http://dx.doi.org/10.21622/ACE.2025.05.1.1332

REAL-TIME MOBILE BROADBAND QUALITY OF SERVICE PREDICTION USING AI-DRIVEN CUSTOMER-CENTRIC APPROACH

Ayokunle A. Akinlabi[1,a*], Folasade M. Dahunsi[2,b], Jide J. Popoola[3,c] and Lawrence B. Okegbemi[4,d]

[1,2]Department of Computer Engineering, The Federal University of Technology Akure, Nigeria

[3]Department of Electrical and Electronics Engineering, The Federal University of Technology Akure, Nigeria

[4]Department of Mechanical Engineering, The Federal University of Technology Akure, Nigeria

Emails: {aaakinlabi@futa.edu.ng, fmdahunsi@futa.edu.ng, jjpopoola@futa.edu.ng, lawrencebolu@gmail.com}

Received on, 05 May 2025 - Accepted on, 26 May 2025 - Published on, 18 June 2025

ABSTRACT

Statistical methods employed in evaluating the quality of service (performance) of mobile broadband (MBB) networks face drawbacks relating to the accurate and reliable processing of the huge amounts of heterogenous real time traffic data generated from MBB networks. Since the traffic patterns experienced in MBB networks are largely complex, highly dynamic, and heterogenous in nature, statistical methods may not adjust adequately to the changing network conditions. This highlighted gap can be addressed by machine learning (ML), as it has been effectively used in the past to support the analysis and knowledge discovery of communication systems' traffic data through the identification of intricate and hidden patterns. This paper presents the application of ML techniques to predict MBB Quality of Service (QoS) in real-time, using a custom-built MBB performance application referred to as MBPerf that collects five (5) network metrics (DNS lookup, download and upload speeds, latency, signal strength), location information, and device characteristics across diverse network conditions in the South West region of Nigeria. The QoS modeling task was carried out using an MBPerf pre-processed dataset. Three (3) classification algorithms, including Random Forest (RF), Support Vector Machine (SVM), and Extreme Gradient Boosting (XGBoost), were trained using the MBPerf QoS dataset and then evaluated in order to determine the most effective model based on certain evaluation metrics – accuracy, precision, F1-Score and recall. Following hyperparameter tuning to improve the model's performance, the selected model was deployed in a real-world network environment to classify QoS into "Above Average," "Average," and "Below Average" categories. Mobile customers receive real-time notifications with actionable insights based on the predicted QoS class, empowering them to optimize their usage and troubleshoot issues. From the performance results obtained for the 3 ML models trained with the MBPerf dataset, SVM (95%) and XGBoost (97%) significantly outperformed RF (59%) in terms of accuracy. However, the performance difference between the SVM and XGBoost models is not significant. Interestingly, the three models showed great capability to accurately make predictions on the three QoS categories (classes) as depicted by the ROC-AUC and mlogloss curves.

http://dx.doi.org/10.21622/ACE.2025.05.1.1332

Lastly, the feature importance plot shows that QoS is the collective effect of service performance and not a function of QoS metrics only that determines the degree of satisfaction of a user of the service. This Artificial Intelligence (AI) powered system promotes a more transparent and efficient MBB experience for all stakeholders in Nigeria's fast evolving digital landscape.

Index words: *Cellular Network, Mobile Broadband, Quality of Service, Machine Learning, Crowdsourcing, Extreme Gradient Boosting, Random Forest, Support Vector Machine*

1. INTRODUCTION

Mobile broadband stands out as one of the most groundbreaking technologies of the 20th century and has continued to revolutionize global communications significantly in the current 21st century. MBB networks, which support a number of vital services essential to present-day society, are arguably now the cornerstone of the worldwide communications infrastructure [1]. As daily activities across education, health, business, entertainment, social life, and news sectors become increasingly bandwidth-intensive and bandwidth-hungry, mobile customers demand fast and highly responsive access wherever they go – whether at home, work, in cities or rural areas, in vehicles or on the move. As a result, their reliance on MBB networks continues to grow. One cannot overstate MBB's many benefits. In the true sense, the fourth generation (4G) technologies are the bedrock of MBB, capable of delivering more than 100 Mbps when customers are on the move. The fifth generation (5G) technologies, which have not yet been fully rolled out in developing markets, are expected to deliver enhanced MBB speeds reaching 20 Gbps, thereby extending 4G's Internet of Things (IoTs) capability and enabling mission-critical applications that demand ultra-high reliability and very low latency. The sixth generation (6G) system is anticipated to be in commercial use by 2030 [2]. The massive amounts of customer data collected continuously on a daily basis by MNOs from different sources, such as network traffic, social media, phone calls, Short Message Service (SMS), Internet activities, and mobile device interactions, provide useful information about usage patterns, service quality, network operations, and market trends. Machine learning offers algorithmic models that can intelligently analyze these data, uncover patterns, make predictions, and recommend actions without having to do explicit programming. The core of this research involves the development and deployment of an AI-powered system capable of accurately predicting MBB QoS on customers' smartphones, based on the historical and real-time data streams comprising five measured QoS metrics: phone, location, and network information. A MBB performance application (MBPerf), custom-built for the purpose of data collection, provided the dataset needed for the QoS prediction and classification tasks.

Since MBPerf data was unlabeled, a widely used unsupervised learning algorithm (k-means clustering) attempted to segment the data into k = 3 distinct groups, effectively uncovering basic patterns within the data [3]. The preprocessed k-means classified MBPerf dataset was used to train and test 3 ML algorithms (classifiers). These classifiers include RF, SVM, and XGBoost. The selection of the three classifiers was based on their collective ability to address the complex, non-linear, and noisy characteristics inherent in real-world network data. Existing studies have also revealed the efficiency of these classifiers in network-related modeling. The combination of the aforementioned strengths allows for robust modeling of the diverse feature interactions and the handling of the high dimensional data, which is likely to result in high prediction accuracy. Based on precision, F1-Score, and recall evaluation metrics, XGBoost was selected as the best performing model. Following hyperparameter tuning to optimize the model's Performance, XGBoost was deployed and integrated into MBPerf to classify the predicted QoS into "Above Average," "Average," and "Below Average" categories. The successful implementation of the QoS prediction system offers significant advantages to key stakeholders within the Nigerian telecommunications ecosystem. Importantly for mobile customers, the integration of the trained ML model into MBPerf provides real-time notifications regarding their predicted QoS class. This empowers users with actionable insights like those experiencing "Above Average" QoS can confidently engage in bandwidth-intensive activities, while users in the "Average" category are informed of typical performance levels. Crucially, users predicted to experience "Below Average" QoS receive notifications suggesting potential causes such as network congestion, weak signal, and actionable steps to mitigate these issues, such as moving to a different location, switching network modes, or contacting their MNOs for support. This transparency enhances user experience, reduces frustration, and grants informed usage of MBB services. This paper is organized as follows: section 2 discusses existing studies relating to network performance improvements and the integration of AI into cellular networks. Section 3 presents the proposed framework employed in mobile broadband quality of service prediction, while section 4 presents and analyzes the results of data validation, clustering techniques, and classifier evaluation. Finally, section 5 concludes the paper.

2. RELATED WORKS

With respect to network performance improvement, a great deal of studies have been carried out on integrating AI into communication networks. Wireline and wireless communication networks are constantly growing due to the growing number of devices that are linked to the Internet. The number of connected people, devices, and data transmission volumes on the Internet is growing at an exponential rate. As such, it is getting harder to control, manage, monitor, and predict traffic patterns in mobile broadband networks through manual techniques due to their vast scale and rapid growth. As a result, ML, which is capable of accurate traffic predictions, can be the key to automating network decisions to boost performance [4].

[5] Suggested a personalized approach for predicting Internet throughput for streaming applications using vectorization and clustering methods. Consequently, decision trees, Naye Bayes, and ARIMA models were trained and evaluated to determine which algorithmic model predicted throughput most accurately. Location coordinates, time of the day, user physical movement speed, and download speed were the features used for the prediction task. The time-series dataset was found to be a better fit for the ARIMA model, but it was not particularly accurate. The decision tree and vectorized inputs showed the best accuracy during the testing phase. The throughput predictor agent was developed using only a single 4G network dataset, which may limit the generalizability of the results to other network technologies or scenarios.

[6] Employed ML and Deep Learning (DL) algorithms, including RF, SVM, and Long Short-Term Memory (LSTM) to predict throughput in cellular networks with Channel Quality Indicator (CQI), cell load, throughput, and mobility mode as test parameters (features). The results of the study show that ML and DL can improve throughput prediction accuracy in cellular networks and that feature engineering and architectural components have a significant impact on prediction accuracy.

[7] Considered a Fuzzy Knowledge-Based (FKB) approach with Triangular Membership Function (TMF) for evaluating commonly used QoS indicators needed to model the MBB network performance provided by three (3) MNOs in the Niger Delta region. Data were collected using the software-based approach. Signal strength, packet loss, and upload and download speeds were the test parameters considered by the authors of this study. Results showed that the selected MNOs vary in QoS with respect to the test parameters compared. The research only considered just three (3) QoS metrics, and the number of days for data collection was too short – 21 days. These factors can limit the research since a high volume of observations and features in the training and test data will enhance the performance of the used model.

[8] Proposed a data-driven framework for detecting 4G cells with underlying network throughput problems with the help of ML techniques. The methodology employed by the authors is a combination of clustering models and deep neural networks (DNNs) that require only a small amount of expert-labeled data. The test parameters measured and fed to the models include CQI, Throughput, Reference Signal Received Power (RSRP), Reference Signal Received Quality (RSRQ), etc. The result showed that resolving problematic cells could boost the network's total throughput by 8%, based on data obtained from real 4G cells. A comparison of the proposed framework with SVM yielded unsatisfactory results, indicating that the DNN is a better choice for this problem. The proposed framework was tested on a single 4G network dataset; hence, it may limit the generalizability of the results to other networks.

[9] Proposed a customer churn prediction model designed to detect potentially dissatisfied

customers at an early stage and implement proactive retention strategies using classification and clustering techniques. The study used six ML models to predict customer churn: Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Decision Tree, Random Forest, and XGBoost. The test parameters considered include international plan, voice mail plan, number of voicemail messages, total day minutes, total day calls, total day charge, total evening minutes, total evening calls, total evening charge, total night minutes, total night calls, total night charge, etc. The results indicate that XGBoost and RF outperformed K-Nearest Neighbors, Support Vector Machines, and Decision Trees, achieving higher prediction accuracy across accuracy, precision, F1-score, and recall metrics.

[10] Presented a methodology for modeling and predicting the Quality of Experience (QoE) of mobile applications in WiFi or cellular networks. This study showed that gathering background information from smartphones and in-situ QoE ratings and applying these data to standard ML methods can create a precise predictive model for 'High' and 'Low' QoE. [10] were able to construct predictive QoE models with qualitative and quantitative data collected in a living laboratory. Their work also provided guidance to mobile application developers on how to design QoE-aware applications. ML has been identified as a proficient AI technology that helps in understanding and predicting customer churn, thereby allowing MNOs to anticipate churn problems by recognizing possible disgruntled consumers early on and taking proactive steps to keep them. Undoubtedly, this will result in enhanced retention tactics and a rise in profitability for the MNOs.

The study carried out by [11] uses a DL-based approach, specifically a hybrid CNN-LSTM model, to detect QoS anomalies. The UNSW-NB15 dataset used was preprocessed by handling missing or inconsistent values and selecting relevant features, including the QoS metrics such as bandwidth, latency, jitter, packet loss, and service availability. The study then trains the CNN, LSTM, and hybrid CNN-LSTM models using the preprocessed dataset and evaluates the models using metrics such as accuracy, precision, recall, F1-Score, and false positives. The hybrid CNN- LSTM model outperforms the individual CNN and LSTM models, achieving high accuracy and efficiency in detecting anomalies. The hybrid model outperforms the individual models with 98.67% accuracy, precision, recall, and F1-Score. This proves its anomaly detection resilience. Measurable results show the model improves network dependability, resource allocation, and user satisfaction. Potential limitations of the study include the need for large amounts of data to train the DL models and the complexity of implementing these models in real-time network environments.

[12] proposed a deep learning-based approach for QoS prediction to address the problems of data sparsity and cold-start inherent in the current QoS prediction approaches, such as collaborative filtering methods. The study also explored the impact of geographical characteristics of services/users and QoS rating on the prediction problem. The methodology involves combining a matrix factorization model based on a deep auto-encoder (DAE) and a clustering technique based on geographical characteristics to improve prediction effectiveness. In achieving the QoS prediction, the QoS data was clustered using a self-organizing map that incorporates the knowledge of geographical neighborhoods. This clustering step effectively handled the cold start problem. In addition, for each cluster, a DAE was trained so as to minimize the squared loss between the ground truth QoS and the predicted one. Next, the missing QoS of a new service is predicted using the trained DAE related to the closest cluster. The effectiveness and robustness of the proposed approach were evaluated using a comprehensive set of experiments based on a real-world web service QoS dataset. The experimental results showed that the method achieves a better prediction performance compared to several state-of-the-art methods.

The objective of the research conducted by [13] was to explore the application of artificial intelligence (AI) techniques, particularly ML, in predicting QoS on mobile networks. The main focus was to test the ability of AI models to predict several QoS parameters, such as throughput, latency, and packet loss. The methods used in this study include data collection from simulations of mobile networks generating a dataset containing 100,000 network traffic samples, data pre-processing, feature selection, and model evaluation using metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). The AI models used include K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Deep Learning (LSTM). The results of this study show that LSTM-based deep learning models have the lowest prediction error rate in estimating packet loss,

http://dx.doi.org/10.21622/ACE.2025.05.1.1332

followed by SVM and then KNN. The study also found that AI approaches, especially Deep Learning such as Long Short-Term Memory (LSTM), are able to predict QoS with high accuracy in a dynamic cellular network environment.

The research conducted by [19] addressed the critical problem of predicting QoS in cellular networks, a key factor in ensuring a seamless user experience in a world of ubiquitous computing and giant data networks. With the help of DL techniques, future complex lags based on time series data in three simulated drift scenarios created in OMNET++, Simu5G, Veins, and Sumo were predicted. The study employed the N-BEATS model – a deep learning architecture specialized in time-series forecasting that can identify unique patterns and trends to predict future values. The implementation was done with PyTorch, including data preparation, creation of N-BEATS model architecture, configuration of loss functions and optimizers, and training of the model for prediction. Evaluation metrics such as MSE, RMSE, and MAE were used for error estimation.

3. PROPOSED SCHEME FOR THE PREDICTION OF MOBILE BROADBAND QUALITY OF SERVICE

3.1. PHASES OF THE RESEARCH

This research is divided into two (2) phases. The flow diagram in Figure 1 presents the step-by-step process followed to achieve the research objectives.

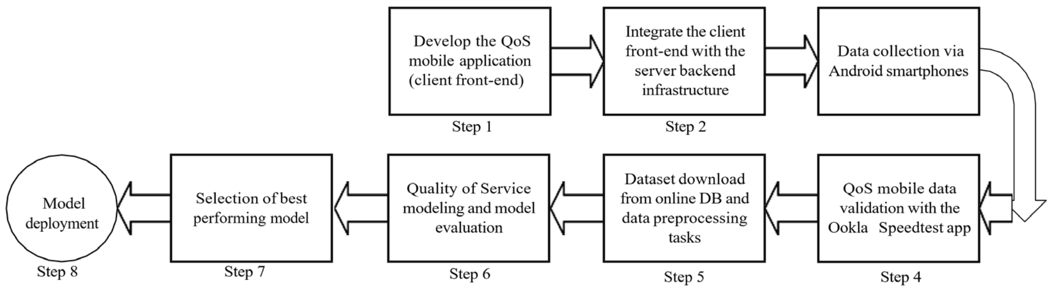

Fig 1: Flow Diagram Showing Step-by-Step Process to Achieve Research Objectives

3.1.1. PHASE ONE – QOS DATA COLLECTION AND VALIDATION

Step 1: Development of MBPerf, which when installed on the mobile devices of customers collected their relevant user-data. Additionally, MBPerf measured five (5) selected QoS metrics. MBPerf was developed for the main purpose of data collection.

Step 2: The client front-end system (MBPerf) was integrated with the server back-end DB (Google Firestore) for seamless data collection and storage.

Step 3: MBPerf was installed on five hundred (500) Android smartphones to collect MBB QoS data for a period of nine (9) months.

Step 4: The dataset collected with MBPerf was validated with the data obtained from the Ookla Speedtest app. The result of the data validation task is presented in section 4.1.

Step 5: The total collected data which amounted to 6500 measurement instances were downloaded from the online DB to the research Laptop (specification: Intel(R) Core (TM) i3-8130U CPU at 2.21 GHz, 8.00 GB RAM and x64-based processor).

The output of this phase is MBPerf QoS dataset needed for phase two. Some of the User Interfaces (UIs) of MBPerf QoS application (Figures 11a, 11b and 11c) and MBPerf's online dashboard (Figures 13 to 16) are shown as supplementary Figures.

http://dx.doi.org/10.21622/ACE.2025.05.1.1332

3.1.2. PHASE TWO – QUALITY OF SERVICE MODELLING AND PREDICTION

Steps 8-11: The QoS modeling task, which is described in detail in section 3.6 was carried out using the pre-processed dataset. Three (3) ML models – RF, SVM and XGBoost were trained with the QoS dataset and then evaluated in order to select the best performing model based on certain evaluation metrics. Following hyperparameter tuning to optimize the model's performance, the selected model was deployed in a real-world network environment.

3.2. QUALITY OF SERVICE MODELLING USING MACHINE LEARNING TECHNIQUES

The QoS prediction system was built with strong foundations in robustness, security, and scalability and relies on a simple, modern, and cloud-based architecture perfectly suited for the dynamics of real-time cellular networks. Robustness is inherent in the choice of Google Firebase as the cloud server backend infrastructure and specifically adopting Firestore as the DB. Firebase provides a fully managed, globally distributed infrastructure, minimizing downtime and ensuring high availability for continuous data collection from the growing customer (volunteer) base. Because the MBPerf mobile application and the online dashboard are based on the Flutter framework, the applications include solid error handling and data validation, which makes sure the data is safe and accurate, regardless of any challenging connection issues. With respect to security, the admin's dashboard login is managed by Firebase authentication and user data is protected, anonymized, and only accessed by authorized stakeholders. Data privacy is very important because we already have over 500 individuals sending in their data. Suffice to state that MBPerf developers place top priority on customers' personal information, which is why network performance data are collected anonymously. None of the customer's personally identifiable information is collected by MBPerf. The more reason volunteers are not asked to sign up or register to install or use the QoS application. Lastly, Firestore and its linked services, all within Firebase, were designed to scale seamlessly, hence, a developer does not have to manage complex infrastructures as user count and data keep rising. Both the Flutter front end and Flutter web dashboard are built for high performance, which keeps the experience smooth as the volunteer base grows and more stakeholders are onboarded. The step-by-step process followed to achieve the QoS modeling and prediction is illustrated in Figure 2 and elaborated in sub-sections 3.2.1 to 3.2.10.

3.2.1. MBPERF DATASET

The first requirement for building a machine learning model is the dataset. A dataset is a collection of data in a specific format for a specific problem. The volunteers' data records, including their location, network, mobile device information, and measurement results stored on Google Firestore, were downloaded as a single .csv file and then mounted on Google Drive. The coding environment utilized for the various ML tasks was Google Colab, a cloud-based service that requires no setup and offers free access to open-source tools such as Jupyter Notebook and Python. Furthermore, it provides computing resources like GPUs and TPUs, making it particularly suitable for ML, data science, and educational purposes [14]. The drive module imported unto the google.colab package, allows access to the QoS dataset already mounted on the Google drive. NumPy, Pandas, Matplotlib, Seaborn, Sklearn, and XGBoost were the necessary libraries needed to be imported for data preprocessing, data analysis, visualization, and machine learning tasks. The QoS dataset consists of 6500 measurement instances (rows) and 20 (features) columns. The dataset is not labeled, i.e., no record of QoS classification, as this is the target variable. The dataset consists of both numerical and categorical variables.

3.2.2. DATASET PREPROCESSING

Raw data must undergo preprocessing and feature engineering to ensure its suitability for ML models. Data cleaning, data encoding, frequency encoding, label encoding and one-hot encoding are the various data preprocessing tasks carried out on the raw QoS dataset. The QoS dataset comprises a mix of categorical and numerical data types. Since many ML models are incompatible with categorical data, hence, it was converted into numerical format before

Volume 5, Issue 1, June 2025- ISSN 2735-5985
http://dx.doi.org/10.21622/ACE.2025.05.1.1332

being utilized in the selected models.

3.2.3. FEATURE ENGINEERING

Feature engineering is a crucial step in preparing data for ML. It is the process of creating appropriate features from pre-existing features, thereby leading to improved predictive performance [15]. Domain expertise/knowledge of this research area and iterative trial and error and model evaluation were followed in performing the following feature engineering tasks, which include feature selection/extraction, feature expansion, feature scaling, and Principal Component Analysis (PCA). Standardization was chosen as the feature scaling method, ensuring that all features in the training and test sets are kept on the same scale. This process also accelerates model training. The mathematical formula for standardization, according to [9], is:

$$x' = \frac{[x-\bar{x}]}{\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})^2}} \qquad (1)$$

where x is original feature, () is the scaled feature, () is the mean of x and the denominator is the standard deviation of x." below the equation, i.e.,

PCA was useful for several purposes, including noise reduction, feature extraction, and dimensionality reduction of the QoS data before applying the selected ML algorithms, which led to improved efficiency and reduced overfitting.

3.2.4. DATA SPLITTING

The preprocessed QoS dataset was divided equally into two parts (50:50). The first half (split A) was used to train the k-means clustering algorithm, and the resulting model was then applied to predict the target variables (QoS categories) for the second half (split B). Split B was now treated as a completey labeled dataset and further divided in an 80:20 ratio, serving as the ground truth for training and testing the three selected classification models – RF, SVM, and XGBoost. The afore-explained data splitting strategy was adopted to prevent any data leakage, which can result in poor model generalization or creating a biased model, by ensuring that first, the clustering algorithm is trained on a separate dataset from what is used in the classification models. Second, the classification models are evaluated on unseen data, and lastly, no information from the test data influences the training process.

http://dx.doi.org/10.21622/ACE.2025.05.1.1332

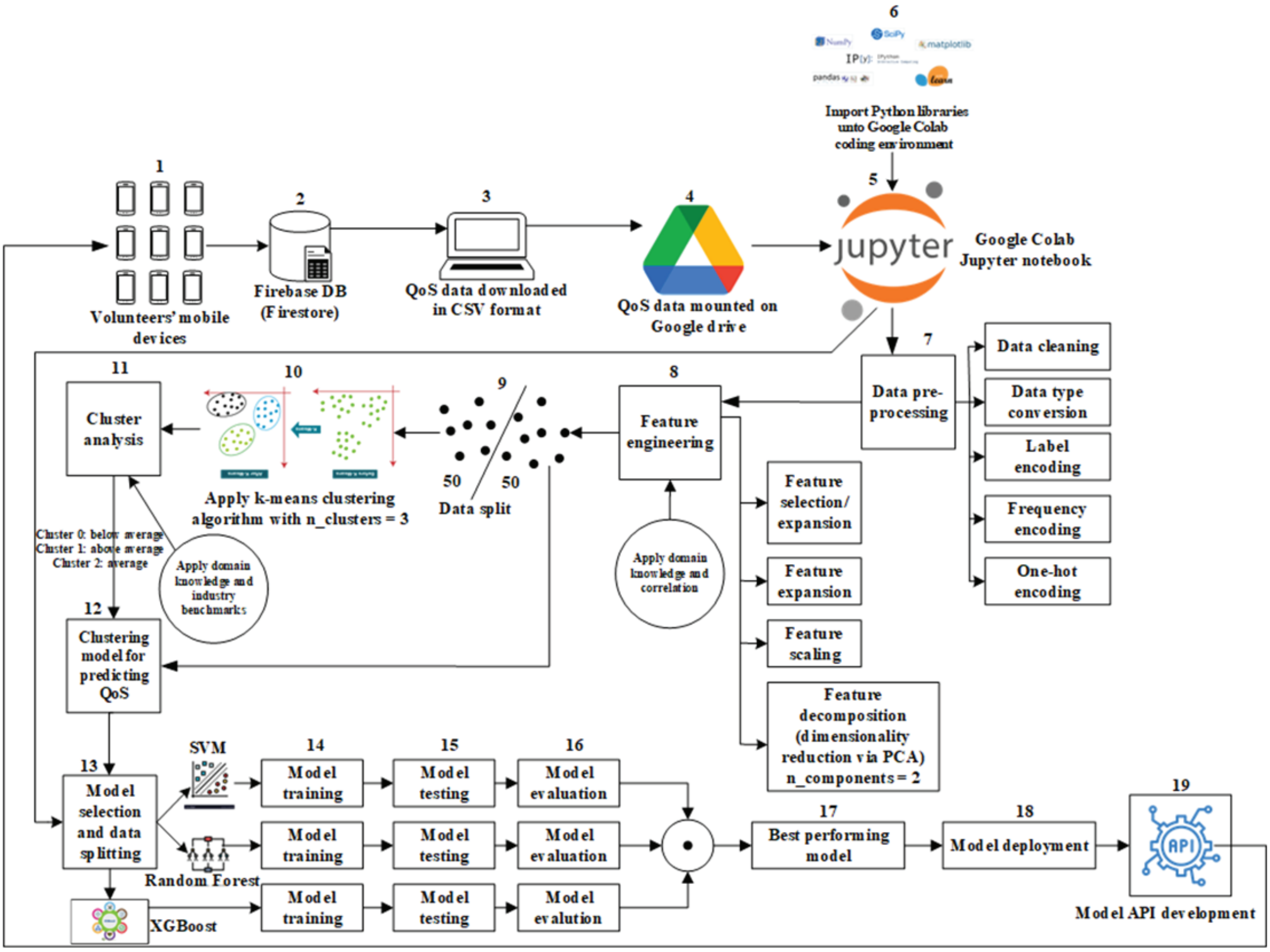

Fig. 2: Step-by-Step Process to Achieve the QoS Modelling

3.2.5. K-MEANS CLUSTERING

Since the QoS dataset was unlabeled, a popular unsupervised learning algorithm known as k- means clustering attempted to divide the data into k = 3 discrete groups and is effective at uncovering basic data patterns [3]. The k-means clustering algorithm worked by splitting the data into 3 clusters representing the three yet unknown QoS classifications/categories. The first step to achieve the 3 cluster classes was to select a centroid for each k cluster. The remaining data points on the scatterplot are then assigned to the closest centroid by measuring the Euclidean distance given by the formula [3]:

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots (x_n - y_n)^2} \qquad (2)$$

where x and y are two data points, and n is the number of features. The centroid of the ith cluster is calculated as [3]:

$$centroid_i = \left(\frac{1}{N_i}\right) * \sum x_j \qquad (3)$$

where $centroid_i$ is the centroid of the ith cluster, N_i is the number of data points in the ith cluster, and x_j are the data points assigned to the ith cluster.

Volume 5, Issue 1, June 2025- ISSN 2735-5985
http://dx.doi.org/10.21622/ACE.2025.05.1.1332

3.2.6. CLUSTER ANALYSIS

The analysis of the outputs of the k-means clustering algorithm are 3 QoS classes/categories designated as 0, 1 and 2. After thorough analysis of the 3 clusters, it was concluded that cluster 0 is the 'average class', cluster 1 is the 'above average class' and cluster 2 is the 'below average class'.

3.2.7. SELECTION, TRAINING, AND TESTING OF ML MODELS

The preprocessed k-means classified QoS dataset was used to train and test 3 ML algorithms (classifiers). As explained below, the classifiers include RF, SVM, and XGBoost. These three classifiers were selected because of their collective ability to address the complex, non-linear, and noisy characteristics inherent in real-world network data. Existing studies have also revealed the efficiency of these classifiers in network-related modeling. The combination of the highlighted strengths allows for robust modeling of the diverse feature interactions and the handling of the high dimensional data, which is likely to result in high prediction accuracy.

i. Random Forests: RFs are an ensemble classifier which fits a large number of decision trees to a dataset, and then combines the predictions from all the trees [3]. Each tree in a RF is a weak learner. However, the RF is a strong learner combining multiple predictions into one aggregate prediction thereby resulting in a more accurate forecast.

ii. Support Vector Machine: SVM worked by constructing an optima hyperplane in a multidimensional space where the hyperplane separates the collected QoS sample data into three (3) different classes. Consider the QoS training sample consisting of N patterns $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ where x is the feature vector, and target $y_i \in \{-1, 0, +1\}$ with corresponding multi-class labels. The SVM parameters are determined by maximizing the margin hyperplane [3]:

$$\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i . \alpha_j) \qquad (4)$$

Subject to the constraints [3]:

$$\sum_{i=1}^{N} \alpha_i \text{ and } 0 \leq \alpha_i \leq C \qquad (5)$$

where $K(x_i, x_j)$ is the kernel function used to map the data from the input space to the feature space and item C is a cost/slack parameter.

iii. XGBoost: is a decision tree ensemble based on gradient boosting, designed to be highly scalable and combines multiple weak learners (typically decision trees) to create a strong predictive model [16]. XGBoost aims to minimize a loss function, which is typically a combination of a training loss and a regularization term:

$$Objective = Loss + Regularization \qquad (6)$$

Table 1 shows the details of the hyperparameters tuned in order to achieve the three predicted QoS classes.

http://dx.doi.org/10.21622/ACE.2025.05.1.1332

Table 1: ML Models Hyperparameters

ML Model	Hyperparameter	Value
k-means clustering	n_clusters	3
	random_state	0
	n_init	auto
	max_iter	300
	tol	0.0001
	random_state	0
RF classifier	n_estimators	1000
	criterion	log_loss
	max_depth	6
	random_state	42
SVM classifier	kernel	linear
	probability	TRUE
	random_state	42
XGBoost classifier	objective	multi: softmax
	num_class	3
	missing	1
	gamma	0
	learning_rate	0.1
	max_depth	3
	reg_lambda	1
	subsample	1
	colsample_bytree	1
	early_stopping_rounds	10
	eval_metric	['merror,' 'mlogloss']
	seed	42

3.2.8. MODELS EVALUATION

To assess the effectiveness of the chosen ML models, selecting the appropriate evaluation metrics is crucial. Using an incorrect metric can lead to poor real-world performance of the ML model. Commonly used evaluation metrics include:

iv. Accuracy: This is used to deduce which ML model has the best capacity to recognize relationships and patterns between variables in the training dataset. It can be calculated by dividing the total number of correct predictions by the total number of predictions and then multiplying by 100 [9].

$$A = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \qquad (7)$$

where True Positive (TP) denotes that the actual value is true and the model predicts true; False Positive (FP) means the actual value is false and the model predicts true; True Negative (TN) means the actual value is false and the model predicts false and False Negative (FN) means the actual value is true, and the model predicts false.

v. Precision: This is the ratio of true positives and total positives predicted. The formula for Precision is given as [9]:

$$P = \frac{TP}{TP+FP} \quad (8)$$

vi. Recall/Sensitivity/Hit-Rate: The recall metric focuses on type-II errors, i.e., FN. However, it cannot measure the existence of type-I errors, which are false positives. It is calculated as [9]:

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \quad (9)$$

vii. F1-score: This is the combination of precision and recall. The F1- score is the harmonic mean of Precision and Recall. The formula of the two essentially is [9]:

$$\mathrm{F1score} = \frac{\mathrm{Precision}}{\mathrm{Precision+Recall}} \quad (10)$$

3.2.9. BEST PERFORMING MODEL

Comparing the results from the evaluation metrics, the best-performing model/classifier, i.e., XGBoost, out of the three (3) previously trained models, was picked for deployment and integration with MBPerf mobile app.

3.2.10. MODEL DEPLOYMENT

Deploying the best-performing model to a production environment, i.e., integrating the ML model with MBPerf, involves the following steps: first, the trained model is saved via Pickle library based on Python, and a model serving platform (Render) is chosen. Second, the Flask web framework was used to develop a RESTful API that accepts MBPerf data, processes it through the ML model and returns the predicted QoS classes and notifications. Third, Render was used to deploy the serialized model. Also, the API was connected to it (Render) so that it could receive the required data, process it through the ML model, and return predictions. Fourth, the developed API is interfaced with MBPerf, allowing it to send measurement data to the API and then receive predicted QoS values and notifications. A notification system was implemented to alert users of the predicted QoS values, implemented using the Firebase Cloud Messaging (FCM) platform. The UI of the MBPerf QoS application, after being integrated with the ML classifier, gives notifications on the predicted QoS, as shown in supplementary Figure 12.

4. RESULTS AND DISCUSSION

In this section, the graphical results for the data validation, k-means clustering and models' performances are presented.

4.1. MBPERF DATA VALIDATION

MBPerf has to be validated to ensure that the accuracy, and reliability of its dataset meets the NCC/FCC standard. Measurement results from the Ookla Speedtest app, a widely recognized and reliable benchmark, were used to validate MBPerf measurement results. Ookla Speedtest is recognized by the Federal Communications Commission (FCC) in the US, the International Telecommunications Union (ITU), and several ISPs and Telcos [17]. For seven

http://dx.doi.org/10.21622/ACE.2025.05.1.1332

(7) days, bi-hourly measurements of the download speed, upload speed, and latency were collected via MBPerf and Ookla Speedtest apps for each of the three MNOs under evaluation. Correlation analysis was carried out on the measurement results and presented in Table 2. The correlation analysis results reveal strong positive correlations between MBPerf and Ookla Speedtest measurements. For download speed, the correlation coefficient, r = 0.856; for upload speed, r = 0.884, while for latency metric, r = 0.755. These results show that MBPerf measurements are significant and, to a very large extent, accurate and reliable.

Table 2: Correlation Results between MBPerf and Ookla Speedtest Measurements

	MBPerf Download (Mbps)	*Ookla Download (Mbps)*	*MBPerf Upload (Mbps)*	*Ookla Upload (Mbps)*	*MBPerf Latency (ms)*	*Ookla Latency (ms)*
MBPerf Download (Mbps)	1					
Ookla Download (Mbps)	**0.856**	1				
MBPerf Upload (Mbps)	0.667	0.595	1			
Ookla Upload (Mbps)	0.775	0.723	**0.884**	1		
MBPerf Latency (ms)	-0.600	-0.621	-0.691	-0.773	1	
Ookla Latency (ms)	-0.572	-0.607	-0.603	-0.697	**0.755**	1

4.2. CLUSTERING RESULT

The result of the k-means clustering is shown in Figure 3, where X denotes the centroids picked by Python's k-means estimator with two principal components ($n\_clusters$ = 2) and three clusters (k = 3) representing the three classified QoS categories: average, above average and below average. To effectively map the k-means clusters to the three QoS classes (Figure 4), clusters' centroids were analyzed. In addition, data distributions across relevant QoS metrics, taking into cognizance the industry benchmarks, were also analyzed to identify clusters representing each of the QoS classes. The mapping was validated using domain knowledge and user feedback.

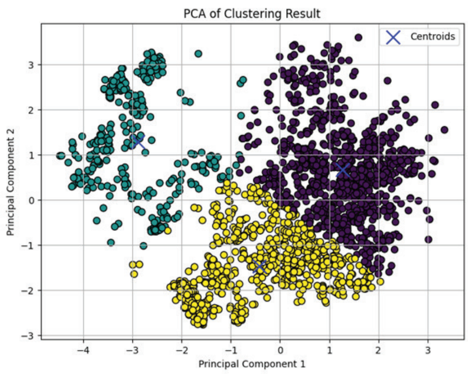

Fig. 3: Scatter Plot Showing Clustered QoS Data

http://dx.doi.org/10.21622/ACE.2025.05.1.1332

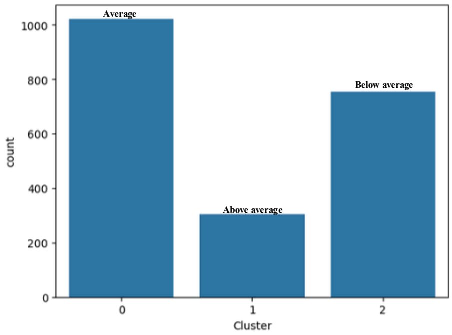

Fig. 4: Count Plotted against Clusters

4.3. RANDOM FOREST

The confusion matrix of Figure 5 shows that the RF classifier performed exceptionally well in identifying the average QoS class (256 correct classifications) but is challenged in distinguishing the above-average and below-average categories with significant misclassifications into the above-average and below-average QoS classes, particularly the average class (98 misclassifications). As shown in the performance comparison results of the ML models in Figure 9, RF achieved a balanced accuracy of 59%, micro precision of 74%, micro recall of 71%, and a micro F1-score of 68%. The high ROC-AUC scores (Figure 6) reveal that RF has the capability to accurately make predictions about the three QoS categories (classes) but is not as accurate as SVM and XGBoost.

4.4. SUPPORT VECTOR MACHINE

The confusion matrix in Figure 7 shows that the SVM classifier performed greatly in identifying all the QoS classes, with only four misclassifications in the average and above-average QoS classes and 15 misclassifications in the below-average QoS class. As shown in the performance comparison results of the ML models in Figure 10, SVM achieved a balanced accuracy of 95%, micro precision of 96%, micro recall of 95%, and a micro F1-score of 96%. These performance results, reveal that SVM has excellent capability to accurately make predictions about the three QoS categories (classes) better than the RF model.

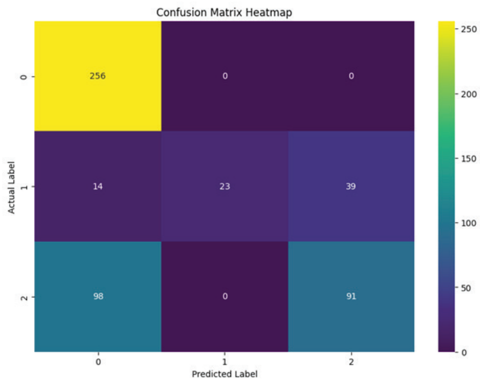

Fig. 5: Random Forest Confusion Matrix Heatmap

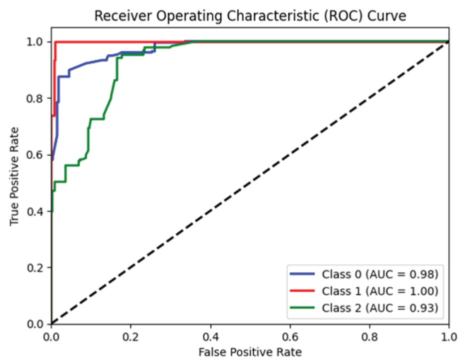

Fig. 6: Random Forest ROC Curve Matrix Heatmap

http://dx.doi.org/10.21622/ACE.2025.05.1.1332

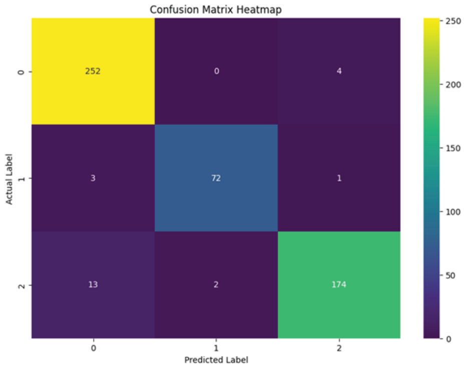

Fig. 7: SVM Confusion Matrix Heatmap

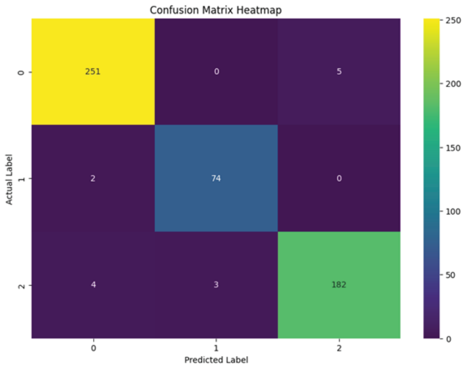

Fig. 8: XGBoost Confusion Matrix Heatmap

4.5. XGBOOST

The confusion matrix in Figure 8 shows that the XGBoost classifier, just like SVM, also performed greatly in identifying all the QoS classes with only five misclassifications in the average QoS class, two misclassifications in the above average class, and seven misclassifications in the below average QoS class. The performance comparison results of the ML models in Figure 9 reveal that XGBoost achieved a balanced accuracy of 97%, micro precision of 97%, micro recall of 96%, and a micro F1-score of 96%. The mlogloss curve presented in Figure 10 indicates a very good model performance. This is true because the curve is seen to decrease as the number of iterations increases, indicating that the model is improving. In summary, the highlighted results show that XGBoost, just like SVM, possesses excellent capability in predicting the three QoS classes. The performance difference between SVM and XGBoost is not significant.

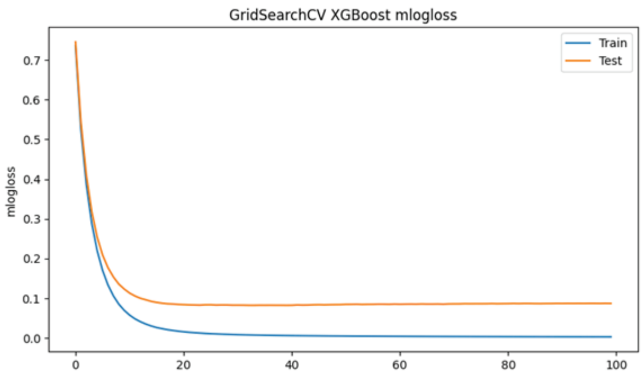

Fig. 9: XGBoost mlogloss Curve

Volume 5, Issue 1, June 2025- ISSN 2735-5985
http://dx.doi.org/10.21622/ACE.2025.05.1.1332

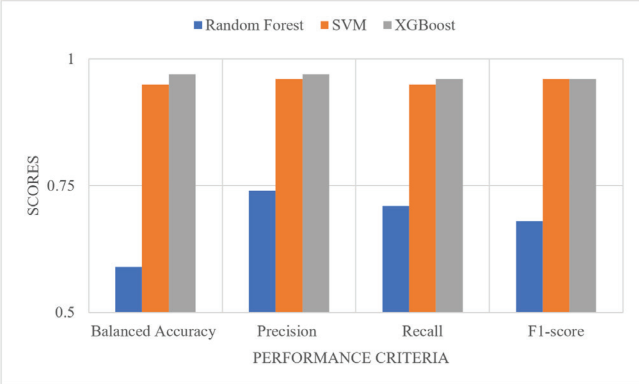

Fig. 10: performance Comparison of RF, SVM, and XGBoost ML Models

The performance comparison of the three (3) ML classifiers is presented in Table 3.

Table 3: Performance Comparison RF, SVM, and XGBoost classifiers

Model	Evaluation Metrics			
	Balanced Accuracy	Precision	Recall	F1-score
RF	0.59	0.74	0.71	0.68
SVM	0.95	0.96	0.95	0.96
XGBoost	0.97	0.97	0.96	0.96

4.6. FEATURE IMPORTANCE

One way to explain a model's behavior is to use feature importance, which measures the marginal contribution of each feature to a model's decisions. Each feature is assigned a score indicating its relative importance in the model, where higher scores indicate greater importance. Figure 11 shows that the OS version, device model, location, operator name, manufacturer, and brand name are the five most prominent features for predicting QoS in this research. A critical look at these features shows that they are the independent variables that are needed to explain the dependent variables, which are the QoS metrics. This important finding conforms with the ITU- T definition of QoS from the end user's perspective that: "QoS is the collective effect of service performance (not measured based on QoS metrics only) which determines the degree of satisfaction of a user of the service [18].

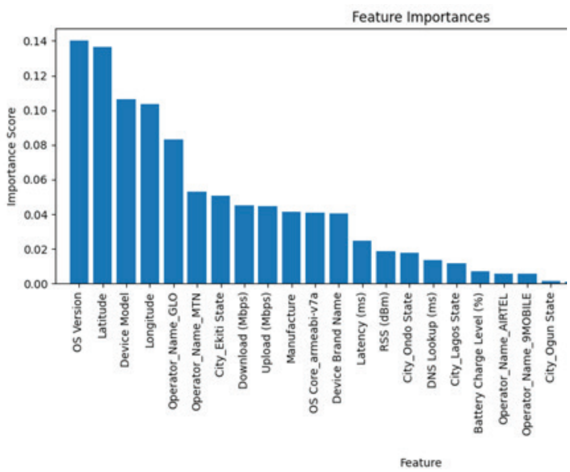

Fig. 11: Feature Importance Bar Chart

http://dx.doi.org/10.21622/ACE.2025.05.1.1332

5. CONCLUSION

This research presents a real-time framework for predicting and classifying MBB QoS, utilizing a customer-centric and AI-driven approach to bridge the information gap between mobile customers and MNOs, as well as empower customers to evaluate the actual quality of MBB services provided by MNOs in Nigeria. By employing a host and crowdsourced based methodology for collecting real-time QoS metrics data directly from a growing network of volunteer devices, the research has demonstrated the possibility of employing machine learning techniques in classifying network performance into "Above Average," "Average," and "Below Average" classes. This QoS system not only provides accurate predictive insights but also provides mobile customers with real-time notifications tailored to their current network conditions. The developed system benefits all stakeholders in Nigeria's telecom industry. The regulator can gain valuable insights into how well networks function in several geographical areas and among different MNOs. These insights will aid evidence-based policymaking. MNOs can be compared, and services are offered in an equitable way, which makes the industry more transparent and accountable. For the MNOs, this system allows for the early identification of congestion hotspots, optimization of resource allocation, and targeted infrastructure investments, leading to improved operational efficiency and customer loyalty. Most importantly, mobile customers are offered more than just the numerical results of Internet performance; they also receive a personalized network QoS prediction that help them improve their network activities and easily manage any problems. Moving forward, as the volunteer base keeps increasing and thousands of measurement instances are collected, attention will now shift to even more advanced prediction techniques. Future works will adopt Deep Learning models, including Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN). Harnessing the power of deep learning enables the discovery of more intricate patterns with better predictive accuracy for dynamic network fluctuations. Out of the three ML models – "employed in this study for the QoS classification task;" experimental results indicate that XGBoost achieved the highest performance, followed by SVM and then RF.

COMPETING INTERESTS

The authors declare that no competing interests exist.

SUPPLEMENTARY FIGURES

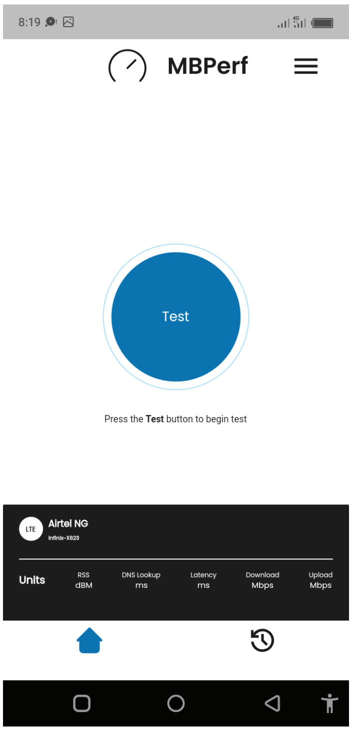

Fig. 11a: MBPerf Home Page

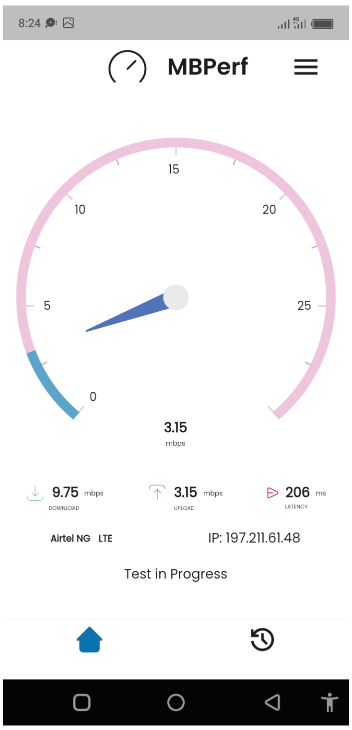

Fig. 11b: MBPerf Test Page

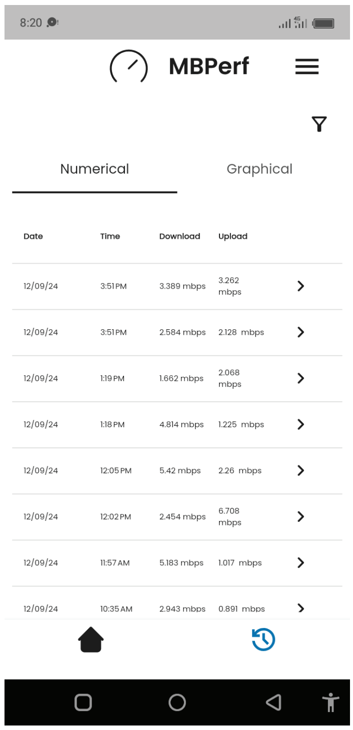

Fig. 11c: MBPerf Results Page

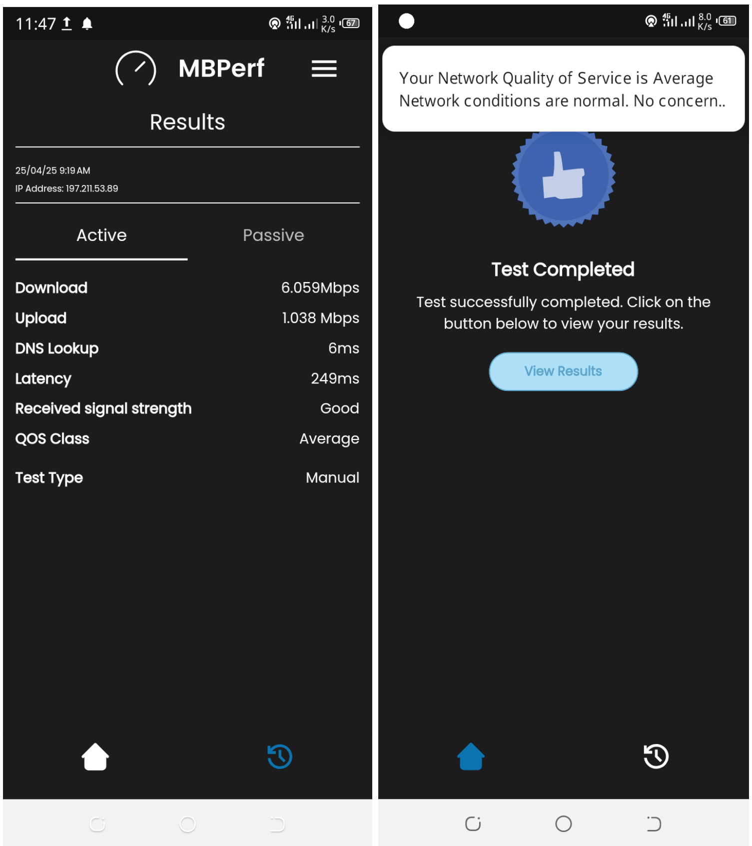

Fig. 12: MBPerf QoS Prediction and Notification

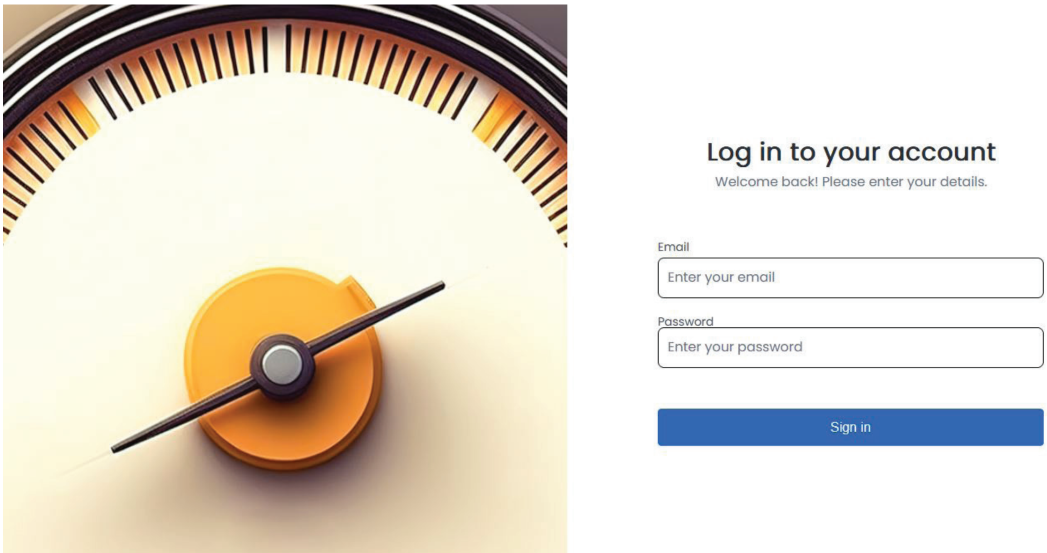

Fig. 13: Login Page

http://dx.doi.org/10.21622/ACE.2025.05.1.1332

Fig. 14: Dashboard Page

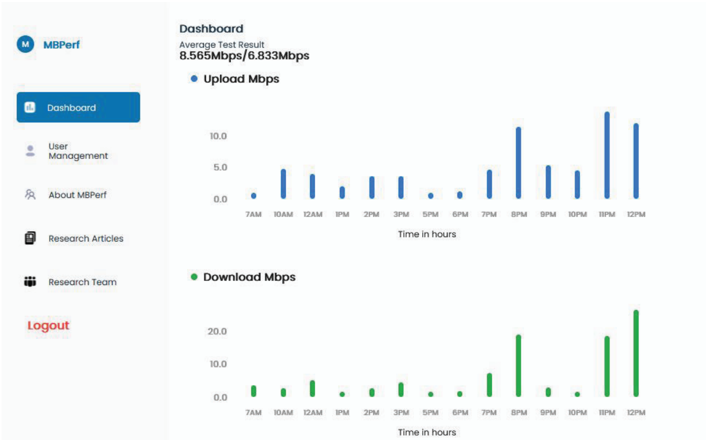

Fig. 15: User Management Page

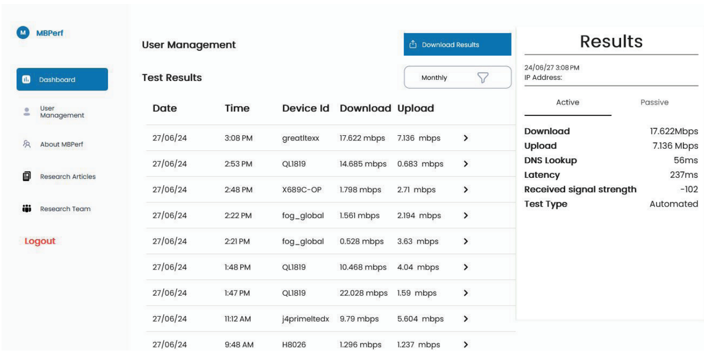

Fig. 16: User Management Page

REFERENCES

[1] A. A. El-Saleh, M. A. Al Jahdhami, A. Alhammadi, Z. A. Shamsan, I. Shayea, and W. H. Hassan, "Measurements and Analyses of 4G/5G Mobile Broadband Networks: An Overview and a Case Study," Wireless Communications and Mobile Computing (Hindawi), vol. 2023, pp. 1–24, Apr. 2023, doi: 10.1155/2023/6205689.

[2] H. Remmert, "When Is 6G Coming, and What Does It Mean for 5G and 4G LTE?" Accessed: May 28, 2024. [Online]. Available: https://www.digi.com/blog/post/when-is-6g-coming-what-does-it-mean-for-5g-4g

[3] O. Theobald, Machine Learning for Absolute Beginners, Second. 2017.

[4] A. Chen, J. Law, and M. Aibin, "A Survey on Traffic Prediction Techniques Using Artificial Intelligence for Communication Networks," Telecom, pp. 518–535, Dec. 2021, doi: 10.3390/telecom2040029.

[5] H. Jha and V. Vijayarajan, "Mobile Internet Throughput Prediction using Machine Learning Techniques," in Proceedings of the International Conference on Smart Electronics and Communication (ICOSEC 2020), IEEE Xplore, 2020, pp. 253–257. doi: 10.1109/ICOSEC49089.2020.9215436.

[6] J. Riihijärvi and P. Mähönen, "Machine Learning for Performance Prediction in Mobile Cellular Networks," IEEE Computational Intelligence Magazine, Jan. 2018, doi: 10.1109/MCI.2017.2773824.

[7] I. J. Umoren and S. J. Inyang, "Methodical Performance Modelling of Mobile Broadband Networks with Soft Computing Model," International Journal of Computer Applications, vol. 174, no. 25, pp. 7–21, 2021.

[8] L. Alho, A. Burian, J. Helenius, and J. Pajarinen, "Machine Learning Based Mobile Network Throughput Classification," in 2021 IEEE Wireless Communications and Networking Conference (WCNC), Nanjing, China: IEEE Xplore, 2021, p. 6. doi: 10.1109/WCNC49053.2021.9417365.

[9] G. Sam, P. Asuquo, and B. Stephen, "Customer Churn Prediction using Machine Learning Models," JERR, vol. 26, no. 2, pp. 181–193, Feb. 2024, doi: 10.9734/jerr/2024/v26i21081.

[10] A. De Masi and K. Wac, "Towards accurate models for predicting smartphone applications' QoE with data from a living lab study," Quality and User Experience, Springer Nature, vol. 5, no. 10, pp. 1–18, 2020, doi: https://doi.org/10.1007/s41233-020-00039-w.

[11] M. Ghuge, N. Ranjan, R. A. Mahajan, P. A. Upadhye, S. T. Shirkande, and D. Bhamare, "Deep Learning Driven QoS Anomaly Detection for Network Performance Optimization," J. Electrical Systems, vol. 19, no. 2, pp. 97–104, 2023.

[12] M. I. Smahi, f. Hadjila, C. Tibermacine, and A. Benamar, "A deep learning approach for collaborative prediction of Web Service QoS," SOCA, vol. 15, pp. 5–20, 2021, doi: 10.1007/s11761-020-00304-y.

[13] D. Rahmayanti, "Predicting Quality of Service on Cellular Networks Using Artificial Intelligence," Journal of Engineering, Electrical and Informatics, vol. 5, no. 2, pp. 28–35, 25AD, doi: https://doi.org/10.55606/jeei.v5i2.3901.

[14] Google, "Google Colaboratory," Google colab. Accessed: Nov. 24, 2024. [Online]. Available: https://colab.google/

[15] H. Samulowitz, U. Khurana, and S. D. Turaga, "Learning Feature Engineering for Classification," in Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17), Aug. 2017, pp. 2529–2534. doi: 10.24963/ijcai.2017/352.

[16] A. Csorg, G. Martınez-Munoz, and C. Bentejac, "A Comparative Analysis of XGBoost," Feb. 17, 2020, Spain. doi: 10.48550/arXiv.1911.01914.

[17] Ookla, "Speedtest," Speedtest. Accessed: Sep. 23, 2024. [Online]. Available: https://www.speedtest.net/about

[18] ITU-T Study Group, "Communications quality of service: A framework and definitions," Series G: Transmission Systems and Media, Digital Systems and Networks. ITU-T G-Series Recommendations (G.1000), 2001.

[19] Arnab, A.A., Shuvro, A.A., Ma, K.F., and Leung, H. (2023). A Deep Learning Approach for a QoS Prediction System in Cellular Networks. 2023 IEEE 9th World Forum on Internet of Things (WF-IoT), 1-6.