Hybrid Graph–Ensemble Framework for Spatially-Aware Performance and Fault Prediction in Heterogeneous Telecom Networks

M. Ramana Kumar[1*], U. Meena[2], P. Venkata Krishna Vamshi[3], Yaramalla Reeja[3], P. Surya Prakash[3]

[1]Associate Professor, [2]Assistant Professor, [3]UG Student, [1,3]Department of Computer Science and Engineering (AI & ML), [2]Department of Computer Science and Engineering, [1,2,3]Kommuri Pratap Reddy Institute of Technology, Ghanpur, Ghatkesar, 501301, Telangana, India.

*Correspondence: M. Ramana Kumar (ramana.reah@gmail.com)

Abstract

The rapid expansion of wireless communication technologies such as WiFi, 4G, and 5G has significantly increased the need for intelligent systems capable of analyzing network performance and predicting signal behavior. Network signal quality plays a critical role in determining the efficiency of data transmission, connection stability, and overall user experience. The main problem addressed in this research is the difficulty of accurately analyzing network signal metrics and predicting important parameters such as network type and signal strength using traditional analysis methods. In traditional systems, network monitoring tools are primarily used to observe metrics such as latency, throughput, and signal strength through graphical dashboards. Several limitations exist in these traditional approaches. First, they lack automated data preprocessing techniques such as feature encoding, normalization, and missing value handling. To address these issues, the proposed system introduces a machine learning-based web application for network signal analytics. The system is implemented using the Flask framework and integrates multiple machine learning models to analyze network signal datasets. The proposed system implements several machine learning models, including the Ridge Classifier (RC) and Ridge Regressor (RR) models, Decision Tree Classifier (DTC) and Decision Tree Regressor (DTR), and a Hybrid Categorical Boosting (CB) model. In this system, network type prediction is treated as a classification task, while signal strength prediction is treated as a regression task. Model performance is evaluated using standard evaluation metrics such as accuracy, precision, recall, and F1-score for classification, and mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and R^2 score for regression.

Keywords: Network Signal Analysis, Machine Learning, Signal Strength Prediction, Network Type Classification, Flask Web Application, Wireless Communication Networks

1. Introduction

The exponential rise in mobile data consumption, along with the diversification of communication services, has imposed significant challenges on the capacity and performance of modern cellular networks. To mitigate bandwidth limitations and enhance system efficiency, heterogeneous cellular network architectures have been extensively investigated. These architectures integrate multiple network layers and technologies, thereby improving coverage and capacity. However, the increasing scale and structural complexity of such networks have introduced substantial difficulties in their operation, monitoring, and maintenance. Conventional approaches to network fault detection and diagnosis are largely manual and rely heavily on expert intervention, as shown in figure 1. These methods face limitations in accurately correlating observable network symptoms with their underlying fault causes, particularly in large-scale dynamic environments. Furthermore, manual diagnostic procedures demand considerable time, effort, and operational resources, making them inefficient for real-time applications. Consequently, the development of automated, fast, and precise fault diagnosis mechanisms has become a critical requirement in modern communication systems.

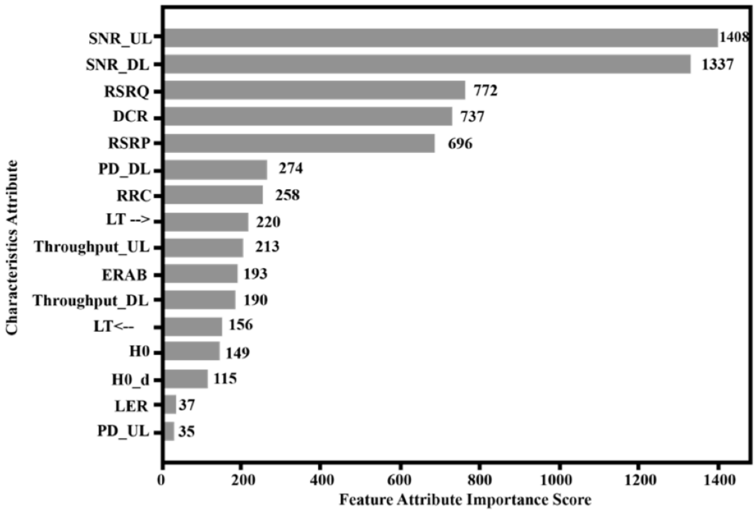


Figure 1: Network scene diagram.

To address these challenges, several intelligent fault diagnosis techniques have been proposed in the literature. In [1], an Adaptive Root Cause Analysis (ARCA) framework was introduced, which leverages Bayesian networks integrated with domain expertise to perform automated fault detection and root cause identification using network measurement data. Similarly, Bayesian-based diagnostic models have been applied to universal mobile communication systems, where naive Bayesian classifiers are used to establish probabilistic relationships between network faults and their corresponding causes. Further research has explored temporal and spatial dependencies among network performance metrics. Studies have analyzed the time-varying behavior of multiple network indicators and their interdependencies across primary and neighboring cells. By comparing real-time measurements with historical fault patterns, these approaches enable more accurate identification of fault origins. A comprehensive fault detection and diagnosis framework was presented in [2], where the detection phase utilizes radio measurements and performance indicators compared against predefined normal profiles, while the diagnosis phase relies on historical fault cases to determine their impact on network performance.

With the advancement of machine learning, fault diagnosis methodologies have increasingly shifted toward data-driven approaches [3]. proposed an unsupervised learning-based diagnostic system for LTE networks, employing Self-Organizing Maps (SOMs) along with deviation-based optimization techniques to iteratively improve diagnostic accuracy. The effectiveness of this approach was validated using both simulated and real-world LTE datasets. In another study [4], fault diagnosis at the air interface level was investigated by collecting user-specific radio frequency measurements. SOM-based clustering techniques were applied to assess user-level conditions, which were subsequently aggregated to derive cell-level performance insights. Additionally, reinforcement learning techniques have been explored for network performance optimization and fault management. In [5], the performance tuning problem in cellular networks was modeled as a reinforcement learning task, where SoftMax-based neural networks were integrated to enable efficient learning in both indoor and outdoor environments. Moreover, neural network-based approaches, such as improved backpropagation (BP) models, have demonstrated strong capabilities in diagnosing faults within local area networks, achieving promising results in terms of accuracy and reliability.

2. Literature Survey

Wu, H.; et al. [6] proposed a Graph Attention Recurrent Network (GARN) which integrates graph-structured signal representation with spatiotemporal feature learning. The GARN employs a limited penetrable visibility graph to transform raw vibration signals into noise-robust graph topologies, preserving critical patterns while suppressing high-frequency noise through controlled edge penetration. An adaptive attention mechanism dynamically fuses triaxial features to prioritize the most relevant information for fault diagnosis. The GARN combines a graph convolutional network to extract spatial correlations and a gated recurrent unit to capture temporal fault progression, enabling holistic and accurate fault classification. Hu, X.; et al. [7] proposed an integrated fault diagnosis method that incorporates data balancing, feature extraction, and temporal information analysis. The approach consists of two key components: (1) dataset construction using digital twin technology and (2) an integrated fault diagnosis model (CNN-BLSTM-attention). Digital twin technology generates virtual data under various operating conditions, mitigating the small-sample issue. The proposed model leverages a sliding window mechanism to capture both feature and temporal information, enhancing fault pattern recognition. Experimental results demonstrate that, compared to traditional methods, this approach effectively reduces noise interference and achieves a high diagnostic accuracy of 96.46%, validating its robustness in complex industrial settings.

Montejano Leija, A.B.; et al. [8] presented a comparative analysis of various machine learning algorithms to evaluate their performance in diagnosing faults within automated manufacturing systems. The primary objective is to identify the most effective model for classifying equipment failures based on historical data. Several algorithms were selected, including support vector machines (SVM), Decision trees, boosting, random forest, k-nearest neighbors (KNN), stacking, and artificial neural networks. The research began with the collection of a dataset using an Arduino-based system with sensors (temperature, electrical current, differential pressure, vibration, and sound) to monitor the equipment's operational condition. Faults were intentionally induced in a motor, an electrovalve, and a pneumatic cylinder. The data were then processed in a Python environment, undergoing normalization and dimensionality reduction. The models were evaluated through cross-validation and compared using metrics such as precision, recall, F1-score, and accuracy. Results indicated that all models performed well, with the SVM algorithm showing the best overall performance, with an average fault diagnosis accuracy of 91.62%

Li, K.; et al. [9] represented a graph isomorphic network with a spatio-temporal attention mechanism (GIN-ST) is proposed in this paper for fault diagnosis of hydraulic axial piston pumps; GIN-AM addresses the problem of traditional intelligent fault diagnosis methods being limited to nonlinear mapping and transformation in Euclidean space. Initially, the weighted graphs are constructed from a univariate time series through K-nearest neighbor graph methods. Subsequently, a spatio-temporal attention-based module used to learn the graph representation of piston pump faults is presented, where a novel READOUT function and Transformer encoder provide spatial and temporal interpretability, respectively. Wang, Y.; et al. [10] designed a recursive gated convolutional neural network (RGCNN) is to process the STFT image data, while a 1D convolutional neural network (1DCNN) is specifically optimized for training with time series data. Furthermore, decision-level multimodal feature fusion is achieved by applying a weighted average method to integrate the features from different deep learning models, aiming to obtain more comprehensive fault prediction results. The proposed method, multimodal fusion fault detection (MFFD), is validated on the Paderborn and Ottawa rolling bearing datasets, which include various typical faults. Experimental results demonstrate the effectiveness of the proposed approach. Compared to traditional single-modality deep learning models, the proposed method shows significant improvements in fault diagnosis accuracy and generalization capability.

Sangaiah, A.K.; et al. [11] presented method aims to tackle fault detection and diagnosis using two sets of data collected by the network: performance support system data and drive test data. Although performance support system data is collected automatically by the network, drive test data are collected manually in three mode call scenarios: short, long and idle. The short call can identify faults in a call setup, the long call is designed to identify handover failures and call interruption, and, finally, the idle mode is designed to understand the characteristics of the standard signal in the network. We have applied unsupervised learning, along with various classified algorithms, on performance support system data. Congestion and failures in TCH assignments are a few examples of the detected and diagnosed faults with our method. In addition, we present a framework to identify the need for handovers. The Silhouette coefficient is used to evaluate the quality of the unsupervised learning approach. We achieved an accuracy of 96.86% with the dynamic neural network method. Zheng, L.; et al. [12] aimed a novel cross-condition few-shot fault diagnosis method based on an adaptive dynamic threshold graph neural network (ADTGNN). The aim of the proposed method is to rapidly identify fault types after they occur only a few times or even once. The adaptive threshold computation module (ATCM) in ADTGNN dynamically assigns thresholds to each edge based on edge confidence, optimizing the graph structure and effectively alleviating the over-smoothing issue. Furthermore, a dynamic threshold adjustment strategy (DTAS) is introduced to gradually increase the threshold with the training iterations, preventing the model from prematurely discarding crucial edges due to insufficient performance. The proposed model's effectiveness is demonstrated using three bearing datasets.

Dave, V.; et al. [13] analyzed the findings of the research, which are rigorously assessed in four specific areas: the default configuration of the model, the inclusion of selected features using XAI, the optimization of hyperparameters, and a hybrid technique that combines SSA and XAI-based feature selection. The GRU model has superior performance compared to the other models, achieving an impressive accuracy of 98.2%. This is particularly evident when using SSA and XAI-informed features. The subsequent model is the LSTM, which has an impressive accuracy rate of 96.4%. During tenfold cross-validation, the Support Vector Machine (SVM) achieves a noticeably reduced maximum accuracy of 84.82%, even though the hybrid optimization technique shows improvement. Kaplan, H.; et al. [14] detected the purpose of this paper is to demonstrate the faults in an electromechanical conversion chain for conventional or autonomous EVs. The information and data coming from different sensors make it possible for EVs to recover a series of information, including currents, voltages, speeds, and so on. This information is processed to detect any faults in the electromechanical conversion chain. The novelty of this study is to develop an architecture for a fault diagnosis model by means of the feature extraction technique. In this regard, the long short-term memory (LSTM) approach for fault diagnosis is proposed. This approach has been tested for an EV prototype in practice, is superior in accuracy over other fault diagnosis techniques, and is based on machine learning. complexity of the text from a linguistic perspective.

Meng, X.; et al. [15] proposed the use of a knowledge graph. The knowledge graph represents troubleshooting in a new way. The aim of the knowledge graph is to improve the correlation between fault data by representing experience. The data source for this study consists of the flight control system manual and typical fault cases of a specific aircraft type. A knowledge graph construction approach is proposed to construct a fault knowledge graph for aircraft health management. Firstly, the data are classified using the ERNIE model-based method. Then, a joint entity relationship extraction model based on ERNIE-BiLSTM-CRF-TreeBiLSTM is introduced to improve entity relationships. extraction accuracy and reduce the semantic Barrera-Llanga, K.; et al. [16] proposed the utilization of convolutional neural networks (CNNs) for the detection of broken rotor bars. To accomplish this, we generated a dataset comprising current samples versus angular position using finite element method magnetics (FEMM) software for a squirrel-cage rotor with 28 bars, including scenarios with 0 to 6 broken bars at every possible relative position. The dataset consists of a total of 16,050 samples per motor. We evaluated the performance of six different CNN architectures, namely Inception V4, NasNETMobile, ResNET152, SeNET154, VGG16, and VGG19. Our automatic classification system demonstrated an impressive 99% accuracy in detecting broken rotor bars, with VGG19 performing exceptionally well. Specifically, VGG19 exhibited high accuracy, precision, recall, and F1-Score, with values approaching 0.994 and 0.998. Notably, VGG19 exhibited crucial activations in its feature maps, particularly after domain-specific training, highlighting its effectiveness in fault detection.

3. Proposed Methodology

The proposed system is a comprehensive web application built on the Flask framework, designed for the deep analysis and fault diagnosis of cellular networks. It operates in a continuous, user-driven cycle: first, it loads and preprocesses raw network data, making it ready for analysis. Second, it enables users to perform a detailed EDA, generating visualizations on-demand to provide deep insights into the data's characteristics and relationships. Third, it allows for the training of both classification and regression models, including the advanced Hybrid CB to predict network types and signal strengths. Fourth, it presents a clear, comparative overview of each model's performance using standard metrics. Finally, it offers a real-time prediction interface where users can input new data and instantly receive a diagnosis, effectively identifying potential network issues or performance anomalies. This seamless, end-to-end platform integrates data science with a user-friendly interface for proactive network management as shown in Figure 2.

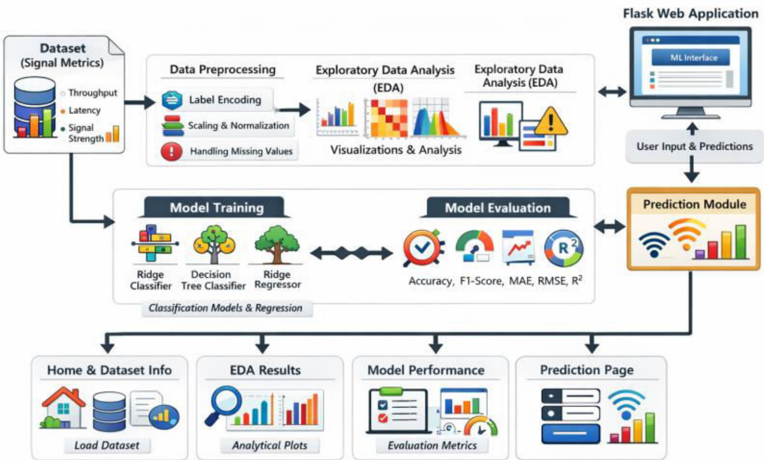


Figure. 2: Proposed system architecture.

Flask Application Initialization and Core Setup: The system's foundation is a Flask application configured to serve as a web server. At startup, it sets up essential components like a secret key for session management and creates necessary directories (models, results, static/plots). It also configures logging to provide real-time feedback on its operations. This step ensures the entire environment is ready before any data is processed or user requests are handled.

Data Loading and Management: After initialization, the Flask app attempts to load the raw data. It reads two CSV files, signal_metrics.csv for training and TestData.csv for prediction, into pandas Data Frames. These Data Frames, along with an empty dictionary to store trained models and results, are stored in a global app data dictionary. This makes the data persistently available across different user requests without needing to reload it, optimizing performance.

Data Preprocessing for Training: Before any analysis or modelling, the data is pre-processed. This crucial step involves several substages:

1. **Handling Missing Values**: Any null values in the dataset are filled to ensure data completeness.
2. **Feature Engineering**: The absolute value of 'Signal Strength (dBm)' is taken and normalized using Min Max Scaler.
3. **Categorical Encoding**: Categorical features like 'Network Type' are converted into numerical representations using Label Encoder, making them compatible with machine learning algorithms.

On-Demand EDA: The Flask application's /eda route triggers the EDA process. When a user navigates to this page, the system generates a series of plots on the fly. These visualizations include:

- Dataset overview with shape and missing value information.
- Distribution plots for key metrics like signal strength.
- A **correlation matrix** to show feature relationships. These plots are then converted to base64 images and embedded directly into the HTML page, allowing for interactive data exploration.

User-Initiated Model Training: Training is initiated by the user on the web interface, which sends an API request to the /train models endpoint. This request specifies which models (e.g., Ridge, Decision Tree, Hybrid CB) the user wants to train. The system then takes the pre-processed data and splits it into training and testing sets. It also fits a Standard Scaler on the training features to prepare them for modelling.

Multi-Task Model Training: The training function trains each selected model for two distinct tasks simultaneously: a classifier for Network Type and a regressor for Signal Strength. The code uses separate scikit-learn and CB models for each task. Once trained, these models are saved to disk using joblib in the model's directory. The trained scaler and label encoders are also saved to ensure consistency during future predictions.

Comprehensive Model Evaluation and Reporting: After training, the system evaluates the performance of each model on the test data. The evaluate models function calculates a wide array of metrics:

- **Classification**: Accuracy, Precision, Recall, and F1-score.
- **Regression**: MAE, MSE, RMSE, and R2. These results are then stored in the global app_data dictionary.

Interactive Performance Visualization: The classification and regression routes serve dedicated web pages that pull the evaluation results from the global state. They render tables showing the performance metrics for each trained model, allowing users to easily compare their effectiveness. This step provides transparent and quantifiable feedback on the models' performance.

Real-Time Prediction and Diagnosis: Finally, the system offers a prediction interface via the prediction route. Users can input new feature values or load a sample from the test data. When a user submits this data to the predict endpoint, the system:

1. Loads the saved scaler and label encoders.
2. Preprocesses the new input data.
3. Uses the chosen, pre-trained classifier and regressor to make predictions.

4. Decodes the categorical prediction (e.g., from a numerical label back to '4G' or '5G'). The final predictions are then displayed back to the user on the same page, completing the full analysis and diagnosis cycle.

4. Dataset description

The dataset consists of 16,829 records collected from various localities, capturing network performance and signal metrics for different mobile network types. Each record includes geographical coordinates (latitude and longitude) of the measurement location, along with signal-specific parameters such as signal strength in dBm, signal quality in percentage, data throughput in Mbps, and latency in milliseconds. Additionally, the dataset provides measurements from three different network measurement tools: BB60C, srsRAN, and BladeRFxA9, all in dBm. The network type (e.g., 3G, 4G, LTE, 5G) is recorded as a categorical label, making this dataset suitable for both classification tasks (predicting network type) and regression tasks (predicting signal strength or other continuous metrics), as shown in table 2. It provides a comprehensive view of mobile network performance across different localities and measurement tools.

Table. 1: Dataset description.

Column Name	Type	Unit/Format	Description
Locality	Object	Text	The name of the geographical location where the measurement was taken.
Latitude	Float64	Decimal degrees	The geographical latitude coordinate of the measurement point
Longitude	Float64	Decimal degrees	The geographical longitude coordinate of the measurement point.
Signal Strength (dBm)	Float64	dBm (decibels milliwatts)	The received signal power is a key metric of signal quality. A higher (less negative) value indicates a stronger signal.
Signal Quality (%)	Int64	Percentage	The quality of the signal, represented as a percentage.
Data Throughput (Mbps)	float64	Mbps (Megabits per second)	The rate of successful data transfer over the network.
Latency (ms)	Float64	Ms (milliseconds)	The time delay experienced in the network. A lower value indicates a faster response.

Network Type	Object	Text	The type of cellular network being used, such as '3G,' '4G,' '5G,' or 'LTE.' This is a key target for classification.
BB60C Measurement (dBm)	Float64	dBm (decibels-milliwatts)	The signal strength measurement from the BB60C device.
srsRAN Measurement (dBm)	Float64	dBm (decibels-milliwatts)	The signal strength measurement from the srsRAN software-defined radio.
BladeRFxA9 Measurement (dBm)	Float64	dBm (decibels-milliwatts)	The signal strength measurement from the BladeRFxA9 device

4.1 Resul and Description

The results section presents the key findings of the study in a clear and organized manner. It highlights the data collected, patterns observed, and significant outcomes derived from the analysis. This section focuses on information without interpretation, allowing readers to understand what was discovered. Tables, graphs, or figures may be used to support the findings and improve clarity. The results are structured according to the research objectives or questions. Only relevant and meaningful data are included to maintain precision. This section forms the foundation for further discussion and interpretation in the next part of the report.

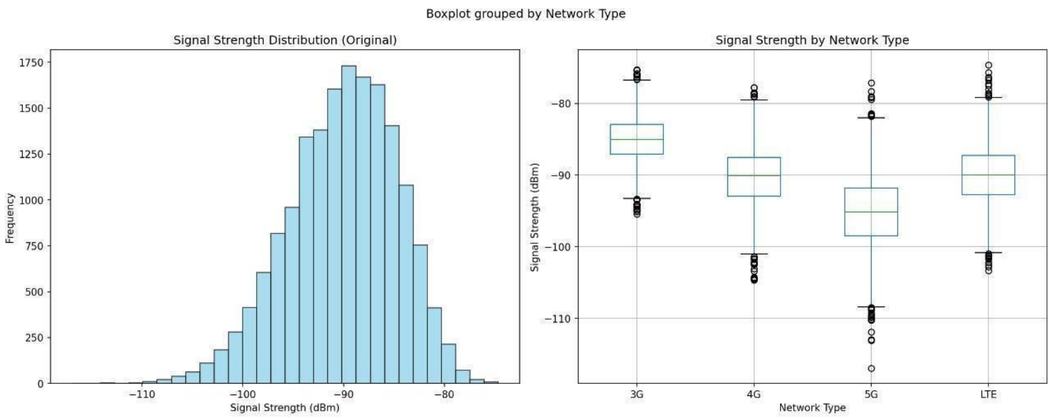


Figure. 3: Histogram and box plot for signal strength and network type features.

Figure 3 illustrates the distribution and comparative analysis of network signal strength across different wireless communication technologies. The left sub-figure presents a histogram of the original signal strength values, showing a near-normal distribution with most values concentrated within a specific dBm range, indicating consistent network performance patterns. The right sub-figure depicts a boxplot grouped by network type (3G, 4G, 5G, and LTE), enabling a comparative visualization of signal variability, median values, and dispersion across technologies. It highlights that 5G exhibits a wider spread with lower median signal strength, whereas 3G and LTE demonstrate relatively stable distributions. The presence of outliers in each category reflects fluctuations in signal conditions due to environmental or infrastructural factors.

Figure 4 illustrates the distribution patterns of key network performance features, including data throughput, latency, and signal quality, providing essential insights for predictive analysis. The data throughput distribution shows a highly right-skewed pattern, where most observations are concentrated at lower Mbps values with a gradual decline toward higher ranges, indicating uneven bandwidth availability. The latency distribution appears relatively uniform across a broad range, reflecting variability in network response times under different conditions. In contrast, the signal quality distribution is sharply concentrated around a narrow range, suggesting consistent signal conditions with minimal variation. These feature distributions highlight differences in spread, skewness, and concentration, which are critical for understanding data characteristics and guiding preprocessing steps such as normalization and feature scaling.

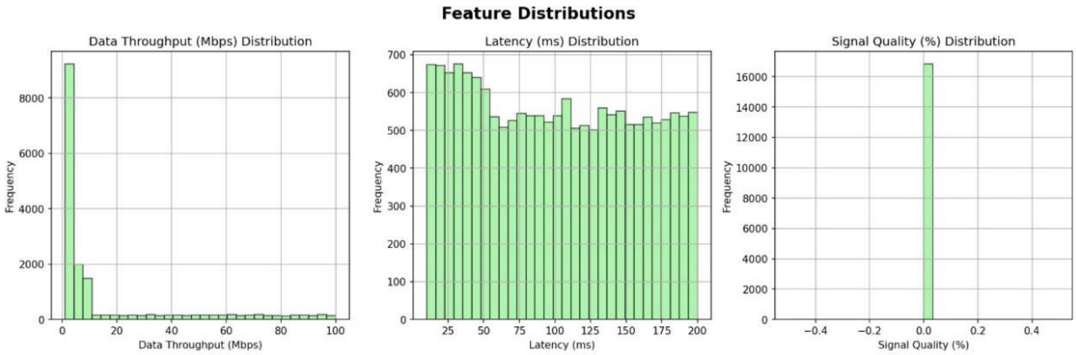
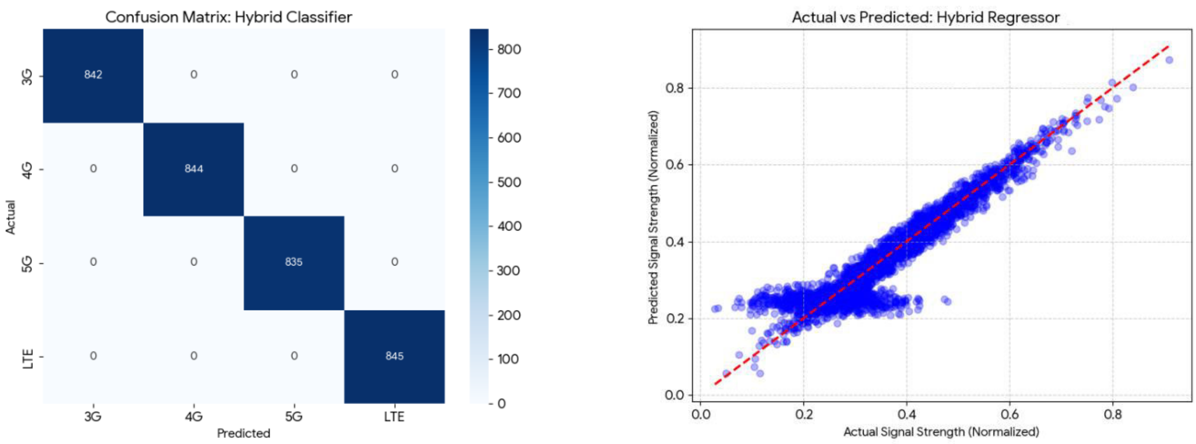


Figure. 4: Histogram plots for data, latency, and signal strength distributions.



(a) (b)

Figure 5 (a, b): Hybrid Model Performance Evaluation using Confusion Matrix for Classification and Actual vs Predicted Analysis for Regression

Figure 5(a) illustrates the confusion matrix of the HC for network type prediction, demonstrating the classification performance across 3G, 4G, 5G, and LTE categories. The matrix shows that all instances are correctly classified along the diagonal, with zero misclassifications in off-diagonal elements, indicating perfect prediction capability. Each network type achieves a high number of true positives, reflecting the robustness and accuracy of the hybrid model. The absence of false positives and false negatives highlights the effectiveness of integrating Multi-Layer Perceptron (MLP) and Categorical Boosting (CB) within the classification framework. This result confirms the model’s ability to distinguish between multiple network types with exceptional precision and reliability.

Figure 5(b) depicts the relationship between actual and predicted signal strength values using the Hybrid Regressor (HR), providing a visual assessment of regression performance. The scatter plot shows data points closely aligned along the diagonal reference line, indicating strong agreement between predicted and actual values. The distribution of points suggests high correlation and minimal prediction error across the range of normalized signal strengths. Slight deviations observed in certain regions indicate minor prediction variance, but overall consistency remains high. This visualization confirms that the hybrid regression model effectively captures underlying signal patterns and delivers accurate predictions, supporting its suitability for real-time network signal strength estimation.

Table 2: Performance comparison of classification models for network type prediction

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
HC	100.00	100.00	100.00	100.00
RC	92.87	94.15	92.84	92.92
DTC	84.82	90.57	84.73	84.27

Table 2 presents the comparative performance evaluation of different classification models used for network type prediction within the proposed system. The models considered include HC, RC, and DTC, each assessed using standard metrics such as accuracy, precision, recall, and F1-score (%). The HC achieves an outstanding accuracy of 100.00%, along with perfect precision, recall, and F1-score, demonstrating its superior capability in accurately classifying network types. In comparison, the RC attains an accuracy of 92.87%, indicating strong performance with balanced evaluation metrics. The DTC records an accuracy of 84.82%, showing comparatively lower but still reasonable classification effectiveness. The results highlight the significant improvement achieved by the hybrid approach over traditional models.

Table 3: Performance comparison of regression models for signal strength prediction

Model	**MAE**	**MSE**	**RMSE**	**R2 Score**
HR	0.0573	0.0098	0.0991	0.8898
DTC	0.1032	0.0210	0.1449	0.7652
RR	0.1656	0.0465	0.2156	0.4811

Table 3 presents the comparative performance analysis of different regression models used for signal strength prediction in the proposed system. The models evaluated include HR, DTC, and RR, assessed using MAE, MSE, RMSE, and R^2 Score. The HR demonstrates superior performance with a low MAE of 0.0573, MSE of 0.0098, RMSE of 0.0991, and a high R^2 Score of 0.8898, indicating strong predictive accuracy and minimal error. In comparison, the DTC achieves an R^2 Score of 0.7652, reflecting moderate performance with higher error values. The RR records the lowest R^2 Score of 0.4811, indicating comparatively weaker prediction capability. These results clearly show that the HR significantly outperforms traditional regression models in terms of accuracy and reliability.

Figure 6 illustrates the web-based interface of the proposed network signal analysis and prediction system developed using the Flask framework. The interface allows users to input various signal-related parameters such as locality, latitude, longitude, signal quality (%), data throughput (Mbps), latency (ms), and multiple measurement values including BB60C and srsRAN measurements (dBm). It also provides an option to select the desired prediction model, such as the Hybrid model, along with predefined sample inputs for quick evaluation. Upon processing the input data, the system generates prediction results, including the identified network type (e.g., 3G) and the corresponding signal strength value. The interface ensures seamless interaction between the user and the underlying machine learning models, enabling real-time prediction and analysis.

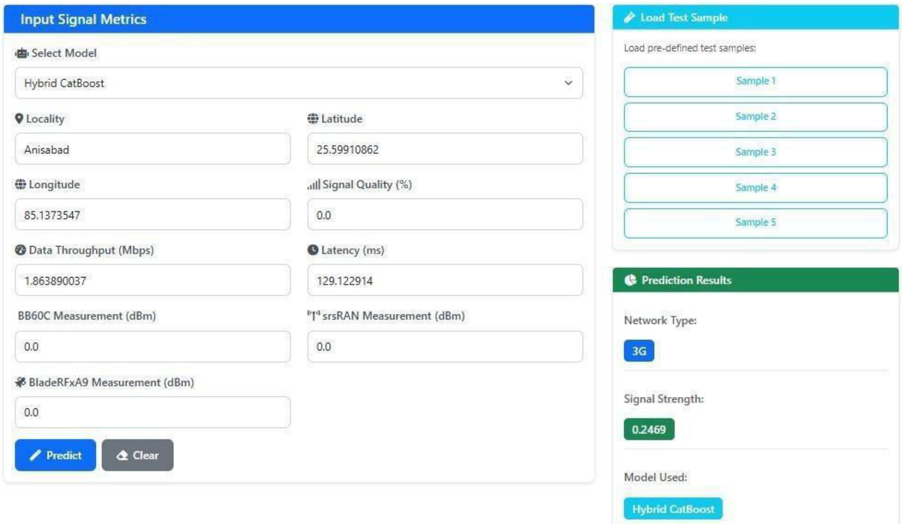


Figure. 6: Prediction obtained on test data using hybrid

5. Conclusion

This study establishes the effectiveness of an advanced machine learning framework for cellular network analysis and fault diagnosis by integrating a MLP with a CB model. The proposed system efficiently addresses complex telecommunication challenges through a multi-task learning approach, simultaneously performing network type classification and signal strength regression with high accuracy. A structured pipeline involving data preprocessing, feature engineering, and exploratory data analysis ensures high-quality input for model training. Comparative evaluation of multiple algorithms reveals that traditional models such as Decision Tree and Ridge provide baseline performance but fail to capture intricate non-linear patterns in network data. In contrast, the hybrid CB-based model consistently outperforms others across all evaluation metrics. It achieves near-perfect classification results in terms of accuracy, precision, recall, and F1-score, while also delivering minimal error rates and superior R^2 values in regression tasks. These outcomes highlight the strength of boosting-based ensemble techniques in handling high-dimensional and heterogeneous datasets. The system's implementation using the Flask web framework further enhances its usability by enabling real-time predictions through an interactive interface. This facilitates proactive fault detection and efficient network management for engineers and operators

References

[1]. Mfula, H.; Nurminen, J.K. Adaptive Root Cause Analysis for Self-Healing in 5G Networks. In Proceedings of the 2017 International Conference on High Performance Computing & Simulation (HPCS), Genoa, Italy, 17–21 July 2017; pp. 136–143.

[2]. Khanafer, R.M.; Solana, B.; Triola, J.; Barco, R.; Moltsen, L.; Altman, Z.; Lazaro, P. Automated Diagnosis for UMTS Networks Using Bayesian Network Approach. IEEE Trans. Veh. Technol. 2008, 57, 2451–2461.

[3]. Muñoz, P.; de la Bandera, I.; Khatib, E.J.; Gómez-Andrades, A.; Serrano, I.; Barco, R. Root Cause Analysis Based on Temporal Analysis of Metrics Toward Self-Organizing 5G Networks. IEEE Trans. Veh. Technol. 2017, 66, 2811–2824.

[4]. Szilagyi, P.; Novaczki, S. An Automatic Detection and Diagnosis Framework for Mobile Communication Systems. IEEE Trans. Netw. Serv. Manag. 2012, 9, 184–197.

[5]. Gómez-Andrades, A.; Muñoz, P.; Serrano, I.; Barco, R. Automatic Root Cause Analysis for LTE Networks Based on Unsupervised Techniques. IEEE Trans. Veh. Technol. 2016, 65, 2369–2386.

[6]. Wu, H.; Yin, L.; Chen, Y.; Li, Z.; Tang, Q. Elevator Fault Diagnosis Based on a Graph Attention Recurrent Network. Electronics 2025, 14, 2308. https://doi.org/10.3390/electronics14112308

[7]. Hu, X.; Liu, L.; Quan, Z.; Huang, J.; Liu, J. A Novel Integrated Fault Diagnosis Method Based on Digital Twins. Signals 2025, 6, 18. https://doi.org/10.3390/signals6020018

[8]. Montejano Leija, A.B.; Ruiz Beltrán, E.; Orozco Mora, J.L.; Valdés Valadez, J.O. Performance of Machine Learning Algorithms in Fault Diagnosis for Manufacturing Systems: A Comparative Analysis. Processes 2025, 13, 1624. https://doi.org/10.3390/pr13061624

[9]. Li, K.; Wu, B.; Xia, S.; Jia, X. A Graph Isomorphic Network with Attention Mechanism for Intelligent Fault Diagnosis of Axial Piston Pump. Appl. Sci. 2025, 15, 6586. https://doi.org/10.3390/app15126586

[10]. Wang, Y.; Wang, H.; Bai, R.; Shi, Y.; Chen, X.; Xu, Q. Enhanced Rolling Bearing Fault Diagnosis Using Multimodal Deep Learning and Singular Spectrum Analysis. Appl. Sci. 2025, 15, 4828. https://doi.org/10.3390/app15094828

[11]. Sangaiah, A.K.; Rezaei, S.; Javadpour, A.; Miri, F.; Zhang, W.; Wang, D. Automatic Fault Detection and Diagnosis in Cellular Networks and Beyond 5G: Intelligent Network Management. Algorithms 2022, 15, 432. https://doi.org/10.3390/a15110432

[12]. Zheng, L.; Jiang, Y.; Jiang, H.; Tang, C.; Jiao, W.; Shi, Z.; Rehman, A.U. Adaptive Dynamic Threshold Graph Neural Network: A Novel Deep Learning Framework for Cross-Condition Bearing Fault Diagnosis. Machines 2024, 12, 18. https://doi.org/10.3390/machines12010018

[13]. Dave, V.; Borade, H.; Agrawal, H.; Purohit, A.; Padia, N.; Vakharia, V. Deep Learning-Enhanced Small-Sample Bearing Fault Analysis Using Q-Transform and HOG Image Features in a GRU-XAI Framework. Machines 2024, 12, 373. https://doi.org/10.3390/machines12060373

[14]. Kaplan, H.; Tehrani, K.; Jamshidi, M. A Fault Diagnosis Design Based on Deep Learning Approach for Electric Vehicle Applications. Energies 2021, 14, 6599.k. https://doi.org/10.3390/en14206599

[15]. Meng, X.; Jing, B.; Wang, S.; Pan, J.; Huang, Y.; Jiao, X. Fault Knowledge Graph Construction and Platform Development for Aircraft PHM. Sensors 2024, 24, 231. https://doi.org/10.3390/s24010231

[16]. Barrera-Llanga, K.; Burriel-Valencia, J.; Sapena-Bañó, Á.; Martínez-Román, J. A Comparative Analysis of Deep Learning Convolutional Neural Network Architectures for Fault Diagnosis of Broken Rotor Bars in Induction Motors. Sensors 2023, 23, 8196. https://doi.org/10.3390/s23198196