International Journal of Web of Multidisciplinary Studies
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
website: *https://ijwos.com* OPEN ACCESS E-ISSN : 3049-2424
Volume 03, Issue 10,October 2026 DOI: https://doi.org/10.71366/ijwos03102607088

NETEYEAI: AI-Based Network Availability Prediction and Alert System

Prof. P L HALGEKAR[1], GANESH MANOJ HABIB[2], SUPRIM ROY[3], RIHAN YALIGAR[4], VINAYAK MALI[5]

[1,2,3,4,5] *Department of Artificial Intelligence and Data Science, Angadi Institute of Technology and Management, Belagavi, India*

Article Info

Article History:
Published: 31 Oct 2026

Publication Issue:
Volume 03, Issue 10
October 2026

Page Number:
620-629

Corresponding Author:
Prof. P L HALGEKAR

Abstract:

NetEyeAI: AI-Based Network Availability Prediction and Alert System is an intelligent mobile-based application designed to help users understand mobile network availability along their travel routes. Mobile network connectivity can vary across geographical locations due to factors such as distance from network infrastructure, physical obstacles, and environmental conditions. Users often experience weak signals or network loss while travelling without having information about connectivity conditions ahead. To address this challenge, NetEyeAI uses GPS-based location information and historical network measurements to estimate network availability approximately 2–5 km ahead. The system utilizes Machine Learning, specifically the LightGBM algorithm, to analyse network-related features and classify predicted network conditions into four categories: poor or no network, fair, good, and excellent. This helps users identify potential weak-network areas before reaching those locations.

An interactive map interface displays route information and predicted network conditions, helping users understand connectivity variations along their journey. The system integrates a Flutter-based mobile application, a FastAPI backend, and a trained LightGBM machine learning model to process prediction requests and present the results. By combining machine learning, GPS-based route analysis, and geospatial visualization, NetEyeAI aims to improve network awareness, provide advance information about potential connectivity issues, and support better travel planning in areas with varying mobile network availability.

Keywords: NetEyeAI, Network Availability Prediction, LightGBM, Machine Learning, GPS-Based Prediction, Mobile Network Monitoring, Geospatial Analysis, Network Quality Classification, Route-Based Prediction.

1. INTRODUCTION

Mobile network connectivity plays an important role in modern communication and everyday activities. People depend on mobile networks for making calls, accessing the internet, using navigation applications, online transactions, and various digital services. Reliable network connectivity is particularly important while travelling, as users may need continuous access to communication and online information throughout their journey. However, mobile network availability can vary significantly from one geographical location to another due to factors such as distance from network infrastructure, geographical conditions, physical obstacles, and network coverage limitations. When users encounter weak signals or complete network loss, it can interrupt communication and affect access to important online services.

One of the major challenges faced by mobile network users is the inability to know network conditions at upcoming locations. Generally, users can check their current signal strength through their mobile devices, but this information does not necessarily indicate the network availability farther along their route. Network conditions may change as users move between different geographical areas. Without advance information, users may unexpectedly enter areas with weak or unavailable connectivity. A system that analyses historical network measurements and geographical

Copyright © 2026 The Author(s) : This is an open access article under the CC BY license (http://creativecommons.org/licenses/by-nc/4.0/)

information can help estimate future network conditions and provide users with better awareness of connectivity along their journey.

Another important aspect of network monitoring is understanding how network quality changes across different locations. Historical signal measurements can provide useful information about previously observed connectivity conditions. By combining these measurements with GPS-based location information, it becomes possible to analyse geographical variations in network availability. Machine Learning techniques can further help identify patterns in the collected data and estimate network conditions at upcoming locations. Such predictions can be useful for identifying potential weak-network areas before users reach them.

The proposed NetEyeAI: AI-Based Network Availability Prediction and Alert System is a mobile-based application designed to predict network availability along a user's route. The system uses GPS-based location information and historical network measurements to estimate network conditions approximately 2–5 km ahead. It employs the LightGBM machine learning algorithm to classify predicted network quality into four categories: poor or no network, fair, good, and excellent. The application aims to provide users with advance information about potential connectivity changes instead of displaying only the current network condition.

The proposed system integrates a Flutter-based mobile application, a FastAPI backend, a trained LightGBM machine learning model, and an interactive map interface. The mobile application handles location information, route visualization, and the display of prediction results, while the backend processes prediction requests and communicates with the trained machine learning model. By combining machine learning, GPS-based route analysis, historical network data, and geospatial visualization, NetEyeAI aims to improve network awareness, help users identify potential connectivity limitations in advance, and support better planning during travel.

2. RELATED WORK

2.1 Mobile Network Coverage Prediction

Predicting mobile network coverage has been an important area of research in wireless communication. Network coverage can vary across geographical locations depending on factors such as distance from mobile towers, physical obstacles, terrain conditions, and network infrastructure. Researchers have explored different techniques to estimate network availability using historical signal measurements and geographical information. These methods help identify areas with strong, weak, or unavailable network connectivity. However, traditional coverage estimation methods may not always provide users with information about network conditions at upcoming locations along their travel routes.

2.2 Machine Learning-Based Network Availability Prediction

Machine Learning is increasingly being used to analyse mobile network performance and predict connectivity conditions. Algorithms such as Decision Tree, Random Forest, Support Vector Machine, Logistic Regression, and Gradient Boosting can analyse historical network measurements and identify patterns associated with different network conditions. These techniques can help estimate network quality using features such as signal strength, geographical coordinates, and other relevant network parameters. In the proposed NetEyeAI system, we use the LightGBM algorithm because it is an efficient gradient boosting framework suitable for structured datasets. The trained model analyses network-related features and classifies predicted network conditions into four categories: poor or no network, fair, good, and excellent.

2.3 Received Signal Strength Prediction

Received Signal Strength Indicator (RSSI) is an important measurement used to understand the strength of a wireless network signal. Researchers have explored different machine learning techniques to predict RSSI by analysing historical signal measurements and relevant input parameters. These prediction methods can help estimate signal strength at locations where direct measurements may not be available. By studying the relationship between geographical locations and previously recorded signal values, it becomes possible to understand how network quality changes across different areas. NetEyeAI uses historical network measurements and location-related information to estimate network conditions at upcoming geographical locations.

2.4 GPS-Based and Geospatial Network Monitoring Systems

GPS and geospatial technologies play an important role in location-based network monitoring. By associating network measurements with geographical coordinates such as latitude and longitude, connectivity information can be analysed

and represented across different locations. These technologies help identify geographical variations in network availability and visualize network conditions on digital maps. Combining GPS information with historical network measurements can provide useful insights into network quality along a user's route. The proposed NetEyeAI system uses GPS-based route information and geospatial analysis to estimate network availability approximately 2–5 km ahead and display the predicted conditions through an interactive map interface.

2.5 Mobile Network Monitoring Applications

Existing mobile network monitoring applications generally focus on displaying current signal strength, network type, internet connectivity, and coverage information. Some applications also provide network coverage maps that help users understand signal availability in different geographical areas. These tools are useful for checking present network conditions and examining previously recorded coverage information. However, applications that mainly focus on current signal measurements may provide limited information about connectivity conditions farther along a user's journey. NetEyeAI aims to address this limitation by combining GPS-based location information, historical network measurements, and machine learning to predict upcoming network conditions and provide users with advance awareness of potential weak-network areas.

3. SYSTEM ARCHITECTURE AND METHODOLOGY

The NetEyeAI: AI-Based Network Availability Prediction and Alert System is designed using a modular architecture that connects the mobile application, backend services, and machine learning model. The Flutter-based mobile application allows users to access location information, view routes, and check predicted network conditions. The FastAPI backend handles prediction requests and communicates with the trained LightGBM machine learning model. The model analyses historical network measurements and location-related information to estimate network availability at upcoming locations.

Architecture Flow:

User → Flutter Mobile Application → GPS Location and Network Measurements

↓

FastAPI Backend / API

↓

Data Preprocessing and Feature Preparation

↓

LightGBM Machine Learning Model

↓

Predicted Network Availability

↓

Network Condition Classification

↓

Interactive Map / Prediction Display

3.1 Main Components:

1. User Interface – This allows users to interact with the NetEyeAI mobile application, access location information, and view predicted network conditions.

2. Frontend – Developed using Flutter, it provides route visualization, GPS-based location handling, and a map interface for displaying predicted network availability.

3. Backend – Developed using FastAPI, this component handles prediction requests from the mobile application, processes input data, and connects the application with the trained machine learning model.

4. Data Processing Component – It prepares historical network measurements and location-related input features in

the required format for the prediction model.

5. ML Model – The system uses the LightGBM classification algorithm to analyse network-related features and predict network availability approximately 2–5 km ahead.

6. Prediction Display and Map Interface – This component displays route information and predicted network conditions by classifying them into four categories: poor or no network, fair, good, and excellent.

3.2 Methodology:

1. Data Collection: Collect historical mobile network measurements and associated geographical information, including GPS coordinates and relevant network-related parameters.

2. Data Preprocessing: Clean and prepare the collected dataset by handling missing values, processing relevant features, and converting the data into a suitable format for machine learning.

3. Feature Preparation: Identify and prepare important network-related and geographical features required for predicting network availability at upcoming locations.

4. Model Training: Train the LightGBM machine learning model using the prepared historical network dataset to learn patterns associated with different network conditions.

5. Network Availability Prediction: Provide the required input measurements and target geographical information to the trained model. The model analyses these inputs and estimates network availability approximately 2–5 km ahead.

6. Network Condition Classification: Classify the predicted network conditions into four categories: poor or no network, fair, good, and excellent, based on the trained model's output.

7. Route-Based Analysis: Use GPS-based location information and route details to associate predicted network conditions with upcoming geographical locations along the user's journey.

8. Map Visualization and Prediction Display: Display the route and predicted network conditions through the Flutter mobile application using an interactive map interface, helping users identify potential weak-network areas in advance.

4. MLOPS INFRASTRUCTURE AND TECHNOLOGY JUSTIFICATION

4.1 MLOps Infrastructure:

1. Data Collection – Collect historical mobile network measurements and geographical information, including GPS coordinates and relevant network-related parameters.

2. Data Preprocessing – Clean and prepare the collected dataset by handling missing values, processing input features, and converting the data into a suitable format for model training.

3. Model Training – Use the prepared network dataset to train the LightGBM machine learning model for predicting network availability.

4. Model Validation – Evaluate the model's performance using suitable classification metrics such as accuracy, precision, recall, and F1-score.

5. Model Deployment – Save the trained LightGBM model and integrate it with the FastAPI backend to provide prediction services through REST API endpoints.

6. Prediction – The Flutter mobile application sends the required network measurements and target-location information to the FastAPI backend. The backend processes the input and returns the predicted network condition.

7. Monitoring and Updating – Monitor prediction performance and system functionality. Update or retrain the machine learning model when suitable new network measurement data becomes available.

4.2 Technology Justification

The proposed NetEyeAI: AI-Based Network Availability Prediction and Alert System combines machine learning, mobile application development, backend API services, and geospatial technologies to predict network availability along a user's route. Each technology plays an important role in processing network measurements, generating predictions, and displaying connectivity information through a mobile application

.4.2.1 Python

Python is used for data preprocessing, machine learning model development, and backend processing. Its libraries support dataset handling, feature preparation, model training, and prediction tasks.

4.2.2 LightGBM

LightGBM is a gradient boosting framework used for machine learning classification. It efficiently processes structured datasets and identifies patterns in historical network measurements. In NetEyeAI, it is used to predict network availability and classify network conditions into four categories: poor or no network, fair, good, and excellent.

4.2.3 FastAPI

FastAPI is used to develop the backend REST API services. It connects the Flutter mobile application with the trained LightGBM model, processes prediction requests, and returns prediction results to the application.

4.2.4 Flutter and Dart

Flutter is used to develop the mobile application, while Dart is its programming language. It provides the user interface, GPS-based location handling, route visualization, and display of predicted network conditions.

4.2.5 MapLibre and Mapping Services

MapLibre and mapping services are used to provide interactive map functionality. They help display geographical locations, routes, and predicted network conditions, allowing users to understand connectivity variations along their journey.

4.2.6 Pandas, NumPy and Scikit-learn

Pandas and NumPy are used for dataset processing, numerical operations, and feature preparation. Scikit-learn supports machine learning preprocessing and evaluation tasks, helping prepare data and assess the prediction model.

4.2.7 Git and GitHub

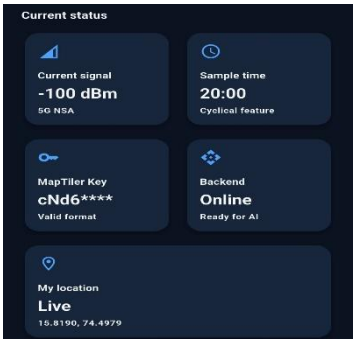
Git and GitHub are used for version control and project management. They help maintain source code, track development changes, and manage collaboration throughout the project.

5. IMPLEMENTATION DETAILS

5.1 User Interface

User Interface – The NetEyeAI system provides a mobile-based interface developed using Flutter. Users can interact with the application to access location information, view routes, and check predicted network availability. The interface presents network prediction results in an understandable format to help users identify potential weak-network areas along their journey.

5.2 GPS and Location Management



GPS and Location Management – The application uses GPS functionality to obtain the user's geographical location. The system uses location information such as:

- Current latitude

- Current longitude

- Target latitude
- Target longitude
- Distance to the target location
- Route direction or bearing

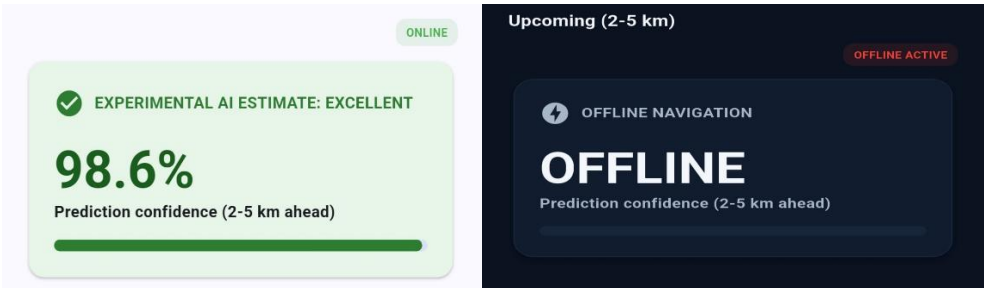
This geographical information helps the system analyse upcoming locations and estimate network availability along the user's route.

5.3 Network Measurement Collection

Network Measurement Collection – The system uses historical mobile network measurements to analyse connectivity conditions. A sequence of 10 network measurements is used as input to the prediction service. These measurements, along with relevant geographical information, provide the input features required by the trained machine learning model.

The collected network information helps identify variations in connectivity and supports the prediction of future network conditions.

5.4 Network Availability Prediction



Network Availability Prediction – The machine learning module uses the LightGBM classification algorithm. The trained model analyses network measurements and relevant location-related features to estimate network availability approximately 2–5 km ahead.

The model classifies the predicted network condition into four categories:

- Poor or No Network
- Fair
- Good
- Excellent

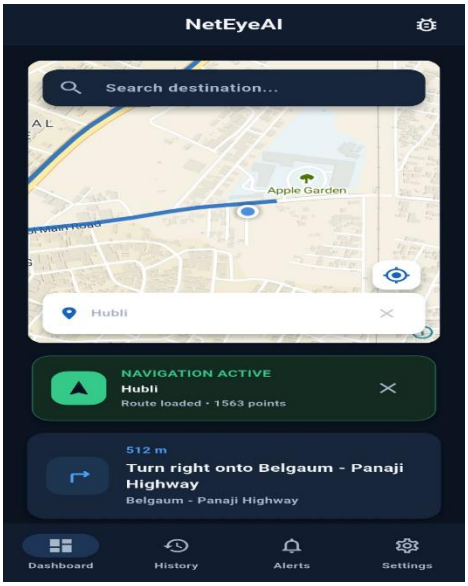
The prediction service processes the input data and returns the estimated network condition to the mobile application.

5.5 Route-Based Network Analysis

Route-Based Network Analysis – The system combines GPS-based location information with historical network measurements to estimate connectivity conditions at upcoming geographical locations. Target-location details, including latitude, longitude, distance, and bearing, are supplied to the prediction service.

This approach helps users understand possible changes in network availability along their journey rather than relying only on their current signal condition.

5.6 Map Visualization



Map Visualization – The application uses MapLibre and mapping services to display geographical information through an interactive map interface. The map provides route visualization and presents predicted network conditions at relevant locations.

This feature helps users identify areas where network connectivity may be poor, fair, good, or excellent, improving their awareness of network availability during travel.

5.7 Telemetry Management

Telemetry Management – The system includes a telemetry component for handling network-related measurement information. The backend provides a dedicated telemetry API endpoint to receive and manage telemetry data. The telemetry service uses SQLite for storing relevant records, with API-key-based access protection.

This information can support network data collection, monitoring, and future improvements to the prediction system.

5.8 Machine Learning Model Management

Machine Learning Model Management – The backend loads the trained LightGBM model and its associated preprocessing and label-encoding components. These components prepare the input measurements in the required format and convert the model output into the corresponding network condition category.

The trained model is integrated with the prediction service so that the application can obtain network availability estimates through API requests**.**

5.9Backend API

Backend API – FastAPI REST API services connect the Flutter mobile application with the machine learning model. The backend handles prediction requests, processes the input information, and returns the prediction results to the application.

6. RESULTS AND DISCUSSION

The developed NetEyeAI: AI-Based Network Availability Prediction and Alert System combines GPS-based location tracking, historical network measurement analysis, and machine learning-based network availability prediction in a mobile application. The system uses a trained LightGBM model integrated with a FastAPI backend to estimate network conditions approximately 2–5 km ahead. The predicted results are presented through a Flutter-based mobile application with an interactive map interface, helping users understand possible variations in network connectivity along their routes.

6.1 Results:

1. Network Availability Prediction: The LightGBM machine learning model analyses historical network measurements and relevant geographical information to estimate upcoming network conditions.

2. Network Condition Classification: The system classifies predicted network availability into four categories: poor or no network, fair, good, and excellent.

3. GPS-Based Location Tracking: The application uses GPS coordinates and target-location information to support route-based network availability prediction.

4. Route Visualization: The interactive map interface displays geographical routes and predicted network conditions, helping users identify potential weak-network areas.

5. Backend Integration: The FastAPI backend processes prediction requests, communicates with the trained LightGBM model, and returns prediction results to the Flutter mobile application.

6. Telemetry Management: The system provides a telemetry API for handling network-related measurement information and uses SQLite for storing relevant telemetry records.

6.2 Discussion:

The proposed NetEyeAI system provides a location-aware approach to mobile network monitoring by combining historical network measurements, GPS information, and machine learning-based prediction. Instead of focusing only on the current signal condition, the system estimates network availability approximately 2–5 km ahead, helping users become aware of potential connectivity changes before reaching upcoming locations.

By incorporating the LightGBM classification model, the system analyses network-related input features and classifies predicted connectivity into four categories. The integration of the Flutter mobile application and FastAPI backend enables prediction requests to be processed and the resulting network conditions to be displayed through an interactive map interface.

The map-based visualization makes it easier for users to understand geographical variations in network availability and identify potential weak-network areas along their journey. Overall, NetEyeAI demonstrates the application of machine learning and geospatial technologies for route-based network availability prediction. It aims to improve connectivity awareness and support users in making informed decisions while travelling through areas with varying mobile network coverage

7. COMPARISON WITH EXISTING APPROACHES

The existing approach to mobile network monitoring mainly relies on checking current signal strength, network type, internet connectivity, and available network coverage information through mobile devices and network monitoring applications. Users generally depend on the signal indicators displayed on their smartphones to understand the quality of their current network connection. Although this approach is useful for checking present connectivity conditions, it provides limited information about network availability at upcoming geographical locations. Users may encounter weak signals or complete network loss while travelling without knowing the connectivity conditions ahead of their route.

Network coverage information is also commonly accessed through coverage maps and previously recorded network measurements. These methods can help users understand network availability in different geographical areas. However, network conditions may change from one location to another due to geographical conditions, physical obstacles, and variations in network infrastructure. Relying only on current signal measurements or general coverage information may make it difficult for users to anticipate network connectivity changes along their journey.

The proposed NetEyeAI: AI-Based Network Availability Prediction and Alert System aims to address these limitations by providing a location-aware and machine learning-based approach to network monitoring. The system combines GPS-based location information, historical network measurements, and geospatial analysis to estimate network availability approximately 2–5 km ahead. Instead of relying only on current signal strength, the system uses the LightGBM machine learning algorithm to analyse relevant input features and predict upcoming network conditions. This provides an additional decision-support mechanism for identifying potential weak-network areas before users reach those locations.

The proposed system also provides an interactive map interface that displays route information and predicted network conditions. The predicted network quality is classified into four categories: poor or no network, fair, good, and excellent. The Flutter mobile application communicates with the FastAPI backend, which processes prediction requests and obtains results from the trained LightGBM model. This integration helps users understand possible

changes in network availability across different geographical locations along their route.

Overall, existing network monitoring approaches primarily focus on current signal conditions and general coverage information, whereas NetEyeAI introduces a route-aware and predictive approach to network availability monitoring. By combining GPS-based location tracking, historical network measurement analysis, LightGBM-based prediction, and map visualization in one mobile application, the proposed system aims to improve network awareness and help users make informed decisions while travelling through areas with varying mobile network coverage.

8. LIMITATIONS

1. Dataset Dependency: The reliability of network availability predictions depends on the quality, quantity, and geographical coverage of the historical network measurement dataset.

2. Prediction Accuracy: The LightGBM model may not always accurately predict network conditions at every location, particularly when actual network conditions differ from the patterns present in the training dataset.

3. Historical Data Dependency: The system relies on previously collected network measurements. If sufficient historical measurements are unavailable for a particular geographical area, prediction reliability may be affected.

4. Limited Geographical Coverage: The prediction capability depends on the availability of network measurement data for different geographical locations. Areas with limited or insufficient data may not produce reliable predictions.

5. Internet Dependency: The application depends on internet connectivity for communication with the FastAPI backend and accessing online mapping services. Network interruptions may affect prediction requests and map functionality.

6. Prediction Limitations: The predicted network conditions are estimates and may differ from actual connectivity conditions due to changes in network traffic, infrastructure, environmental conditions, and other external factors.

7. Environmental and Geographical Factors: Physical obstacles, terrain variations, weather conditions, and changes in mobile network infrastructure can influence signal strength. The system may not capture all these factors when generating predictions.

8. Real-Time Data Collection: The prediction system depends on the availability and quality of network measurements. Limitations in collecting continuous and updated measurements may affect the reliability of future predictions.

9. CONCLUSION AND FUTURE SCOPE

The NetEyeAI: AI-Based Network Availability Prediction and Alert System provides a location-aware approach to predicting mobile network availability along a user's route. The system combines GPS-based location tracking, historical network measurement analysis, machine learning-based prediction, and map visualization in a single mobile application. By utilizing the LightGBM machine learning algorithm, it estimates network conditions approximately 2–5 km ahead and classifies them into four categories: poor or no network, fair, good, and excellent. The Flutter mobile application and FastAPI backend work together to process prediction requests and display the estimated network conditions through an interactive map interface. Overall, the system aims to improve network awareness, help users identify potential weak-network areas in advance, and support better decision-making while travelling through locations with varying mobile network coverage.

Looking ahead, the team has identified several areas for future development:

1. Advanced AI Models: Exploring advanced machine learning and deep learning algorithms could improve network availability prediction and help identify more complex patterns in historical network measurements.

2. Real-Time Network Monitoring: Integrating continuous network measurement collection could provide updated information about network conditions and support more frequent predictions.

3. Offline Prediction Support: Developing an offline prediction capability could allow users to access previously stored network information and predictions in areas with limited internet connectivity.

4. Cloud Deployment: Deploying the FastAPI backend and machine learning model on cloud platforms could improve scalability, availability, and accessibility for a larger number of users.

5. Automated Alerts and Notifications: Integrating push notifications, SMS, or other alert mechanisms could inform users about upcoming weak-network or no-network areas along their routes.

6. Expanded Geographical Coverage: Collecting network measurements from more geographical locations and different mobile network operators could improve the coverage and usefulness of the prediction system.

7. Multiple Network Operator Support: Extending the system to analyse and compare network availability across different telecom operators could help users understand connectivity variations and make better network-related decisions.

8. Improved Map-Based Analysis: Enhancing the map interface with detailed network coverage visualization, route-based connectivity analysis, and historical network information could provide users with a more comprehensive understanding of network availability.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to Prof. P L Halgekar for his invaluable guidance and supervision during this project. They also extend their thanks to the Angadi Institute of Technology and Management, specifically the Department of Artificial Intelligence and Data Science, for providing the resources and support that made this work achievable.

References

[1] M. F. A. Fauzi, R. Nordin, N. F. Abdullah, and H. A. H. Alobaidy, "Mobile Network Coverage Prediction Based on Supervised Machine Learning Algorithms," IEEE Access, vol. 10, pp. 55782–55793, 2022. DOI: 10.1109/ACCESS.2022.3176619.

[2] O. Sahin and V. Sathya, "Network Quality Prediction in a Designated Area Using GPS Data," Journal of Network and Computer Applications, vol. 231, article 104002, 2024. DOI: 10.1016/j.jnca.2024.104002.

[3] S. Zeng, Y. Ji, W. Chen, L. Yan, and X. Zhao, "Mobile Network Coverage Prediction Using Multi-Modal Model Based on Deep Neural Networks and Semantic Segmentation," Sensors, vol. 24, no. 16, article 5178, 2024. DOI: 10.3390/s24165178.

[4] R. Azoulay, E. Edery, Y. Haddad, and O. Rozenblit, "Machine Learning Techniques for Received Signal Strength Indicator Prediction," Intelligent Data Analysis, vol. 27, no. 4, 2023. DOI: 10.3233/IDA-226750.

[5] G. M. Perihanoglu and H. Karaman, "Spatial Prediction of Received Signal Strength for Cellular Communication Using Support Vector Machine and K-Nearest Neighbours Regression," The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLVIII-4/W9, pp. 291–297, 2024. DOI: 10.5194/isprs-archives-XLVIII-4-W9-2024-291-2024.

[6] P. Fraccaro, M. Benatan, K. Reusch, C. Fare, B. Edwards, and E. Pyzer-Knapp, "Modelling Mobile Signal Strength by Combining Geospatial Big Data and Artificial Intelligence," 2020.

[7] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," Advances in Neural Information Processing Systems, vol. 30, pp. 3149–3157, 2017.

[8] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[9] FastAPI, "FastAPI Documentation." [Online]. Available: https://fastapi.tiangolo.com/

[10] Flutter, "Flutter Documentation." [Online]. Available: https://docs.flutter.dev/

[11] MapLibre, "MapLibre Documentation." [Online]. Available: https://maplibre.org/