FAULT IMPACT ANALYSIS: TOWARDS SERVICE-ORIENTED NETWORK OPERATION & MAINTENANCE

Julius Maina

The author is a research student in data analytics at KCA University in Kenya

September 10th, 2023

Abstract *– This report introduces an innovative solution addressing fault management in telecom O&M, focusing on Radio Access Network (RAN) fault analysis. The conventional rule-based approach lacks accuracy in assessing the impact of faults on network Key Performance Indicators (KPIs), hindering effective resource allocation. Leveraging machine learning and AI, this solution predicts fault impacts on RAN KPIs, enabling prioritization based on actual effects. However, challenges arise due to uncertainties in dynamic RAN behavior, complex network topology, diverse user segments, and non-RAN faults. By bridging this gap, this approach elevates fault management to a service-centric paradigm, enhancing network stability and resource optimization.*

Keywords – Access Success Rate, Block Error Rate (BLER), Channel Quality Indicator (CQI), Data Rate, Fault Duration, Modulation and Coding Scheme (MCS), Network Element (NE), Radio Access Network (RAN), Relation, Resource Utilization Rate, Time Advanced (TA).

1. INTRODUCTION

In the world of telecom Operation and Maintenance (O&M), fixing network problems is key to keeping things running smoothly. One big part of this is analyzing faults, which means figuring out what went wrong.

In today's complex network landscape, engineers confront a multitude of daily challenges. Though they typically rely on established rules to address these issues, these rules often overlook the true extent of a problem's impact on the network's performance. This oversight can result in reduced network reliability and hindered resource allocation. Acknowledging the importance of thoroughly assessing these challenges' effects, we can enhance the ability to allocate resources effectively and elevate network dependability, leading to a higher standard of quality in the network operations.

This report highlights the potential of combining predictions from various Machine Learning algorithms using ensemble techniques to enhance fault impact prediction on a network. The objective is to forecast the effects of faults on data rate changes, assisting fault engineers in prioritizing issues that may result in data rate reduction and, consequently, prevent customer churn. Additionally, a brief evaluation of the findings will be presented along with potential improvements for the current solution.

2. DATASET

A dataset containing RAN KPI data from over 100 5G and 4G NEs was provided to enable the training of the machine-learning models. This dataset included crucial information about each NE's performance, encompassing six key RAN KPIs related to data rate feature. These parameters were recorded on an hourly basis, forming a foundational basis for the analysis. The KPIs are explained briefly below:

- ☐ Access Success Rate-a measure of the success rate of service requests made by users.
- ☐ Resource Utilization Rate-indicates the percentage of network resources being utilized during the hour.
- ☐ TA-a parameter related to the timing advance used in wireless communications.
- ☐ BLER-represents the percentage of blocks or packets with errors in the transmission.
- ☐ CQI-An indicator of the quality of the wireless communication channel.
- ☐ MCS -refers to the specific modulation and coding scheme used in data transmission.

These KPIs impacted the data rate by either increasing it from one hour to another or decreasing it. This change formed the target variable.

Other features on top of the KPIs were also provided:

- ☐ Fault Duration-the duration of a fault in seconds that occurred during the hour.
- ☐ Relation-the proximity or association of the

NE to the fault that occurred (refer to the NE's distance from the location of the fault).

- ☐ NE ID-a unique identifier for each NE, used to distinguish between different network elements.
- ☐ Hourly Timestamp-the timestamp of the data, indicating the specific hour when the measurements were taken.

The train dataset comprised of 7,256 train files equivalent to the same number of NEs and the test had 1932 NEs. Each of the files was organized in such a way that it constituted data rate and other features collected before the fault occurs, where fault duration is zero and then fault duration and relation in the first hour during which the fault appears.

The train datasets were merged into a single dataset, resulting in 908,922 rows and 11 columns. Similarly, the test datasets were consolidated to create a test dataset comprising 175,150 rows and 11 columns.

2.1 Data Preprocessing

Once the datasets were consolidated, the first challenge encountered was the pre- processing data phase. This is a very delicate phase since the way the features are prepared will have a big impact when training the model. The order of the preprocessing steps particularly holds great significance; altering their sequence could generate varying outcomes.

Starting with a temporary data integration, the training and validation datasets were merged to create a cohesive dataset with 1,084,072 rows for consistent preprocessing. A new feature called 'predict row' was introduced. If the fault duration exceeded zero for the test rows only, it was marked as one; otherwise, zero. This distinction aided in identifying validation rows that required prediction in the unseen dataset.

The target variable data rate trend at time t+1 was subsequently created. If the data rate of the previous non-fault row was higher than the data rate of the fault row, the label was set to one; otherwise, it was set to zero.

Following the creation of the target variable, the next step involved data transformation. During this phase, a data shift operation was implemented to leverage the KPIs from the previous day in order to make predictions for the current hour. In other words, the purpose was to use previous hour KPIs to predict current hour data rate change.

Next, an additional feature called 'fault' was introduced, generated based on the fault duration attribute. Specifically, instances where the fault duration exceeded 0 were assigned a label of 1, indicating the presence of a fault. This step facilitated the capture and differentiation of instances with faults from those without faults in the dataset.

Examination of the feature correlations within the datasets revealed that no strong correlations were present among the variables. This outcome was beneficial as it signified that each variable contributed distinct and non-redundant information, thereby enhancing the performance of the model. The correlation map depicting these relationships is displayed in Fig 1.

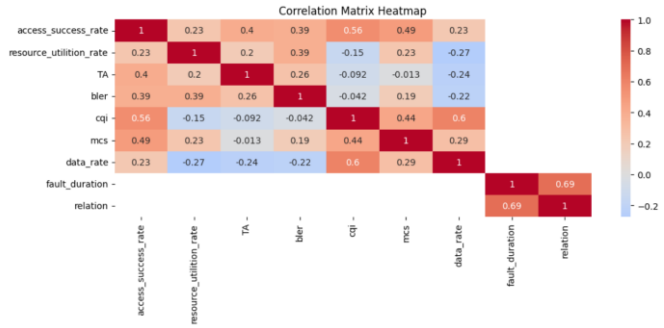

Fig. 1 – Feature Correlation

Worth mentioning is the significant effort dedicated to pre-processing, guided by exploratory data analysis (EDA), but the resulting f-score improvements were limited. Various techniques were explored, such as log-transforming skewed variables like the data rate, which exhibited right-skewed distribution (as seen in Fig 2). Attempts to normalize this skewness through log transformation did not yield the expected improvement in the scoring metric.

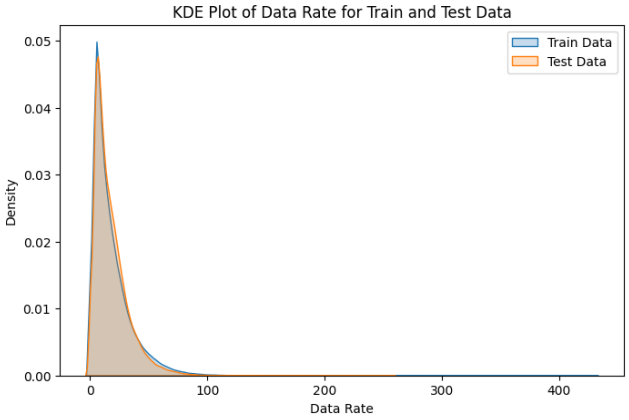

Fig. 2 – Distribution of 'Data Rate' Feature in Train and Test Datasets

In the attempt to improve model performance also, adversarial validation to assess the model's ability to distinguish between the training and testing datasets was pursued. In adversarial validation, a binary

classifier, adversarial classifier, is trained to predict if a sample belongs to the test dataset [1]. The ROC curve, depicted in Fig. 2, illustrates the trade-off between true positive and false positive rates, providing insights into the model's performance in detecting potential data leakage and ensuring the robustness of the training process.

Interestingly, during the preprocessing stage, an attempt was made to enhance the model's performance by dropping the features TA and Access Data Rate. This decision was based on the observation that removing these features resulted in a significant improvement in the AUC Score, improving it from 0.82 to approximately 0.63. However, we refrained from proceeding with the removal of these features as it did not produce favorable results and, in fact, negatively impacted the scores.

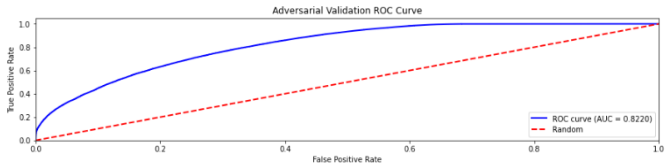

Fig. 3 – Adversarial Validation ROC Curve demonstrating the lightGBM effectiveness in distinguishing between the training and testing datasets

Efforts to standardize the columns using various scaling techniques, including min-max scaling, standard scaling, and robust scaling, also led to a degradation in results rather than their anticipated enhancement. Scaling, generally employed to bring features to a standardized range, should ideally improve model performance by making it less sensitive to the scale of individual features. However, in this specific context, the outcomes indicated otherwise probably indicating that the datasets provided were already standardized averages.

2.2 Data Cleaning

Upon initial examination, it became evident that certain rows in the same file contained multiple instances of faults. To streamline the analysis, a decision was made to consider only the first occurrence of a fault while disregarding any subsequent instances. This approach simplified the dataset and allowed a focus on the primary fault instance for analysis, resulting in a reduction in the combined dataset size from 1,084,072 rows to 995,544.

In the effort to ensure the dataset's coherence and concentrate on meaningful instances, rows where both the data rate and fault duration columns were zero were excluded. It's worth noting that data rate typically registers as zero during faults in standard situations. However, rows with both attributes equal to zero often presented inconsistencies unlikely to provide valuable insights for analysis. This decision further reduced the combined dataset's size, bringing it down from 995,544 rows to 957,480.

Subsequently, after applying the shifting process to incorporate the previous day's Key Performance Indicators (KPIs), it was discovered that the first row of each unique ID contained null values due to this transformation. To maintain data integrity and prevent potential misinterpretations, these rows were removed from the dataset. This step, while essential, had the effect of reducing the combined dataset from 957,480 rows to 948,292.

During the data cleaning efforts, some strategies did not yield the desired results. These strategies included attempts to handle missing values through experimentation with imputation methods such as means, medians, and the KNNImputer to fill gaps. Additionally, removing columns with greater than 30% missing values and eliminating rows with missing values had adverse effects on the scores.

2.3 Feature Engineering

To enhance the model's performance, supplementary features designed to exploit latent relationships among KPIs were introduced. In the absence of these additional features, the LightGBM model achieved an f-score of approximately 0.64, while CatBoost reached 0.65 during training and validation on distinct subsets of the training data. However, the incorporation of these supplementary features yielded significant improvements. Specifically, the f-scores for both models increased to around 0.7, reflecting around 9%-10% enhancement for both models.

Initially, time-related attributes were derived from the hourly timestamps. These attributes encompassed the hour of the day, month, day of the week, and a binary indicator denoting weekends. Notably, among these features, the "hour" attribute exhibited the highest importance in both models. In LightGBM, it emerged as the most influential feature as shown in fig.1, whereas in CatBoost, it held the position of the third most critical feature, according to feature importance rankings.

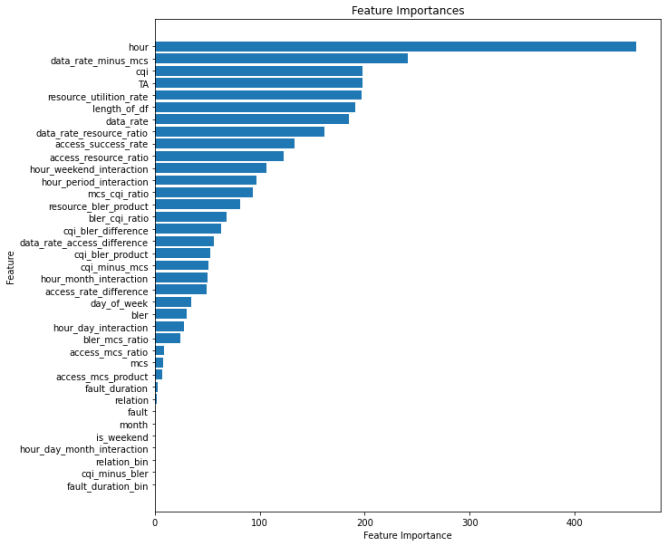

Fig. 4 – Summary of Feature Importance in LightGBM for top features

In addition to the time-related features, a set of interaction features was crafted from the KPIs and time-related attributes within the dataset. These interaction features encompass various calculations that aim to reveal nuanced relationships between the key performance indicators (KPIs). Ratios, differences, and products of KPIs were computed, shedding light on intricate associations. For instance, the 'access_resource_ratio' captures the proportion of access success rate concerning resource utilization rate, while the 'bler_cqi_ratio' delineates the balance between block error rate (BLER) and channel quality indicator (CQI). Moreover, these interaction features extend to highlight disparities and harmonies, such as the 'data_rate_access_difference,' cqi_bler_difference,' 'cqi_bler_product,' and 'access_mcs_product.' These engineered interaction features offer a richer context for modeling and are poised to enhance the predictive capabilities of the machine learning algorithms employed in the analysis.

To comprehend the dataset's temporal distribution, the frequency of occurrences for each unique 'NE ID' in the combined dataset was computed. This calculation yielded a new feature called 'length_of_df,' denoting the quantity of instances linked to each 'NE ID.' The values in the column indicated how many hours had lapsed up to to the occurrence of the fault in that particular NE.

An introduced discretization technique was used to categorize continuous features into bins, simplifying the analysis of 'relation' and 'fault_duration' values. For 'relation,' four bins were defined based on specific value ranges, and each value was assigned to an appropriate bin. A similar approach was applied to 'fault_duration,' where three bins were created based on predefined intervals, resulting in the 'fault_duration_bin' column. This binning process enhances the dataset's interpretability and usability by grouping related data points and facilitating further analyses.

Following the incorporation of Key Performance Indicators (KPIs) and the integration of newly engineered features, the consolidated dataset ultimately comprised 948,292 rows and encompassed 37 distinct features. At this juncture, the dataset was partitioned into training and validation subsets. This division resulted in a training subset comprising 946,360 rows and a separate validation subset containing 1,932 rows from the initial fault validation dataset.

Notably, the training dataset now consisted of more rows than the original training dataset. This enlargement incorporated the non-fault rows from the initial validation set for training purposes. Importantly, this did not pose a data leakage problem since the project did not treat the problem as a time-series task. In fact, leveraging these additional rows improved results slightly.

3. PROPOSED MODEL

To address missing values in the dataset, careful consideration was given to selecting machine learning models that excel in handling this challenge. The models taken into account included XG-Boost, CatBoost, HistGradientBoosting, and LightGBM. Given the dataset's considerable size, efficiency in terms of memory and speed was also a crucial factor. Following a comprehensive evaluation, CatBoostClassifier and LightGBMClassifier emerged as the optimal choices, striking a balance between dataset size and computational efficiency.

The two classifiers basically were similar with slight changes in thresholding which was 0.489 and 0.482 for LightGBMClassifier and CatboostClassifier respectively as shown below.

Algorithm: Predictions with classifier

1. Instantiate the classifier model:
    - Set random seed for reproducibility.
    - Set verbosity to False for minimal output during training.
    - Set thread count for parallel processing.
2. Fit the model on the training data and labels:
    - Use the train dataset as input features.
    - Use y as the corresponding target labels.
3. Predict probabilities of class 1 for the test data
4. Adjust the threshold for classification:
    - Choose a threshold value (0.49 for LightGBMClassifier and 0.482 for CatboostClassifier) to classify data points.
    - Convert the predicted probabilities to binary predictions based on the threshold.

It's also worth noting that the models predicted probabilities and not binary classes. These probabilities were later rounded to integers. Attempts to predict binary classes hurt the results.

In the final step of the analysis, predictions from two different machine learning models, CatBoostClassifier and LGBMClassifier, were combined to create a more robust prediction. Predictions for each row from both models were summed, and if the total reached or exceeded 1, it was considered a positive prediction.

4. TESTING AND EVALUATION

Log loss learning curves for evaluating results were considered. Learning curves provide insight into the dependence of a learner's generalization performance on the training set size. This important tool can be used for model selection, to predict the effect of more training data, and to reduce the computational complexity of model training and hyperparameter tuning [2].

In the context of the training log loss curves, a distinct pattern was observed while using both models: the validation loss curve initially began at a relatively high value and experienced a slight decrease before it flattened around 0.6. The training loss curve on the other hand commenced at a lower point and underwent a slight increase before it plateaued around 0.1. The fact that the training log loss was small indicated low bias.

It was also noted that there was huge gap between the training and validation losses as evidenced in fig.5 indicating a high variance problem. However, this could not be exactly described as an underfitting problem which in many cases is solved by adding more training samples. This is because the curves flattened out in the end meaning that adding more training examples would most likely not improve results.

These observations indicated a big irreducible error likely caused by complexities in the dataset such as the dynamic nature of RAN behavior, intricate network topology, diverse user segments, and the presence of non-RAN faults, as mentioned earlier.

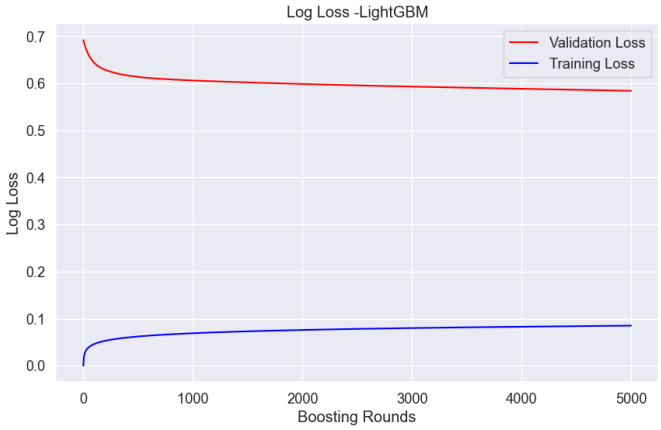

Fig. 5 – Training Loss curve with 5000 rounds and an early stopping of 10 rounds (80% Training, 20% Validation)

F1 Score was also considered for evaluating results as a comprehensive performance metric, combining precision and recall into a single measure. The F1 Score is computed as the harmonic mean of these two metrics, providing a balanced assessment of model performance. Precision represents the ratio of correctly identified positive items to the total items labeled as positive, while Recall (also known as Sensitivity or True Positive Rate) signifies the proportion of correctly identified positive items out of the total actual positives. These metrics, alongside True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN), were employed for comprehensive evaluation using scikit-learn.

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times precision \times recall}{precision \times recall}$$

When assessing the performance of the LightGBM model on a validation set generated through an 80-20 split of the training dataset, it achieved a score of 0.706, as detailed in Table 1. Subsequently, the same model was utilized to train on the entire training dataset and make predictions on an unseen withheld dataset, resulting in a score of 0.722.

Table. 1 – Performance of the models (80% Training, 20% Validation)

Metric	LightGBM f1-Score	Catboost f1-Score
Precision	0.679100	0.679106
Recall	0.737172	0.724415
F1-Score	0.706946	0.701029

On the other hand, CatBoost model achieved a score of 0.701, as detailed in Table 1 and 0.724 on the unseen withheld dataset.

The final ensemble model achieved a score of around 0.75 on the unseen withheld dataset.

5. DISCUSSION AND FUTURE WORK

The successful outcome of the proposed model solution serves as a foundation for future endeavors in advancing the field of service-oriented network operation and maintenance. There are several exciting avenues for further research and development that can build upon the accomplishments of the current model as discussed below.

1. Hyperparameter tuning

Perform hyperparameter tuning using grid search and cross-validation techniques to optimize each model's performance. This was never done for this project.

2. Enhanced Feature Engineering:

A deeper exploration of feature engineering could be undertaken. This involves identifying and incorporating additional relevant features from the provided dataset to potentially enhance the model's predictive capabilities.

3. Multi-Modal Data Fusion:

Considering the complexity of network fault analysis, integrating multi-modal data sources could prove beneficial. Combining RAN KPI data with other types of network-related data, such as maintenance logs, weather conditions, or geographical information, could offer a more holistic view of network behavior and contribute to more accurate fault impact predictions.

4. Transfer Learning and Generalization:

Finetuning the model on a larger, more diverse dataset, particularly those pretrained on time series problems, might improve its ability to generalize across various network scenarios and fault types, ensuring robust performance even in previously unseen conditions.

5. Real-time and Adaptive Analysis:

Adapting the model for real-time analysis could lead to proactive fault management and quicker response times. Investigating methods to incorporate streaming data and designing algorithms that dynamically adjust to changing network conditions could be a promising area of research.

6. CONCLUSION

In summary, the solution presented has the potential to assess the impact of faults on network performance effectively. The results have been examined through the utilization of two distinct models and, ultimately, the creation of the ensemble solution. Furthermore, potential innovative enhancements have been suggested to improve prediction accuracy in future projects.

ACKNOWLEDGEMENT

Special thanks to Antonio De Domenico from Huawei and Vishnu Ram from ITU for their comments. Much appreciations too to ITU and Zindi for organizing and hosting this interesting competition-*problem statement PS-002 - Fault Impact Analysis: Towards Service-Oriented Network Operation & Maintenance*.

REFERENCES

[1] J. Pan, V.Pham, M. Dorairaj , H. Chen, J. Lee. “Adversarial Validation Approach to Concept Drift Problem in User Targeting Automation Systems at Uber.” Available: https://arxiv.org/abs/2004.03045

[2] Viering, Tom, and Marco Loog. "The shape of learning curves: a review." IEEE Transactions on Pattern Analysis and Machine Intelligence (2022). Available: https://shorturl.ac/7bny6

[3] Source code of the proposed model (open-source) https://shorturl.ac/7bnxq

[4] International Telecommunication Union (ITU) official webpage: https://www.itu.int/