided by users using an algorithm. Machine learning can make fairly accurate predictions and decisions. Every aspect of machine learning is highly beneficial in project development, starting with data cleaning, learning and understanding various models, and ending with visualization and interpretation of prediction or classification results (Ningrum & Ihsanudin, 2023).

Machine learning has been widely utilized in the field of cellular networks to support optimization processes and network performance management. (Wahid et al., 2022) stated that ML can be used in predictive network analytics to predict network performance in locations without direct measurement data, thus assisting operators in optimizing coverage and network capacity. (Didigwu & Anichi., 2024) also stated that Machine Learning implementation enables network management automation, improved resource allocation, and enhanced service quality for users.

(A. Fauzi et al., 2022) emphasized that Machine Learning-based prediction models outperform traditional methods in computational efficiency and accuracy, especially in predicting RSRP parameters representing cellular network signal coverage. Additionally, (Eyceyurt et al., 2022) mentioned that uplink throughput prediction using ML is crucial considering the increasing uplink data demand due to IoT-based applications and cloud services, which, if not optimized, will cause congestion on the uplink channel.

2.3. *Algoritma Random Forest*

Random Forest is a technique that uses a collection of decision trees as a basic model for classification or regression. Random Forest is one of the ensemble learning methods used to make more accurate and stable predictions. In classification using Random Forest, this method uses a voting approach to make majority decisions based on the results of the trees that have been formed (Sholihah & Hermawan, 2023).

Random Forest is one of the most widely used supervised learning algorithms in cellular network studies due to its ability to handle complex and non-linear data. (Didigwu & Anichi., 2024) showed that Random Forest Regression achieved the highest accuracy in predicting downlink throughput compared to algorithms such as Linear Regression, Gradient Boosting, SVR, and KNN. (Wahid et al., 2022) also obtained similar results, where Random Forest produced an R^2 value of 0.79, higher than KNN (0.66) and Gaussian Process Regression (0.34) in network throughput prediction.

The main advantages of Random Forest include its capability to perform feature importance analysis to identify which parameters most significantly affect network performance and its ability to reduce variance, thus increasing model accuracy. Furthermore, Random Forest is effective for both classification and regression tasks, making it suitable for application in network optimization recommendation systems that require classification of optimization types and prediction of performance parameters after optimization.

2.4. *System Usability Scale (SUS)*

The System Usability Scale (SUS) is a simple yet effective evaluation instrument used to measure how easy a system is to use by its users. The SUS questionnaire consists of 10 statements arranged alternately between positive and negative statements. Users are asked to rate each statement using a Likert scale from 1 (strongly disagree) to 5 (strongly agree) (Dako & Ridwan, 2022).

In the scoring process, for odd-numbered (positive) statements, the score is reduced by 1, while for even-numbered (negative) statements, the score is calculated as 5 minus the user's answer. The values from all statements are then summed and multiplied by 2.5 to obtain the final score ranging from 0 to 100 (Dako & Ridwan, 2022). This final score is then interpreted into usability level categories as shown in Table 2.

Table 2. SUS Score Criteria

Score Range	Category
86 – 100	Best Imaginable
81 – 85	Excellent
71 – 80	Good
51 – 70	OK
26 – 50	Poor
0 – 25	Worst Imaginable

2.5. *Google Lighthouse*

Lighthouse is an open-source automated tool used to test and analyze website performance. Lighthouse provides scores and reports related to loading speed, resource usage, and best web development practices (Google, n.d.). Lighthouse has standardized scoring as shown in Table 3.

Table 3. standardization of Google Lighthouse scores

Score Range	Category
0 – 49	Poor
50 – 89	Fair
90 – 100	Good

2.6. *GTMetrix*

GTmetrix is a tool developed by Gossamer Threads, a company based in the United States, used to analyze website performance to determine the performance level of the website tested. The results of website performance testing using GTmetrix are obtained in the form of grades and scores (Hidayanti, 2022). The scoring system used in GTmetrix is based on the Google Lighthouse scoring system. The grade in GTmetrix is a combination of the Performance Score and Structure Score (GTmetrix, 2020). The grade standardization in GTmetrix is shown in Table 4.

Table 4. GTmetrix Grade Standardization

GTmetrix Grade	GTmetrix Grade Letter Grade	Category
90 – 100	A	Excellent
80 – 89	B	Good
70 – 79	C	Fair
60 – 69	D	Poor
50 – 59	E	Bad
0 – 49	F	Very Bad

3. Method

This research began with the collection of drivetest datasets consisting of location and network performance parameters. The data were processed through preprocessing, feature selection, and normalization stages to ensure data quality before building the machine learning models. The Random Forest Classifier algorithm was used to determine the type of network optimization recommendation, while the Random Forest Regressor was used to predict performance parameter values after optimization. The network optimization recommendation system website was developed using the Flask framework integrated with the machine learning models. Testing was conducted on both the models and the website to evaluate prediction performance and website quality in terms of usability and performance efficiency.

3.1. *Dataset Collection*

The data used in this study were obtained through drivetests, which are direct measurements of network performance at a Telkomsel operator site located in Cikarang using PHU Smart software. This dataset consists of 3,546 measurement data points. The recorded technical parameters include location (longitude and latitude), signal strength (RSRP), signal quality (SINR), as well as downlink and uplink throughput. This drivetest dataset served as the basis for building machine learning models for classification and network performance prediction.

3.2. *Data Preprocessing*

The collected dataset was prepared and cleaned to produce good and accurate machine learning model performance. The preprocessing stages include:

3.2.1. *Feature Selection*

Feature selection is the process of selecting the most relevant features from the dataset to be used in building machine learning models. This process aims to reduce the risk of overfitting, accelerate training time, and improve model accuracy and efficiency (Jovic et al., 2015). In this study, the features used consist of

Longitude, Latitude, RSRP, SINR, Downlink Throughput, and Uplink Throughput, selected based on their relevance to cellular network performance and their potential contribution to prediction and optimization recommendation processes.

3.2.2. *Handling Missing Values*

Missing values in the dataset can affect the accuracy and performance of machine learning models when implemented (Das et al., 2019). One solution to address this issue is imputation, a technique to fill missing values in the dataset, which can be done using various methods such as filling based on mean, median, or machine learning-based approaches (Hasan et al., 2021). In this study, missing value handling was carried out using two approaches: first, deleting rows with empty values in location features (Longitude and Latitude); second, filling missing values in RSRP, SINR, Downlink Throughput, and Uplink Throughput features using the median value of each feature.

3.2.3. *Data Normalization*

Normalization is the process of transforming numerical data into a specific scale so that each feature has balanced contributions during model training, preventing domination by features with large scales which can cause bias in machine learning algorithms (Patro & Sahu, 2015). This study used Min-Max Scaling through MinMaxScaler, which transforms numerical feature values into a range of [0, 1], ensuring all features are on a uniform scale. This approach has been proven to improve model accuracy and performance (Pranolo et al., 2024). In this study, RSRP, SINR, Downlink Throughput, and Uplink Throughput features were normalized using MinMaxScaler, while Longitude and Latitude were not normalized as they are in consistent geographic degree scales and do not require additional transformation.

3.2.4. *Clustering*

Clustering is a data mining technique aimed at grouping data into several clusters based on the similarity of their characteristics. The goal is to ensure data within a cluster have high similarity, while data between clusters have significant differences (Rachmatullah, 2022).

The K-Means algorithm works by dividing data into k groups based on the distance between each data point and the cluster center (centroid) (Mulyo & Heikal, 2022). In this study, the optimal number of clusters was determined using the Elbow method, which identifies the best k value by observing the inflection point in the graph between the number of clusters and within-cluster sum of squares (WCSS) (Hidayat et al., 2025).

All available features in the dataset, namely Longitude, Latitude, RSRP, SINR, Downlink Throughput, and Uplink Throughput, were used in the clustering process to ensure clusters formed reflect variations in network performance characteristics comprehensively.

3.2.5. *Encoding*

Encoding is the process of converting categorical data into numerical representations so that they can be processed by machine learning algorithms. In this study, One-Hot Encoding was used, which represents each category as a binary vector where only one element has a value of “1” and the rest are “0” (Almajid & Arifudin, 2021).

In this study, the categorical feature generated from the classification model in the form of network optimization recommendation types was encoded using One-Hot Encoding to prepare it as input for the next stage, namely building the model to predict network parameter values after optimization.

3.3. *Machine Learning Model Training and Testing*

After preprocessing the data, the next stage was training and testing the machine learning models. Two models were used in this study, namely Random Forest Classifier and Random Forest Regressor. Both models were trained using drivetest datasets that had undergone preprocessing to ensure optimal data quality before training.

3.3.1. *Random Forest Classifier*

In the classification stage, Random Forest Classifier was used to build a model that provides recommendations for the type of network optimization. This model was trained using features from preprocessing, namely Longitude, Latitude, RSRP, SINR, Downlink Throughput, and Uplink Throughput.

3.3.2. *Random Forest Regression*

To predict network parameter values after optimization, Random Forest Regressor was used. This model was trained with the same features as the classifier model, with the addition of encoded results from network optimization recommendations as input. The target parameters predicted include RSRP, SINR, Downlink Throughput, and Uplink Throughput after optimization.

3.4. *Model Evaluation*

Model evaluation was conducted to measure the performance of the built Random Forest models for both classification in providing network optimization recommendations and regression in predicting network performance parameters after optimization. Various evaluation metrics were used to assess model performance, including Accuracy, Precision, Recall, and F1 Score for classification. Accuracy provides an overview of the percentage of correct predictions, while Precision is important to ensure that every positive prediction made by the model is truly relevant to the correct category (Geng, 2024). Additionally, F1 Score is calculated to assess the balance between precision and recall, which is crucial when both metrics need to be balanced.

For regression model evaluation, metrics used include R^2 Score, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). R^2 Score measures how well the model explains the variability of the target data, with high R^2 indicating the model explains most of the data variation. MAE provides an overview of the model’s average absolute prediction error, while MSE and RMSE are more sensitive to large errors, with RMSE providing a clearer interpretation of prediction error magnitude in the same scale as the target value.

In addition to testing using the main dataset, model evaluation was also carried out on 10 different drivetest datasets to measure model performance stability on site data with varying characteristics. This testing aims to ensure that the model has good generalization capability and can be used on other sites with different data characteristics.

3.5. *Development of Network Optimization Recommendation System Website*

The network optimization recommendation system website was developed to facilitate users in analyzing drivetest data and obtaining network optimization recommendations automatically. The website was built using the Flask framework based on the Python programming language. The trained machine learning models were saved as pickle (.pkl) files and integrated into the website to perform real-time inference.

The website interface was designed using HTML, CSS, and JavaScript to create a responsive and user-friendly display. Additionally, the website is equipped with interactive visualization features using Plotly and Folium libraries, enabling users to view the distribution of network performance parameter values on maps based on longitude and latitude coordinates. This website has several main pages, namely Home, Upload, Result, Help, About, and network performance parameter visualization pages.

3.6. *Website Evaluation*

Website testing was conducted to evaluate the quality and performance of the developed network optimization recommendation system website. Testing was carried out on two main aspects: usability and performance efficiency, using standardized methods and tools.

3.6.1. *Usability Testing*

Usability testing was carried out using the System Usability Scale (SUS) method. The SUS questionnaire was distributed to 9 respondents who work as RF Engineers to assess the ease of use, learnability, and user satisfaction with the website. The test results were processed to obtain the total SUS score and determine the website's feasibility level based on SUS categories.

3.6.2. *Performance Efficiency Testing*

Performance efficiency testing was conducted using two tools: Google Lighthouse and GTmetrix.

a. Google Lighthouse
Testing using Google Lighthouse was performed on each website page to evaluate four main aspects: Performance, Accessibility, Best Practices, and SEO. The scores obtained from each aspect were used to assess the website's overall technical quality.
b. GTMetrix
Testing using GTmetrix was conducted based on global server locations (Vancouver, London, Hong Kong, and Sydney) to measure website performance when accessed from various geographic regions. The evaluated parameters include Performance Score, Grade, Largest Contentful Paint (LCP), Total Blocking Time (TBT), and Cumulative Layout Shift (CLS). These values ensure that the website has good and stable loading performance across various server locations.

4. Results and Discussion

This section explains the results of building the machine learning models and developing the network optimization recommendation system website. In addition, it presents the results of model evaluation and website testing to assess performance and efficiency.

4.1. *Results of Machine Learning Model and Network Optimization Recommendation System Website Development.*

This subsection presents the results of implementing machine learning models, including Random Forest Classifier and Random Forest Regressor, as well as the results of developing a website as a medium for integrating the network optimization recommendation system.

4.1.1. *Random Forest Classifier Results*

The classification model development began with collecting a clean dataset obtained from drivetest results. The collected data were processed to ensure quality and readiness before being used for model training. The next stage was training and testing the model using the Random Forest Classifier algorithm to build a model capable of providing network optimization recommendations. After training, the model's performance was evaluated using evaluation metrics, including accuracy, precision, recall, and F1-score. If the evaluation results showed satisfactory performance, the trained model was used; otherwise, the process returned to the data pre-processing stage for further improvements. Model performance evaluation was conducted comprehensively using evaluation metrics and a confusion matrix to obtain a more in-depth analysis.

Table 5 shows the performance evaluation results of the classification model in recommending network optimization types based on drivetest data, evaluated using four main metrics: accuracy, precision, recall, and F1-score, calculated automatically using Python's scikit-learn library.

Table 5. Classification Model Evaluation Metrics

Evaluation Metric	Score
Accuracy	0.96
Precision	0.96
Recall	0.96
F1 Score	0.96

An accuracy of 0.96 indicates that the model correctly classified 96% of test data. However, for multi-class classification, evaluation using accuracy alone is insufficient. Therefore, other metrics such as precision, recall, and F1-score were also used. Precision of 0.96 reflects the model's ability to produce correct predictions, while recall of 0.96 shows the model's capability to detect relevant data. The F1-score, a harmonic mean between precision and recall, is also 0.96, indicating a good balance between the two.

The consistent high values across all four metrics indicate that the model has excellent and stable classification performance, both in overall accuracy and in recognizing specific patterns in the data. This makes the model highly potential for application in network optimization recommendation systems based on field data.

Figure 1. Confusion Matrix

A confusion matrix is a common evaluation method in classification systems to measure model performance in distinguishing target classes. The matrix compares the model's predicted labels with actual labels from test data, providing detailed insight into the types of classification errors (S. Sathyanarayanan & Tantri, 2024). Figure 1 shows the confusion matrix for the classification model.

The confusion matrix visualization in Figure 1 shows that the classification model performs excellently in recognizing the four network optimization categories. The "Melakukan Antenna Electrical Tilt dan Power Control" category was classified with an accuracy of 93.9%, with a small misclassification to "Melakukan Load Balancing dan Carrier Aggregation" (2%). The "Melakukan Load Balancing dan Carrier Aggregation" category had a high accuracy of 97.4%, with minor misclassifications into two other classes. "Melakukan Physical Tuning" was correctly classified 97.3% of the time, with minor errors into one other class. The "Tidak diperlukan optimasi" category showed a classification accuracy of 95.8%, with errors distributed into two other classes.

Overall, the classification model distinguishes the four types of network optimization actions with very high accuracy, with all classes having correct prediction rates above 90%. This indicates that the model has good generalization capabilities and can be effectively used to detect optimization types in network systems..

4.1.2. *Random Forest Regression Results*

The regression model development began with collecting a clean drivetest dataset, processed to ensure data quality and readiness for model training. Optimization recommendation results obtained using Random Forest Classifier were also used as input alongside drivetest data for model training. The model was then trained using Random Forest Regressor to predict network parameter values after optimization. The model was evaluated using metrics such as MAE, MSE, RMSE, and R^2. If evaluation results showed satisfactory performance, the model was ready for use; otherwise, the data were returned to the pre-processing stage for further improvements. Model evaluation was conducted using evaluation metrics and prediction vs actual plots for detailed analysis. The evaluation results are shown in Table 6, which illustrates the accuracy level of the model in predicting network parameter values after the optimization process.

Table 6. Regression Model Evaluation Metrics

Evaluation Metric	Score
R^2 Score	0.9569
Mean Absolute Error (MAE)	0.0189
Mean Squared Error (MSE)	0.0012
Root Mean Squared Error (RMSE)	0.0344

The model produced an R^2 of 0.9569, indicating it could explain approximately 95% of target data variability based on input. This high R^2 reflects strong model performance in capturing variable relationships. MAE of 0.0189 indicates that the average absolute error between predictions and actual values is very small, meaning the model provides consistent estimations close to actual values. MSE of 0.0012 shows that the average squared prediction error is very low, minimizing the possibility of large errors. RMSE of 0.0344, as the root of MSE, shows how far the model's predictions deviate from actual values on average, in the same scale as the target (normalized data). The low RMSE demonstrates that the model's predictions are very close to actual values.

Overall, these evaluation metrics show that the regression model is highly reliable for predicting network parameter values after optimization, supporting data-based decision-making in cellular network optimization planning and evaluation.

The visualization of the actual values and the predicted values generated by the regression model is shown in Figure 7. This visualization aims to evaluate how well the model can accurately map the relationship between input and output data. In the graph, the horizontal axis (x) represents the actual values of the network parameters after optimization, while the vertical axis (y) shows the predicted values generated by the model. The red line displayed on the graph is the ideal reference line (y = x) that represents perfect prediction conditions, i.e., when all predicted values are identical to the actual values. The blue dots indicate the distribution of the model's predictions relative to the actual values for each test data point.

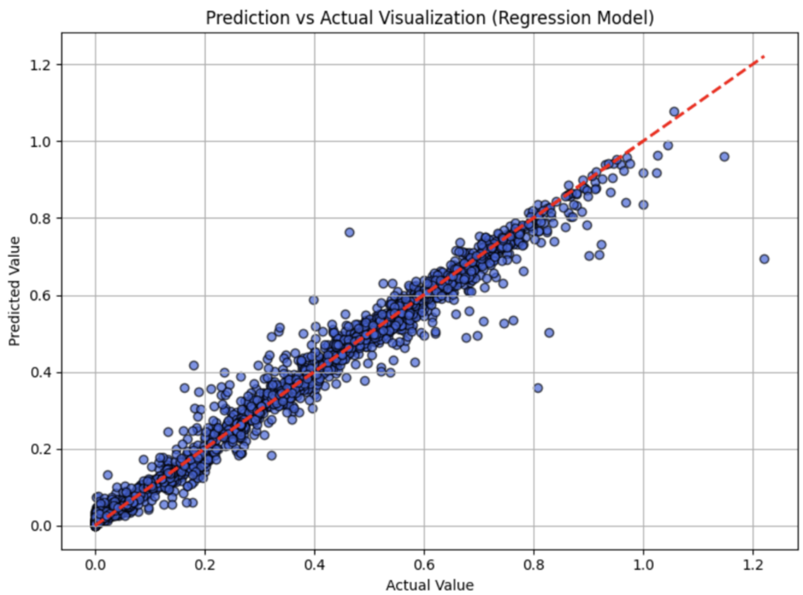

Figure 2. Predicted vs Actual Visualization

In Figure 2, most points are very close to the reference line, showing that the regression model provides predictions very close to actual values. The tight clustering around the diagonal indicates high accuracy and precision in modeling input-output relationships. This consistency demonstrates good performance in capturing data patterns and generalizing to test data, proving the model's effectiveness not only in classification but also in predicting parameter changes after optimization in field-based recommendation systems. This visualization provides additional supporting evidence for the model's performance in a real-world application context on a field data-based recommendation system (drivetest).

4.1.3. *Website Development Results*

The network optimization recommendation system website was developed using the Flask framework. The website has several main pages: Home, Upload Drivetest Dataset, 4G Parameters, Help, and About. Each page was designed with a simple and user-friendly interface to support RF Engineers in analyzing drivetest data efficiently. Figures 3–10 show the page interfaces.

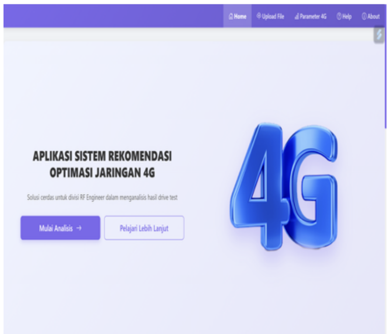

Figure 3. Home Page

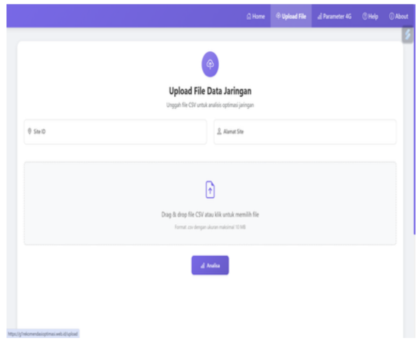

Figure 4. Upload Dataset Page

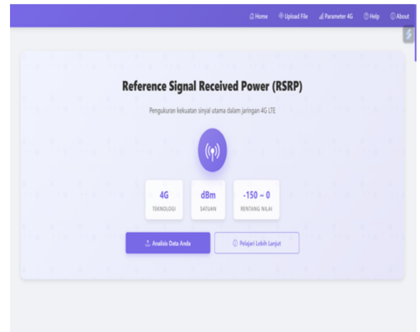

Figure 5. RSRP Parameter Page

Figure 6. SINR Parameter Page

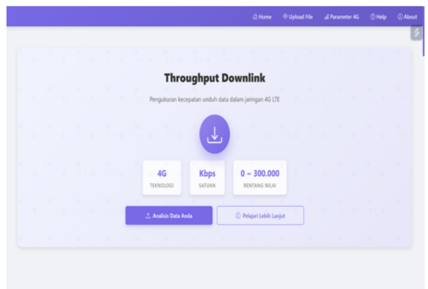

Figure 7. Throughput Downlink Parameter Page

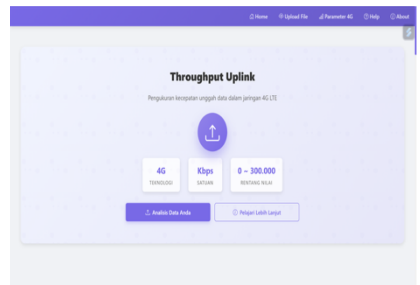

Figure 8. Throughput Uplink Parameter Page

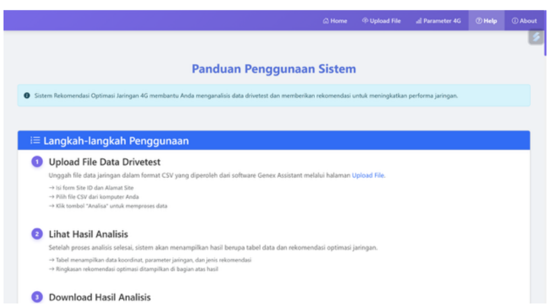

Figure 9. Page Help

Figure 10. Page About

4.2. *Machine Learning Model and Website Testing*

This section describes the results of testing the machine learning model and the network optimization recommendation system website that has been developed. The testing was conducted to determine the performance of the model on different drivetest data and to evaluate the website in terms of usability and performance efficiency.

4.2.1. *Machine Learning Model Testing*

a. Random Forest Classifier Model Testing

The Random Forest Classifier model was tested on 10 different drivetest datasets to evaluate the consistency of the model's classification performance. The test results are presented in a table containing the accuracy, precision, recall, and F1-score values for each dataset. The results of testing the random forest classifier model on 10 different drivetest datasets are shown in Table 7.

Table 7. Classification Model Evaluation Metrics Testing Data

Testing	Accuracy Score	Precision Score	Recall Score	F1-Score
Data Drivetest Site A	0.95	0.95	0.95	0.95
Data Drivetest Site B	0.96	0.96	0.96	0.96
Data Drivetest Site C	0.94	0.95	0.94	0.94
Data Drivetest Site D	0.95	0.95	0.95	0.95
Data Drivetest Site E	0.95	0.95	0.95	0.95
Data Drivetest Site F	0.93	0.94	0.93	0.93
Data Drivetest Site G	0.96	0.96	0.96	0.96
Data Drivetest Site H	0.93	0.93	0.93	0.93
Data Drivetest Site I	0.94	0.94	0.94	0.94
Data Drivetest Site J	0.98	0.98	0.98	0.98
Mean	0.949	0.951	0.949	0.949

The evaluation results of the classification model built using the Random Forest Classifier algorithm show high and consistent evaluation metric values. The average accuracy value reached 0.949, precision was 0.951, recall was 0.949, and the F1-score was 0.949. The consistency of these values indicates that the model has good generalization capabilities and does not suffer from overfitting. Therefore, this model can be relied upon to accurately identify the type of network optimization recommendations under various data conditions.

b. Random Forest Regression Model Testing

The Random Forest Regressor model was tested on 10 different drivetest datasets to evaluate the accuracy of the model in predicting network parameter values after optimization. The test results table shows the R^2, MAE, MSE, and RMSE values for each dataset. The results of testing the random forest classifier model on 10 different drivetest datasets are shown in Table 8.

Table 8. Regression Model Evaluation Metrics Testing Data

Testing	R^2 Score	Mean Absolute Error (MAE)	Mean Squared Error (MSE)	Root Mean Squared Error (RMSE)
Data *Drivetest* Site A	0.9788	0.0117	0.0004	0.0194
Data *Drivetest* Site B	0.9765	0.0150	0.0004	0.0206
Data *Drivetest* Site C	0.9708	0.0122	0.0005	0.0214
Data *Drivetest* Site D	0.9765	0.0120	0.0004	0.0199
Data *Drivetest* Site E	0.9714	0.0133	0.0005	0.0214
Data *Drivetest* Site F	0.9668	0.0166	0.0008	0.0288
Data *Drivetest* Site G	0.9854	0.0110	0.0003	0.0165
Data *Drivetest* Site H	0.9586	0.0151	0.0005	0.0229
Data *Drivetest* Site I	0.9767	0.0126	0.0004	0.0200
Data *Drivetest* Site J	0.9553	0.0101	0.0003	0.0173
Mean	0.9716	0.01296	0.00045	0.02082

The results of the regression model evaluation also show excellent performance. The average R^2 score of 0.9716 indicates that the model is able to explain approximately 97.16% of the variation in the target data. In addition, the average MAE value of 0.01296, MSE of 0.00045, and RMSE of 0.02082 indicate a very low prediction error rate. This shows that the regression model has good generalization capabilities and does not experience overfitting, making it reliable for predicting the performance parameters of the network after optimization.

4.2.2. *Website Testing*

a. Usability Testing

Usability testing was conducted to assess the extent to which a software or system is easy to use, efficient, and provides satisfaction to users. Usability testing was conducted using the System Usability Scale (SUS) questionnaire, which was completed by nine respondents from the RF Engineer division to assess the ease of use of the website. The test results are presented in a table containing the SUS scores of each respondent and the average usability score of the website. Table 9 shows the SUS scores of all respondents.

Table 9. Usability Test Score Results in the RF Engineer Division

Respondents	Question										Score	SUS Value
	1	2	3	4	5	6	7	8	9	10	Total	(Score Total x 2.5)
1	3	3	3	3	3	3	3	2	3	3	29	72.5
2	4	2	4	4	4	4	4	4	4	2	36	90
3	4	3	3	3	4	3	4	3	4	3	33	82.5
4	3	3	4	3	3	3	3	3	4	3	32	80
5	3	3	4	3	3	3	3	3	3	3	31	77.5
6	4	3	4	3	3	3	4	3	3	3	33	82.5
7	4	3	3	3	3	3	3	3	3	3	31	77.5
8	3	3	3	3	3	3	3	2	3	3	29	72.5
9	3	3	3	3	4	3	3	3	3	3	31	77.5
											Results	79.16

Based on the results of the System Usability Scale (SUS) questionnaire completed by 9 respondents from the RF Engineer division, an average score of 79.16 was obtained. This score falls into the "Good" category according to the SUS interpretation standards listed in Table 2. These results indicate that the network optimization recommendation system website generally has good usability, is easy to understand, and can be used effectively by the RF Engineer division. Overall, the test results show that this system is suitable for use and capable of providing a good user experience.

b. Performance Efficiency Testing

Performance efficiency testing is conducted to measure how efficiently a system or software uses resources to provide fast response times. Performance efficiency testing is conducted using Google Lighthouse and GTmetrix, automated audit tools that analyze web pages and provide detailed reports on website performance.

1) Google Lighthouse

Testing using Google Lighthouse was conducted on each website page to evaluate aspects of performance, accessibility, best practices, and SEO. The test results are displayed in a table containing the scores for each aspect on each website page. Table 10 shows the results of performance efficiency testing using Google Lighthouse on each website page.

Table 10. Website Performance Efficiency Test Results Network Optimization Recommendation System

No.	Page	Performance Score
1	Home Page	98
2	Upload Page	99
3	RSRP Page	96

No.	Page	Performance Score
4	SINR Page	97
5	Throughput Downlink Page	99
6	Throughput Uplink Page	99
7	Help Page	96
8	About Page	92
9	Result Page	71
10	RSRP_Result Page	60
11	SINR_Result Page	52
12	Throughput Downlink_Result Page	54
13	Throughput Uplink_Result Page	53
	Mean	82

Based on the test results in Table 10 on performance efficiency using the Lighthouse extension, the average performance score for all pages of the website was 82. Referring to the Lighthouse standards in Table 3, the network optimization recommendation system website falls into the Moderate or fairly good category for Performance Efficiency. This means that the website is suitable for use and provides a fairly good user experience.

2) GTMetrix

Testing using GTmetrix was conducted from several global server locations to evaluate the overall performance efficiency of the website. The GTmetrix results table displays the Performance Score, Grade, LCP, TBT, and CLS values for each server location. Table 11 shows the results of performance efficiency testing using GTMetrix at each server location.

Table 11. Performance Efficiency Test Results using GTmetrix

Gtmetrix Server Location	Grade	Performance	Structure
Vancouver, Canada	B	86%	92%
London, UK	B	82%	87%
Hongkong, China	A	99%	92%
Sydney, Australia	A	95%	90%
	Mean	90.5%	90.25%

Based on the test results shown in Table 11, performance efficiency testing was conducted using GTmetrix on four different server locations, namely Vancouver (Canada), London (UK), Hong Kong (China), and Sydney (Australia). The performance scores obtained ranged from 82% to 99%, with an average of 90.5%. Meanwhile, the structure scores ranged from 87% to 92%, with an average of 90.25%. Based on the GTmetrix Grade Standardization outlined in Table 4, the website for the network optimization recommendation system falls into the “Excellent” category and is suitable for use.

5. Conclusion

This study successfully developed a machine learning-based 4G network optimization recommendation system implemented in the form of a website. The Random Forest Classifier model was used to provide recommendations on the type of network optimization based on drivetest data, while the Random Forest Regressor was used to predict network performance parameter values after optimization. Evaluation results showed that both models performed well with high evaluation scores on both the main dataset and 10 different drivetest datasets, indicating the models’ ability to generalize to site data with varying characteristics.

The developed recommendation system website was also tested in terms of usability and performance efficiency, with the System Usability Scale (SUS) test results showing an average score categorized as “Good.” In addition, testing using Google Lighthouse and GTmetrix showed that the website had good and stable performance when accessed from various global server locations. Therefore, this website can assist RF engineers in conducting drivetest data analysis and determining network optimization recommendations more quickly, practically, and efficiently.

References

A. Fauzi, M. F., Nordin, R., Abdullah, N. F., & Alobaidy, H. A. (2022). Mobile Network Coverage Prediction Based on Supervised Machine Learning Algorithms. *IEEE Access*.

Almajid, A. S., & Arifudin, R. (2021). Multilayer Perceptron Optimization on Imbalanced Data Using SVM-SMOTE and One-Hot Encoding for Credit Card Default Prediction. *Journal of Advances in Information Systems and Technology*, 67-74.

Dako, R. D., & Ridwan, W. (2022). Pengukuran Usability terhadap Aplikasi Tesadaptif.Net dengan System Usability Scale. *Jambura Journal of Electrical and Electronics Engineering* , 207-212.

Damayanti, L., Damelia, Regina, S., Wulandari, A., Hikmaturokhman, A., & Hidayatullah, A. (2023). Design and Build4G OpenRadio Access Network at SmartLab Politeknik Negeri Jakarta. *JITE (Journal of Informatics and Telecommunication Engineering)*, 414-423.

Das, D., Nayak, M., & Pani, S. K. (2019). Missing Value Imputation-A Review. *International Journal of Computer Sciences and Engineering*, 548-558.

Didigwu, F. U., & Anichi, J. (2024). Prediction of Mobile Network Performance Using Supervised Machine Learning Models. *International Journal of Advanced Research in Science and Technology*, 1533-1546.

Eyceyurt, E., Egi, Y., & Zec, J. (2022). Machine-Learning-Based Uplink Throughput Prediction from Physical Layer Measurements. *Electronics*.

Geng, S. (2024). Analysis of the Different Statistical Metrics in Machine Learning. *Highlights in Science, Engineering and Technology*, 350-356.

Google. (t.thn.). *Lighthouse*. Diambil kembali dari Chrome for developers: https://developer.chrome.com/docs/lighthouse

GTmetrix. (2020, 11 16). *Everything You Need to Know About GTmetrix Reports*. Diambil kembali dari GTmetrix Website: https://gtmetrix.com/blog/everything-you-need-to-know-about-the-new-gtmetrix-report-powered-by-lighthouse/#performance-tab

Hasan, M. K., Alam, M. A., Roy, S., Dutta, A., Jawad, M. T., & Sunanda Das. (2021). Missing value imputation affects the performance of machine learning: A review and analysis of the literature (2010–2021). *Informatics in Medicine Unlocked*, 1-23.

Hidayanti, N. H. (2022). Analisis Performa Website Kantor Pencarian dan Pertolongan Palembang Menggunakan GTmetrix. *Seminar Nasional Teknologi Dan Multidisiplin Ilmu SEMNASTEKMU.*

Hidayat, A. L., Zulqan, H., & Triyono, G. (2025). Klasterisasi Provinsi di Indonesia Berdasarkan Angka Harapan Hidup Menggunakan K-Means dengan Evaluasi Elbow Method. *JSiI | Jurnal Sistem Informasi*, 81-85.

Jovic, A., Brkic, K., & Bogunovic, N. (2015). A review of feature selection methods with applications. *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1200-1205.

Mulyo, I. A., & Heikal, J. (2022). Customer Clustering Using The K-Means Clustering Algorithm in Shopping Mall in Indonesia. *Management Analysis Journal*, 365-371.

Ningrum, A. A., & Ihsanudin. (2023). Penerapan Framework Flask Pada Machine Learning Dalam Memprediksi Umur Transformer. *KONVERGENSI*, 51-59.

Patro, S. G., & Sahu, K. K. (2015). Normalization: A Preprocessing Stage. *International Advanced Research Journal in Science, Engineering and Technology*, 20-22.

Pranolo, A., Setyaputri, F. U., Paramarta, A. K., Triono, A. P., Fadhilla, A. F., Akbari, A. K., . . . Uriu, W. (2024). Enhanced Multivariate Time Series Analysis Using LSTM: A Comparative Study of Min-Max and Z-Score Normalization Techniques. *ILKOM Jurnal Ilmiah*, 210-220.

Rachmatullah, M. I. (2022). Proposed Modification of K-Means Clustering Algorithm with Distance Calculation Based on Correlation. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, 136-143.

S. Sathyanarayanan, & Tantri, B. R. (2024). Confusion Matrix-Based Performance Evaluation Metrics. *African Journal of Biomedical Research*, 4023-4031.

Sholihah, N. N., & Hermawan, A. (2023). Implementation Of Random Forest And SMOTE Methods For Economic Status Classification In Cirebon City. *Jurnal Teknik Informatika (JUTIF)* , 1387-1397.

Wahid, H., Abdul Razak, N. I., & Che Abdullah, S. A. (2022). Machine Learning Model for Performance Prediction in Mobile Network Management. *Journal Of Electrical And Electronic Systems Research*, 101-107.