Fault Impact Analysis: Towards Service-Oriented Network Operation AND Maintenance by ITU

yisak bule

wachemo university

September, 15, 2023

Abstract *– In the telecom industry, the most significant thing is faulty management. Traditional fault analysis in RAN does not consider the impact of each fault on RAN KPIs. Machine learning can be used to predict the impact of faults on KPIs, which can help to improve the efficiency of fault management. The challenges of predicting the impact of faults on RAN KPIs include the stochastic nature of KPIs, the complexity of the RAN network, and the uncertainty caused by customer segmentation and non-RAN faults. In this experiment, I used machine learning methods to predict the impact of faulty on RAN KPIs. Different feature engineering and selection methods were employed. The most important features were the difference and ratio of RAN KPIs, lag and rolling, mean and target encoding, clustering, and operations. The results stack of LightGBM and Catboost in 50, 80, and 100 features had a 74.81 F1 score.*

Keywords – RAN KPIs, difference, ratio, mean encoding, binning

1. INTRODUCTION

In the telecom industry, fault management is essential for ensuring the reliability of networks and services. In the RAN, the most significant part of fault management is fault analysis. The traditional way to analyze faults is to set rules based on network experts' experience. However, this method does not take into account the impact of each fault on network KPIs [1].

A new approach to fault analysis is to use machine learning to predict the impact of faults on RAN KPIs. This approach has several advantages over the traditional method. First, it can take into account the complex relationships between faults and KPIs. Second, it can be used to predict the impact of faults that have not been seen before. Third, it can be used to optimize the allocation of O&M resources [1].

The challenges of predicting the impact of faults on RAN KPIs include the stochastic nature of RAN KPIs, the complexity of RAN networking and mechanisms, the uncertainty caused by customer segmentations, and the uncertainty caused by non-RAN faults [1].

2. Data preprocessing

2.1 Data cleaning

The output of a machine learning model depends on its input value [2]. All machine learning models need clean input data, to give better results [2]. This dataset contained different corrupt data points. From the dataset file, 52 records had no non faulty data point, and so those records were dropped because we can not create a label if there is no non faulty time data rate value. Also, some rows contain relation values less than 0, And they are replaced with 0. In addition, there were some rows that had more than a one hour gap between before the fault occurred and after the fault occurred, these rows were not dropped due to the short time to experiment with its effect.

2.2 Feature engineering

Machine learning models can not capture some patterns without human help, And feature engineering is an essential step in a machine learning project [3]. Feature engineering is the creation of new features from existing ones [4]. Feature engineering gives more generalization for machine learning models and a better way to improve scores instead of tuning model hyperparameters [3]. In this project, I used many feature engineering methods.

2.2.1 The difference and ratio change of KPI features

In the experiment, I calculated the difference one, difference two, and difference three of the RAN KPIs features. These features greatly helped the model learn more about the data. For example, suppose there are two last hour data rate values [0, 5, 15] and [100, 94, 93], and the first one increases exponentially (5 to 15) while the second one decreases from 94 to 93, But when we compare

Both values in the second (93) are greater than the first (15), so if we don't give differencing information for the model, the model will overfit on this data point, This improves the performance and training stability of the model. Likewise, other KPI features such as access_success_rate, resource_utilition_rate, TA, bler, cqi, and mcs were computed as the difference one, two, and three.

And another interesting feature was the binarization of difference, if the difference value is greater than 0, the label will be 1, otherwise 0. This avoids outliers and gives the probability of increasing or decreasing. For example, this is the last three hour data rate value [-3, 40, -10] and [-1, 1, 3]. When computing the mean value, the first is 9 and the second is 1, so the mean value is indicating a more positive value for the first one, but the first one has decreased in the last hour, while the second one is increasing slightly. When the model gets this mean value, it will give a higher probability for the second one to decrease in the next hour. That is why binarizing the difference prevents outlier impacts on the model's performance.

Table 1 – *Difference one of RAN KPIs*

endTime	data rate	diff	diff1
2/20/2023 8:00	38.5	nan	
2/20/2023 9:00	24.5	24.5- 38.5	-14
2/20/2023 10:00	21.8	21.8 - 24.5	-2.7
2/20/2023 11:00	23.2	23.2 - 21.8	1.4

Table 2 –*Difference two RAN KPIs features*

endTime	diff1	diff2
2/20/2023 8:00		
2/20/2023 9:00	-14	
2/20/2023 10:00	-2.7	-2.7-14
2/20/2023 11:00	1.4	1.4-2.7

And another one ratio of KPI values this give more information for the model for example if have [2.3, 4.6] and [120, 130] and this time if compute the difference the second one has more positive value but increase from 120 to 130 is not so high ratio(130/120) but when we compute the ratio second value is increasing double(2.3 to 4.6) so this time the model can understand some information that how much weight increased the data value and highly prevent model overfitting in new data because if we give new value [1000, 1200] then when compute the ratio it will be 1.2 that means less than from first one([2.3, 4.6]). And also, the ratios 2 and 3 were computed.

Table 3 *Ratio* of RAN KPIs features

endTime	data rate	Ratio change	Ratio one
2/20/2023 8:00	38.5		
2/20/2023 9:00	24.5	24.5 / 38.5	0.64
2/20/2023 10:00	21.8	21.8 / 24.5	0.89
2/20/2023 11:00	23.2	23.2 / 21.8	1.06

2.2.2 The simple weighted mean

The data rate value had a correlation with previous values of data rate, and I took the 8 weighted mean of data rate features . This gives more weight to recent values, and so recent values have a greater impact on the mean value that may indicate the current data rate change value. All RAN KPIs features were computed as a simple weighted mean in differences one, two, and three features, from windows two to eight.

2.2.3 The lag and rolling features

To get some previous time patterns about data rate values, I took 24 lag features and also 5 rolling mean, max, and min KPI features.

2.2.4 Creating categorical columns

To compute the mean or target encode, we need to have categorical columns. Sometimes, if we don't create it carefully, it may overfit,, and it needs more understanding of the nature of data.

In this project, many categorical columns were created. The first way was by binning the numerical columns, Such as the faulty duration and relation columns. The faulty duration columns ranged from 1 to 3600 seconds; this was binned in ten categories, which means six minutes in one category. The second one's relation column was categorized into ten categories, the relation range was 0 to 1, so the first category was 0 to 0.1, the second one was 0.1 to 0.2, and so on. Another way was combining two or more columns, for example, hour, week of day, and faulty duration that was binned, then combine them and create one column. In this way, more than 20 columns were created. I saw an interesting column that creates the binary of diff1, diff2, and diff3, then sums them and combines them with other columns.

2.2.5 Mean encoding

Some raw data had not much information about previous hour KPI features, more than 30% of test data had only one hour KPI features (before occurring faulty), and 50% of test data had less than 16 hour non faulty history KPI points. So the machine learning algorithms will not capture the pattern properly and it will underfit, so that means encoding helps to model to get more data and learn more about patterns of data. For example, hour is a common feature that may be available for all NE IDs, so grouping by hour and computing the mean, max, min, and so on. Finally, we can merge them with an hour, e.g., (1:0.1, 2:0.7, 3:0.5). The important thing here is creating new features that may not be available for half of the data and improving the generalization of the model across all of the data. Also, it discovers some facts about the probability of the data rate increasing or decreasing in the next hour, that may not be found in the model from other features. The experiment implies that The most important features were grouping by NE ID and hour, then computing the mean, max, and Min. In this experiment, we didn't compute the 25 quantile and 75 quantile because, due to RAM memory shortage, it probably would have given more information.

2.2.6 Target encoding

When we compute the target encoding in wrong way, the model will overfit easily. Target encoding needs to be created carefully, otherwise, it will be overfitted. In this project, I experimented with different target encoding methods, but data leakage was major problem. To avoid data leakage problems, I used only target encoding in a few columns, like hour, day, and other columns.

2.2.7 Some operations on KPI features

Feature engineering is time consuming by nature, and it needs a good understanding of data [3]. Automatically creating feature engineering is a good option to save time and create unexpected features [6]. In this project, an automatically create features function, was created. This function takes two arguments, which are df and cols. Then it will take one column and add, subtract, multiply, and divide it from the rest of RAN KPI features.

2.3 Feature selection

Feature selection is an effective way to reduce high dimensionality to low dimensionality [7]. Machine learning models overfit on redundant features, so feature selection is the best way to improve the accuracy and stability of the model [8]. In this experiment, LightGBM feature importance was used for feature selection.

2.4 Validation strategy

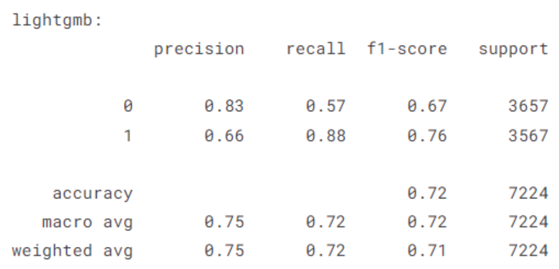
Machine learning algorithms need a balanced class to provide good accuracy. And when target class is imbalanced, it will overfit the majority class [9]. To avoid this kind of problem, choosing the right evaluation metric and cross validation strategy is a critical step. Since the target class is slightly imbalanced and the metric is F1 score, the stratified K fold was employed in 10 splits.

3. Results

3.1 model selection

Depending on the nature of the data, some models outperform others. ANN (artificial neural network) well fits unstructured data [10], but boosting tree models are still leading on tabular data [11]. Depending on nature of dataset, I used tree based models such as LightGBM and Catboost.

3.2 model results

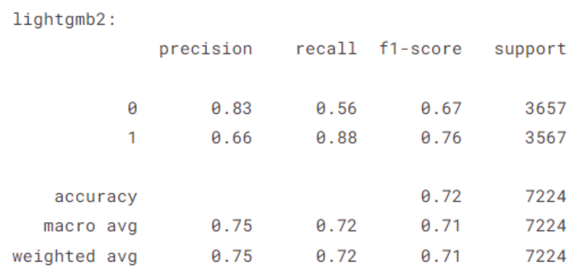


```
lightgmb:
              precision    recall  f1-score   support

           0       0.83      0.57      0.67      3657
           1       0.66      0.88      0.76      3567

    accuracy                           0.72      7224
   macro avg       0.75      0.72      0.72      7224
weighted avg       0.75      0.72      0.71      7224
```

Fig. 1 – lightgmb model on 500 features classification report

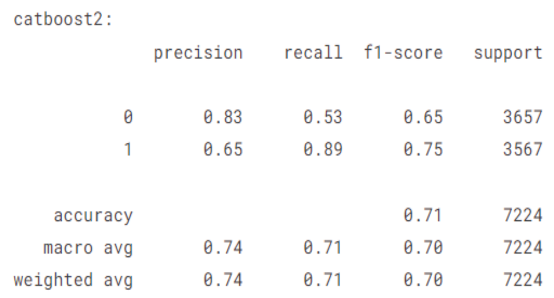


```
lightgmb2:
              precision    recall  f1-score   support

           0       0.83      0.56      0.67      3657
           1       0.66      0.88      0.76      3567

    accuracy                           0.72      7224
   macro avg       0.75      0.72      0.71      7224
weighted avg       0.75      0.72      0.71      7224
```

Fig. 2 – lightgmb model on 300 features classification report.

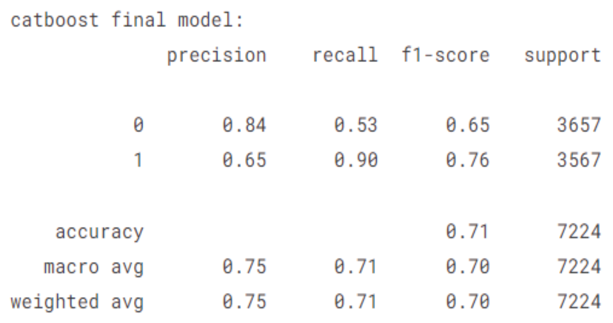


```
catboost2:
              precision    recall  f1-score   support

           0       0.83      0.53      0.65      3657
           1       0.65      0.89      0.75      3567

    accuracy                           0.71      7224
   macro avg       0.74      0.71      0.70      7224
weighted avg       0.74      0.71      0.70      7224
```

Fig. 3 – catboost model on 500 features classification report.



```
catboost final model:
              precision    recall  f1-score   support

           0       0.84      0.53      0.65      3657
           1       0.65      0.90      0.76      3567

    accuracy                           0.71      7224
   macro avg       0.75      0.71      0.70      7224
weighted avg       0.75      0.71      0.70      7224
```

Fig. 4 – stack of three model classification report.

Table 4 *The result of models*

model name	n_feature	f1 score	
		local	private
lightgbm	500	75.7	74.53
lightgbm	300	75.8	74.44
catboost	500	75	75.0
stack of three		75.5	0.7495

3.3 feature importance

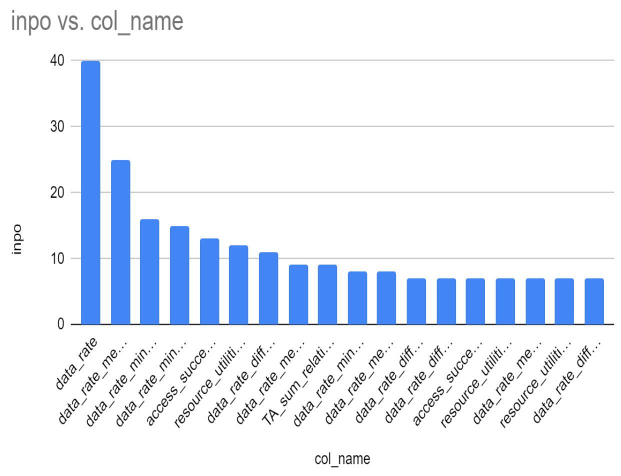


Fig. 5 – features importances.

CONCLUSION

The machine learning method showed better results than conventional methods for analyzing the impact of faulty RAN KPIs. In this experiment, the gradient boosting method achieved a better score than another machine learning model. In 50, 80, and 100 selected features, the LightGBM and Catboost models have achieved a 74.81 F1 score. In this experiment, there were more than 30,000 features, but due to a shortage of time, I used less than 100 features and 7222 samples. This result was achieved only on less than 100 features, and when adding more features, it increases the results more than this.

ACKNOWLEDGEMENT

Special thanks, to ITU team for organizing the competition and zindi team for hosting this great competition.

REFERENCES

[1] ITU "fault impact analysis: towards service-oriented network operation & maintenance by ITU." ITU zindi (2023)

[2] Lee, Ga Young, Lubna Alzamil, Bakhtiyar Doskenov, and Arash Termehchy. "A survey on data cleaning methods for improved machine learning model performance." arXiv preprint arXiv:2109.07127 (2021): 2-3.

[3] Verdonck, Tim, Bart Baesens, María Óskarsdóttir, and Seppe vanden Broucke. "Special issue on feature engineering editorial." Machine Learning (2021): 1-12.

[4] Turner, C. Reid, Alfonso Fuggetta, Luigi Lavazza, and Alexander L. Wolf. "A conceptual basis for feature engineering." Journal of Systems and Software 49, no. 1 (1999): 3-15.

[5] Pargent, Florian, Florian Pfisterer, Janek Thomas, and Bernd Bischl. "Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features." Computational Statistics 37, no. 5 (2022): 2671-2692.

[6] de Melo, Vinicius Veloso, and Wolfgang Banzhaf. "Automatic feature engineering for regression models with machine learning: An evolutionary computation and statistics hybrid." Information Sciences 430 (2018): 287-313.

[7] Khalid, Samina, Tehmina Khalil, and Shamila Nasreen. "A survey of feature selection and feature extraction techniques in machine learning." In 2014 science and information conference, pp. 372-378. IEEE, 2014.

[8] Li, Jundong, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. "Feature selection: A data perspective." ACM computing surveys (CSUR) 50, no. 6 (2017): 1-45.

[9] Raeder, Troy, George Forman, and Nitesh V. Chawla. "Learning from imbalanced data: Evaluation matters." Data Mining: Foundations and Intelligent Paradigms: Volume 1: Clustering, Association and Classification (2012): 315-331.

[10] Lu, Hongxia, Louis Ehwerhemuepha, and Cyril Rakovski. "A comparative study on deep learning models for text classification of unstructured medical notes with various levels of class imbalance." BMC medical research methodology 22, no. 1 (2022): 181.

[11] Grinsztajn, Léo, Edouard Oyallon, and Gaël Varoquaux. "Why do tree-based models still outperform deep learning on typical tabular data?." Advances in Neural Information Processing Systems 35 (2022): 507-520.

AUTHORS

First Author I am Yasak Birhanu Bule, and i am 23 years old. I g deploma in HNS (hardware network service) at Hossana Polytechnic College and also have a bachelor's in Biology at Wachemo University. Currently, I hold a master's degree in applied microbiology.