Exploring the WARA-Ops 5G dataset for QoS prediction

Risberg Alaküla, Anton
anton.risberg_alakula@cs.lth.se

Nyberg Carlsson, Max
max.nyberg_carlsson@control.lth.se

Momeni, Niloofar
niloofar.momeni@matstat.lu.se

Ng, Harald
hng@kth.se

October 2023

Abstract

5G radio data contains many properties that can be measured and reacted to appropriately. The strength of the received signal, the number of lost packets, the latency of those packets, etc. are different ways to measure the quality of the connection. A wireless device needs to determine its signal strength in order to adjust how much energy to put in the outgoing signal, how many error correction bits to include in the message, etc. In this project we received a large set of 5G radio data, and tried analyzing it to determine the shape of the data and which properties are most important for determining signal quality. We also tried to predict issues before they appear, so that a 5G device would be able to act proactively to combat signal quality degradation.

The results are unfortunately mixed; the data set is very noisy, and the different signals correlate less than expected. Still, we have some results to show. This document contains the findings and links to experiments that can be explored further.

1 Introduction

In this work we focus on the phone use case of 5G radio communications. Phones (falling under the category user equipment, or UE) have some challenges to overcome to make communications reliable. For one, phones are usually running on a battery, and thus they want to spend as little energy as possible on communication in order to prolong battery life. At the same time, spending more energy increases the likelihood of the message arriving intact. Correctly determining the quality of the signal can help the UE decide how to transmit and encode messages.

We received a large data set from Ericsson that contains 5G radio data, the data set contains nearly a million samples where each sample contains around 65 fields. The data is elaborated upon in Section 1.2. The goal of the project is to detect signal quality anomalies, preferably ahead of time. This might seem easy to do due to the amount of data we have access to. However, the term "anomaly" is not well defined. In fact, a common problem when studying anomalies is how rarely they can occur. Therefore, part of our task is to determine what anomaly means. Once done, we should develop a model that detects that anomaly and can predict when an anomaly likely will occur.

1.1 Supplementary Resources

The result of the work is presented in multiple parts.

- This document contains some background information and overall descriptions of the performed work.
- There are a number of Jupyter Notebooks available[1] for cleaning and analayzing the data. These have been supplied to WARA-Ops.
- We have a website[2] available that lets the user interactively explore the dataset.
- Two videos showing the background problem and solution, to be shown at the WASP Winter Conference 2024.

1.2 Supplied Data

The data to be analyzed was collected over a span of 6 months, when the UE was transported throughout southern Sweden in a car or boat. Data from different layers, e.g. application and radio layers, have been compiled in a comma separated value file where each row represents measurements from a certain point in time.

Data collected throughout multiple data collection sessions were concatenated into the supplied data file, occasionally causing large time intervals between data points. The type of data collected may also differ between sessions, adding additional columns which have to be handled appropriately when parsing the file. To accommodate for this, fields to be analyzed can be chosen beforehand and all relevant data can be extracted into a pruned data file.

The field naming convention appears to follows the Android API, yielding some limited hints of the meaning of the data. Not provided however, and thus had to be deduced, is the units. Notably, the notation for when no measurement seems to be available differs between fields with numerical data. Some fields, such as ping values, are set to zero whereas radio data, such as signal strength indicator, are set to the maximum signed 32-bit integer.

[1] `https://github.com/Kevlanche/wasp-2023-project-6`

[2] Hosted on Ericsson's data center at: `https://129.192.69.188/`

1.3 Error Correction Codes

When two nodes in a network communicate, there is always a risk that the messages get slightly changed during transport. 5G radio, for example, has to send signals over air and might encounter:

- Signals from other devices
- Building, trees, hills, etc.
- Various small particles (smoke, pollen, rain, snow, etc.)

All of these can cause a few bits of the transported message to change or not even arrive at the intended destination. To combat this issue, a common solution is to use error-correcting codes. These are extra bits of information that are not part of the message intended to be sent by the user, but can be used to verify that the received message is correct, and even to repair it in case some data was changed during transport.

In 5G, the standard is to use Polar Codes, which was introduced by Erdal Arikan [1] in 2009, and allows a varying number of error bits to be used. In this project, we aim to predict cell signal issues just before they happen. If successful, then this prediction could be used to proactively set the number of error correction bits accordingly. This would allow the phone to be more efficient by sending a low number of error correction bits while the cell signal is strong, and avoid failed messages by sending more error correction bits just before the signal degrades.

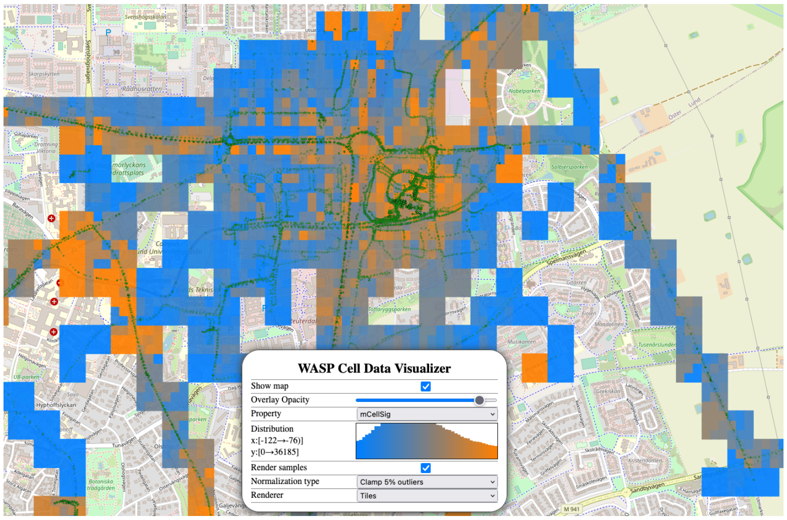
2 Solution

This work was performed by a group of people with varied interests and skill sets. To accommodate this, the work was done as several smaller experiments. Each experiment aimed at trying to understand or predict the data and underlying system behavior. In this section, each experiment and its contribution to the overall goal will be described in turn.

2.1 Exploration Website

One way to build understanding of data is to visualize and play with it. To this end, an interactive website was created that allows the user to visualize the supplied data on top of a world map. A screenshot from the website is visible in Figure 1.

The areas on the map can be colored and/or made opaque/transparent based on the property value associated with that point in the map. In Figure 1, the property being displayed is *mCellSig*, a measurement for cell signal strength. The distribution of cell signal values ranges from -122 to -76 in the dataset, after removing some outliers such as ”not available”. Values close to the bottom of this range are rendered blue, and those close to the top of the range are orange,



WASP Cell Data Visualizer

Show map ☑

Overlay Opacity

Property: mCellSig

Distribution
x:[-122→-76)]
y:[0→36185]

Render samples ☑

Normalization type: Clamp 5% outliers

Renderer: Tiles

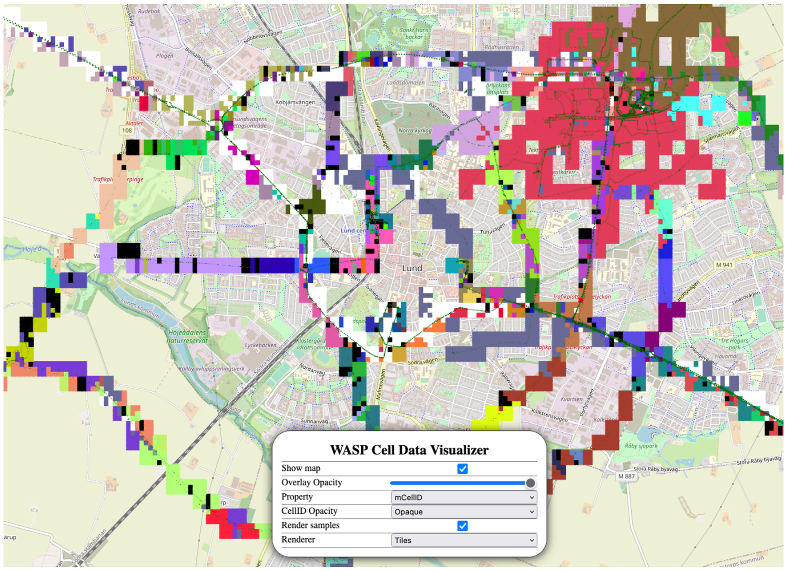
Figure 1: Screenshot from data visualizer website, showing *mCellSig* values. Orange squares indicate a high (good) value, and blue are low (bad). The small green circles are individual sample points. Each background square is colored based on the closest individual sample point.

with gradients in between. The tiny green and black circles are individual sample points. The figure is centered on the main Ericsson office in Lund. That office has two base stations on top of it, so cell signal quality being good there is expected.

The `Property` drop-down can be used to visualize the high and low values of a large number of properties. One property, however, is rendered differently, namely *mCellID*, and this can be seen in Figure 2. Each radio base station has one or more identifiers associated with it, and they are represented as *mCellID* in our data. Since these identifiers are somewhat arbitrary, rendering distributions as high/low does not produce any meaningful results. Instead, the visualizer website attempts to assign unique colors for each identifier. This results in a map where we can see roughly which areas a *mCellID* covers.

2.2 AI/Notebook Analysis

We converted the timestamp data from nanoseconds to seconds for easier understanding. When we plotted a variable against time, it became clear how data was collected in batches throughout the year. The data set covers six months, but it has extensive periods without any recordings. Only a few instances show continuous data, usually lasting just a few minutes or hours within the entire six-month timeframe.



WASP Cell Data Visualizer

Show map ☑

Overlay Opacity

Property: mCellID

CellID Opacity: Opaque

Render samples ☑

Renderer: Tiles

Figure 2: Cell IDs rendered with unique colors in visualizer website.

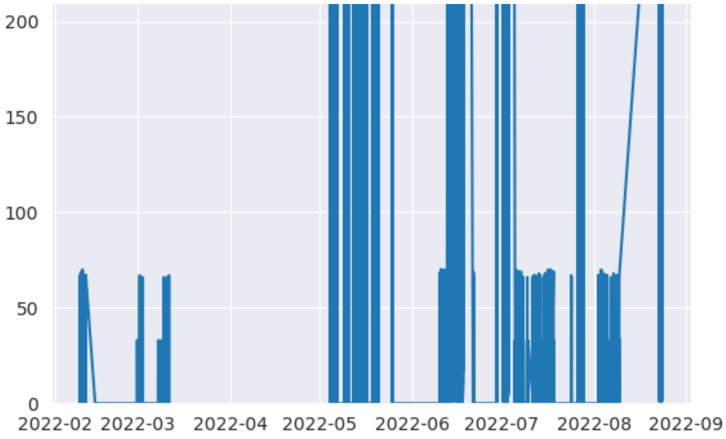


Figure 3: Ping Loss over time

2.2.1 Feature Engineering

New columns are being created based on various time-related attributes and specific conditions within the dataset. The extracted time components include seconds, minutes, hours, month, day of the month, day of the year, and day of the week from the date column. This allows for more granular analysis based on different time units. A new binary column *is_wknd* is created to identify weekends. It assigns a value of 1 if the day falls on a weekend (Saturday or Sunday) and 0 otherwise. Two additional columns, *is_month_start* and *is_month_end*, are generated. They denote whether the timestamp is at the start or end of a month, respectively. A new column *ChangeIndicator_CellID* is created to track changes in the *mCellID* column. It has a value of 1 when the *mCellID* changes compared to the previous row and 0 otherwise. This feature can help identify transitions in the cell ID. Another new column *Loss* is generated to denote network packet loss. It assigns a value of 1 if there is non-zero ping loss (*mPingLoss* is not equal to 0) and 0 otherwise.

We scan through the data and wherever it encounters the value 2147483647, it substitutes that value with 'NaN', ensuring that these values do not interfere with subsequent analysis or computations.

2.2.2 Channel Quality Error Prediction using Machine Learning

Among the fields is *mCqi*: Channel Quality Indicator, which indicates the quality of the wireless channel. Notably, *mCqi* displays numerous occurrences of the maximum signed 32-bit integer value (2,147,483,647), commonly signaling an error, overflow, or missing data in many computing environments. With over 262000 instances, *mCqi* surpasses other variables in exhibiting this behavior. We define the *Error* target variable, which is 1 when *mCqi* contains the maximum integer value and is 0 otherwise.

The analysis reveals highly correlated features[3] associated with the *Error* target variable. Notably, the following features exhibit strong correlations:

Table 1: Highly correlated features with *Error*

app build type	board	boot loader	hard ware	host	mCell Conn	mCell ID	mCell Pci	mCell Tac	mCqi	m Earc fn	m Rssnr
0.82	0.81	0.62	0.79	0.54	0.81	0.72	0.48	0.73	1.00	0.46	0.74

The high correlations suggest a strong relationship between these features and the occurrence of communication errors indicated by *mCqi* overflow conditions. These features encompass diverse aspects, such as device information (*DEVICE_INFO_appbuildtype*, *DEVICE_INFO_board*, *DEVICE_INFO_hardware*), connectivity metrics (*mCellConn*), and cell identification (*mCellID*, *mCellTac*).

[3] Used here interchangeably with the word fields, as it is common jargon in the field of machine learning.

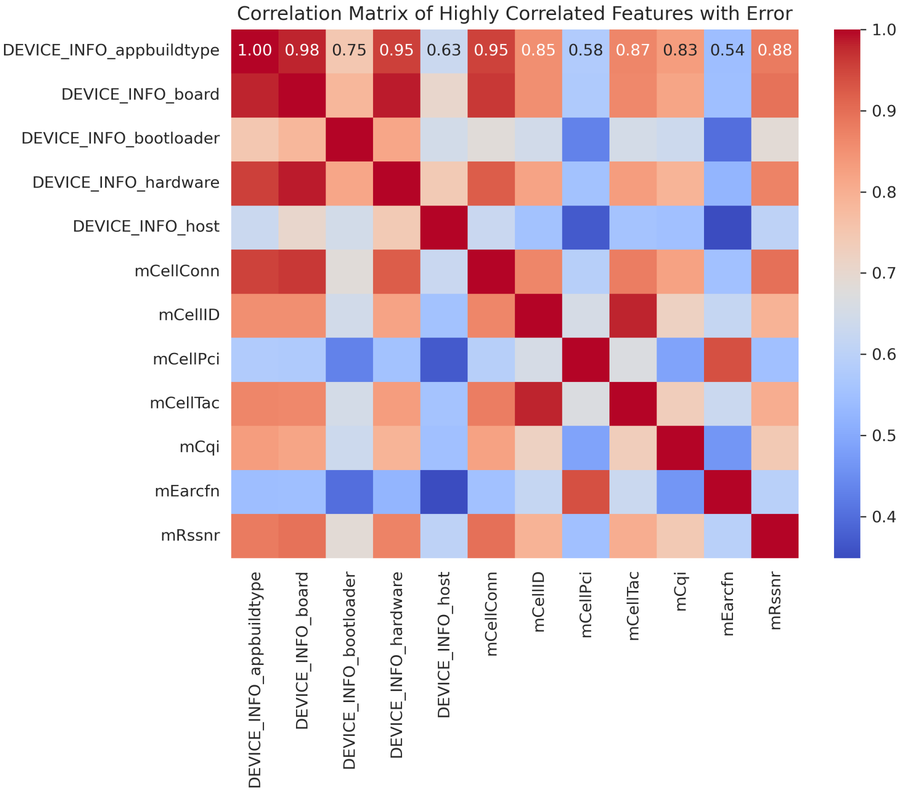


Figure 4: Correlation matrix for highly correlated features when there is an error in channel quality

XGBoost, short for eXtreme Gradient Boosting, is a powerful and popular machine learning algorithm used for predictive modeling. It belongs to the gradient boosting family and is known for its speed, accuracy, and efficiency in handling structured data. It builds a series of decision trees sequentially, focusing on minimizing errors and enhancing prediction accuracy through an iterative process [2]. An XGBoost model is trained on the training data excluding the *mCqi*, *Error*, *date*, *mPingHost*, *time*, *mTimeStamp*, *time_second*, and *mCellInfoTS* variables. The target variable is *Error* which indicates when there is an error in channel quality. The classification results on the unseen test data are as follows.

Table 2: Classification report for the channel quality error prediction

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0	1.00	1.00	1.00	144891
1	0.99	0.99	0.99	52851
Accuracy	0.99 (197742 samples)			
Macro Avg	0.99 (Precision), 0.99 (Recall), 0.99 (F1-Score), 197742			
Weighted Avg	0.99 (Precision), 0.99 (Recall), 0.99 (F1-Score), 197742			

In Figure 5, the significance of features on error prediction is illustrated. The features are arranged according to their importance within the machine learning model's error prediction. SHAP (SHapley Additive exPlanations) values are a method used in machine learning to explain the individual predictions of a model. They assign each feature in a prediction a value, indicating its contribution to the prediction's outcome. SHAP values aim to provide insights into how much each feature influences a specific prediction, contributing to better interpretability and understanding of a model's behavior [3]. A positive SHAP value implies a tendency towards predicting an error, while a negative SHAP value indicates a prediction leaning towards no error. The color intensity of feature values is scaled, where higher values are represented in red and lower values in blue. For instance, as the value of the *mRssi* feature increases, the likelihood of predicting an error also rises, as indicated by its higher red shading and placement on the positive side of the SHAP plot. Figure 6 shows the contribution of *mRssi* (Received Signal Strength Indicator), *mTimingAdvance* (a parameter used to synchronize transmissions in the mobile network), *mPingAge* (the age of ping or network latency measurements), *mSsRsrp* (Reference Signal Received Power in mobile network communication), *mPingLoss* (Packet loss during network communication), *ChangeIndicator_CellID* (Indicate when the Cell ID has changed), *mSpeed* (physical speed of the UE), and *mRsrp* (Reference Signal Received Power) to the channel quality error prediction.

For *mRssi* (Received Signal Strength Indicator) feature the plot in 6 indicates a predictive pattern observed in the model where different ranges of *mRssi* correspond to specific SHAP values. When *mRssi* values are outside a certain range (-130 to zero), the model predicts a decline in channel quality (positive SHAP). However, when *mRssi* falls within that range, the prediction leans toward an improvement in channel quality (negative SHAP).

For the *mTimingAdvance* we can see that the model predicts that there is a decline in the channel quality (positive SHAP value) and when the timing to synchronize transition is small and less than zero then the model prediction is there is no error in channel quality. A higher *mTimingAdvance* value might imply issues such as signal interference or problems in timing synchronization, leading to a decline in channel quality.

The relationship between *mPingAge* and channel quality can depend on multiple factors and is not straightforward in this model.

The model's predictions regarding *mSsRsrp* exhibit some inconsistencies. The model forecasts low channel quality when the signal strength falls below -140, which aligns with conventional understanding. However, it also predicts errors when the signal strength exceeds -60, which contradicts expectations since stronger signal strength (higher mSsRsrp) generally signifies better channel quality. Yet, the model's predictions within the mSsRsrp range of -140 to -60, indicating good channel quality and no error, seem reasonable, as this range typically represents robust signal strength levels in line with better channel conditions.

The prediction based on *ChangeIndicator_CellID* aligns with our expectations. Specifically, when there is an error in channel quality, the change indicator reflects a cell ID alteration (*ChangeIndicator_CellID* is 1 for positive SHAP), as anticipated.

2.2.3 Packet Loss Prediction using Machine Learning

The feature *mPingLoss* represents packet loss during network communication, indicating the percentage of data packets lost. Another target variable can be defined as loss which is one when the *mPingLoss* is not zero and zero otherwise.

An XGBoost model is trained on the training data excluding *mPingLoss*, *Loss*, *date*, *mPingHost*, *time*, *mTimeStamp*, *time_second*, and *mCellInfoTS* features. The target variable is *Loss* which indicates when there is packet loss. The classification results on the unseen test data are:

Table 3: Classification Report

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0	0.95	1.00	0.97	136983
1	0.99	0.88	0.93	60759
Accuracy	0.96 (197742 samples)			
Macro Avg	0.97 (Precision), 0.94 (Recall), 0.95 (F1-Score), 197742			
Weighted Avg	0.96 (Precision), 0.96 (Recall), 0.96 (F1-Score), 197742			

In Figure 7, the significance of features on loss prediction is illustrated. The features are arranged according to their importance within the machine learning model's loss prediction. A positive SHAP value implies a tendency towards predicting loss, while a negative SHAP value indicates a prediction

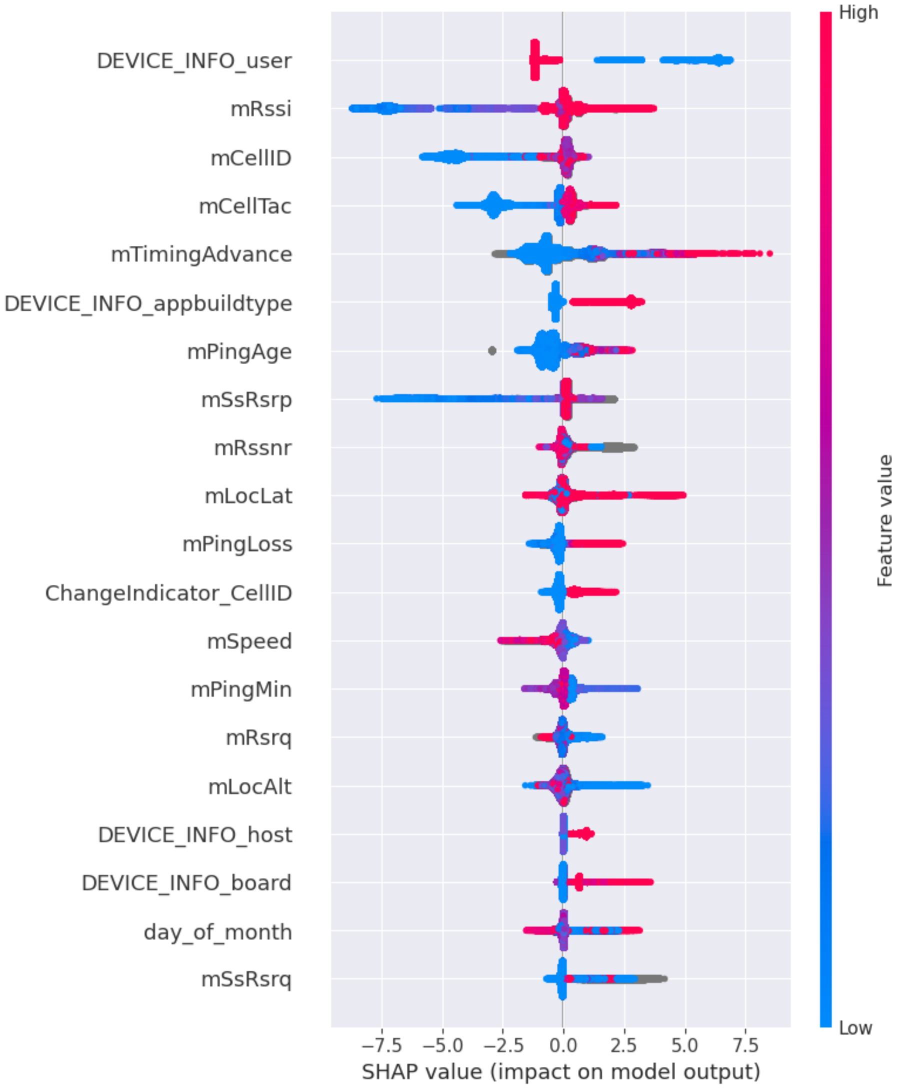


Figure 5: Most important features and their contribution to the channel quality error prediction.

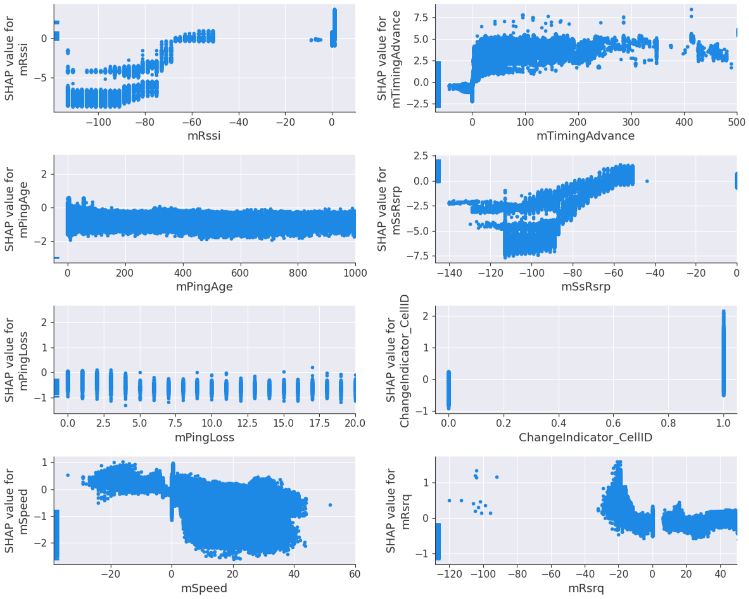


Figure 6: Contribution of features to the channel quality error prediction. Positive SHAP value means prediction toward error and negative SHAP value means there is no error predicted.

leaning towards no loss. In Figure 7, as the value of the *mRsrp* (Reference Signal Received Power) is very large, the likelihood of predicting a loss also rises, as indicated by its higher red shading and placement on the positive side of the SHAP plot. Figure 8 shows the contribution of *mRsrp* (Reference Signal Received Power),*mPingAvg*, *mRssnr*, *mCqi*, *hour*, and *mSpeed*.

For *mRsrp* (Reference Signal Received Power) feature the plot in 8 indicates when it is out of a signal power range, that is when it is more than zero, (a mobile signal power levels typically range from around -50 dBm to -120 dBm) the likelihood of predicting a loss also rises since the SHAP value is positive.

For *mPingAge*, Figure 8 shows that for very low and very high average ping the loss prediction decrease, and for a certain range of ping value the chance of loss increase.

For the analysis involving the *mRssnr* (Received Signal to Noise Ratio) in conjunction with the *mRssi* the plot reveals that as the signal-to-noise ratio decreases, there is an observed increase in the prediction of signal loss. Interestingly, it's notable that when the signal-to-noise ratio diminishes, the signal strength appears extraordinarily high (depicted in red). This anomaly in signal strength, however, seems misleading as it reaches the maximum integer value, falling outside the expected range. Conversely, it's evident that with higher signal-to-noise ratios, the signal strength tends to be consistently higher.

For *mCqi* (Channel Quality Indicator) plot in Figure 8 shows that when the

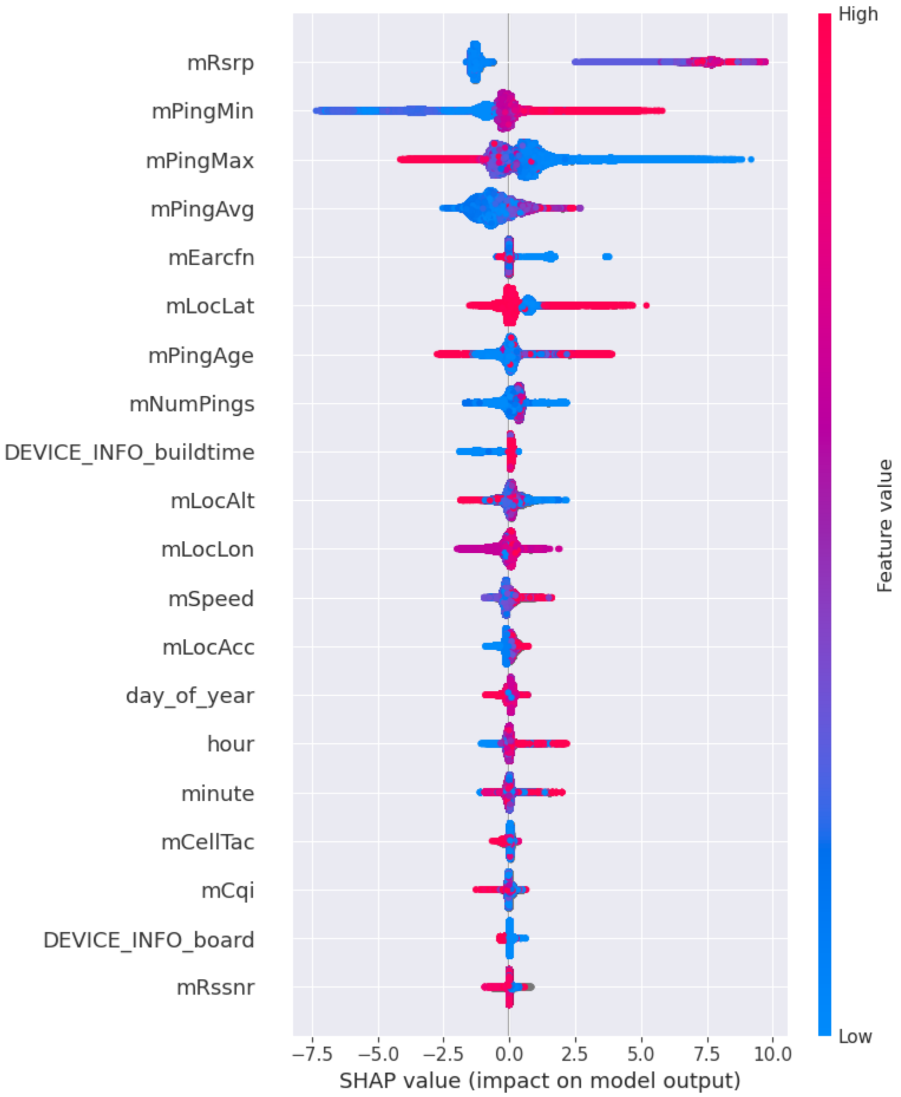


Figure 7: Most important features and their contribution to the packet loss prediction.

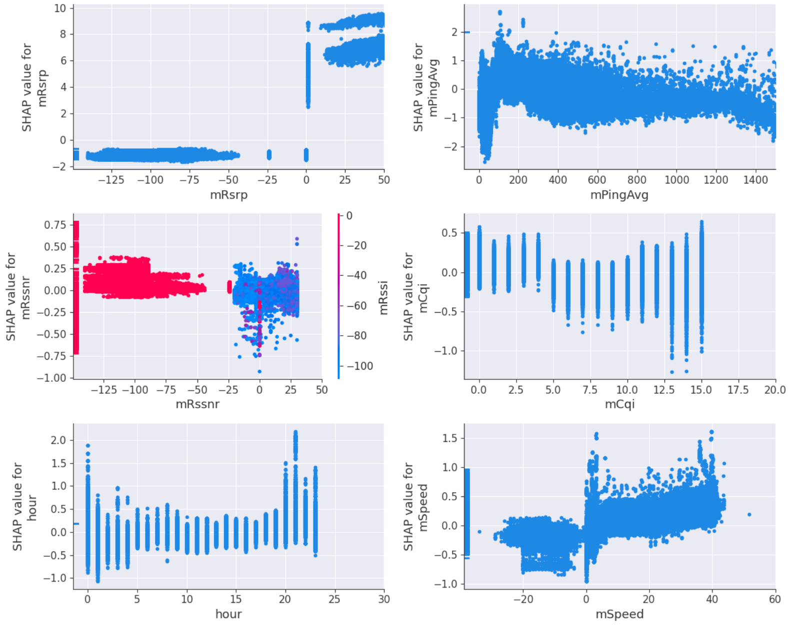


Figure 8: Contribution of features to the packet loss prediction. Positive SHAP value means prediction toward loss and negative SHAP value means there is no loss.

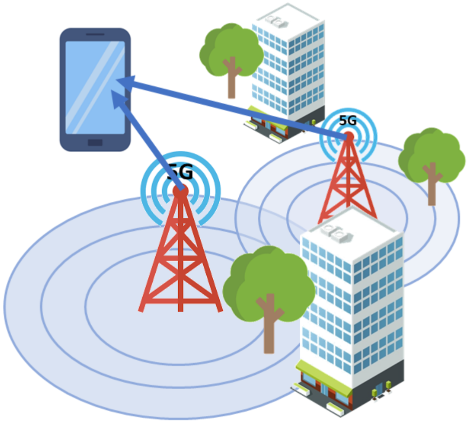


Figure 9: Handover in mobile 5G network.

channel quality decrease, the probability of predicting loss increase.

The plot for *hour* illustrates a decline in the likelihood of experiencing loss between 1 am and 4 am. This decrease might be attributed to an improvement in service quality during these early morning hours, likely due to reduced network congestion. Conversely, between 8 pm and 12 am, predominantly during the night, there appears to be an increased probability of encountering network packet loss, possibly due to increased network activity or peak usage hours, leading to potential congestion and higher chances of packet loss.

The plot depicting *mSpeed* demonstrates that as the speed increases, there is a corresponding rise in the probability of network packet loss. This relationship is logical since higher speeds might lead to transitions between network cells or areas at a faster pace, potentially causing intermittent connectivity or handover issues, thereby increasing the likelihood of packet loss.

2.2.4 Handover Prediction using Machine Learning

In a mobile network, a handover (also known as a handoff) occurs when a mobile device transitions its connection from one cell to another within the network while maintaining an ongoing communication session. This process is necessary to ensure continuous and seamless connectivity as a mobile device moves from the coverage area of one cell to another.

When the cell ID changes, it signifies that the mobile device has moved from the coverage area of one base station (cell) to another as shown in 9.

One experiment was how changes in *mCellID*, cell identity which is a unique identifier for a cell in the mobile network, can be predicted. The data set contains 3423 unique *mCellID* identifiers, and one plausible target variable is identifying cell ID changes, possibly indicative of handover events. We define a handover variable which is one when *mCellID* changes its value and zero otherwise.

An XGBoost model is trained on the training data excluding *mCellID*, *ChangeIndicator_CellID*, *date*, *mPingHost*, *time*, *mTimeStamp*, *time_second*, *mCellInfoTS* variables. The target variable is *Loss* which indicates when there is packet loss. The classification results on the unseen test data are:

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0	0.83	0.97	0.89	145853
1	0.82	0.45	0.59	51889
Accuracy	0.83 (197742 samples)			
Macro Avg	0.83 (Precision), 0.71 (Recall), 0.74 (F1-Score), 197742			
Weighted Avg	0.83 (Precision), 0.83 (Recall), 0.81 (F1-Score), 197742			

Table 4: Classification Report

In Figure 10, the significance of features on handover prediction is illustrated. The features are arranged according to their importance within the machine learning model's handover prediction. A positive SHAP value implies a tendency towards predicting handover, while a negative SHAP value indicates a prediction leaning towards no need for handover. In Figure 10, as the value of the *mEarcfn* (E-UTRA Absolute Radio Frequency Channel Number in LTE networks, representing the frequency channel used for communication) is very low, the likelihood of handover rises, as indicated by its higher blue shading and placement on the positive side of the SHAP plot.

Figure 11 shows the contribution of mEarcfn, mRsrq, mSpeed, mCqi, mRssnr, mPingAge, mCellSig, and mRsrp.The

For the feature *mEarcfn* (E-UTRA Absolute Radio Frequency Channel Number) in LTE networks, the plot referenced in Figure 11 reveals insightful trends. When 'mEarcfn' assumes negative values, the SHAP value being negative indicates a decrease in the likelihood of handover. This negative SHAP value signifies that as the 'mEarcfn' feature decreases (or takes negative values), it tends to contribute to reducing the probability of a handover occurrence, according to the model's predictive behavior.

Negative values for this parameter might not align with the standard range of values expected for E-ARFCN.

In certain contexts or datasets, negative values for E-ARFCN could potentially denote abnormal or uncommon scenarios, such as data corruption, measurement errors, or specific conditions in the network environment. These unusual or irregular occurrences, represented by negative values, might signify a departure from the typical frequency channel usage or a unique situation within the network.

Conversely, for positive values of 'mEarcfn,' the SHAP value hovering around zero suggests that the model does not heavily rely on this feature alone to make predictions regarding handover events. A SHAP value around zero indicates that 'mEarcfn' in isolation might not decisively influence the model's prediction

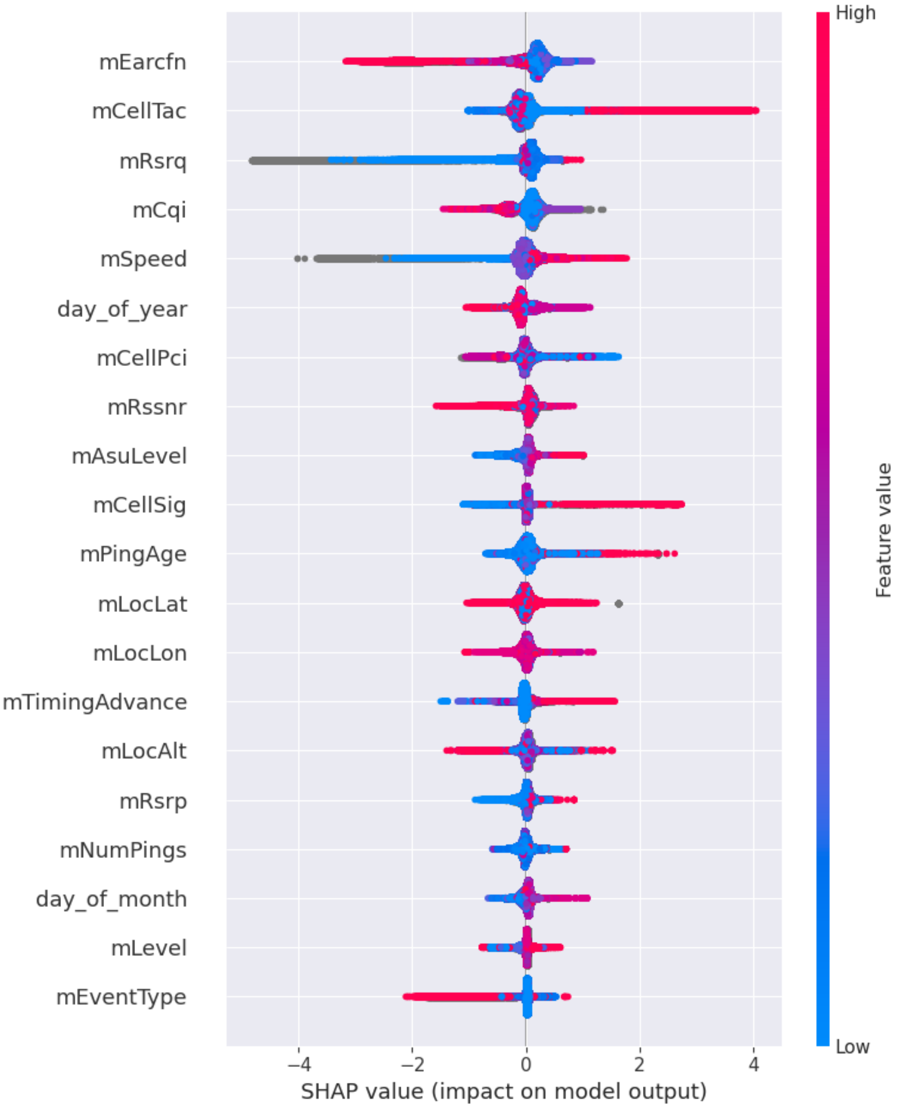


Figure 10: Most important features and their contribution to the handover prediction

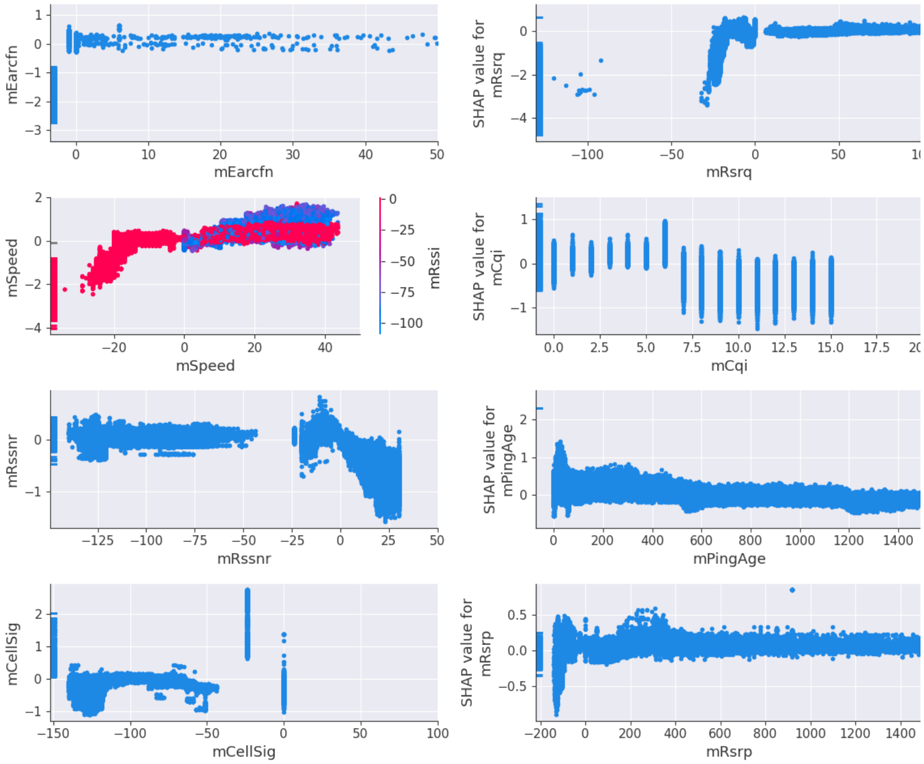


Figure 11: Contribution of features to the handover prediction. Positive Shap value means prediction toward handover and negative Shap value means there is no handover.

of handover events. Therefore, the model might consider other features or a combination of factors alongside 'mEarcfn' to accurately predict or determine the occurrence of handovers in LTE networks.

For *mRsrq*, (Reference Signal Received Quality, indicating the quality of reference signals in LTE networks.) Figure 11 shows that when the signal quality is less than -20 dBm the probability of handover decreases. this might be when the handover already occurred the received signal is from the new and nearby cell ID, therefore the quality of the signal is high.

For *mSpeed*, as the speed increases, there's a greater likelihood of handover, yet when the signal strength remains high even at elevated speeds, handover becomes unnecessary.

Regarding *mCqi*, a lower channel quality corresponds to an increased likelihood of handover.

When *mRssnr* (Signal to Noise Ratio) is high, denoting a stronger signal-to-noise ratio, the necessity for handover diminishes.

The model's prediction using *mPingAge* for handover is intricate and not straightforward.

For *mCellSig* (Cell Signal Strength) falling within the range of good signal strength (-140 dBm to -50 dBm), handover is unnecessary.

2.2.5 Anomaly Detection

Detecting anomalies in mobile network data involves various techniques and approaches based on the nature of the data and the specific anomalies you are trying to identify. We can calculate summary statistics (mean, median, standard deviation) for numerical fields like signal strength (mRsrp, mRsrq, mRssi, etc.). Anomalies could be values significantly outside the expected range. We can also use histograms, box plots, or density plots to visualize the distribution of numerical fields. Outliers or unexpected patterns might indicate anomalies.

Isolation Forest (iForest) [4] is an anomaly detection algorithm based on the principle of isolating anomalies in a dataset by using random forests. Here we do not have any label for the target anomalies in the data, hence it is an unsupervised learning problem.

Illustrated in Figure 12, a majority of the data receive a high anomaly score, indicating they are considered normal. Conversely, when the anomaly score is negative and low, it indicates anomalous data. We designate data as anomalous based on the threshold for the anomaly score (-0.15).

Given the unsupervised nature of this machine learning problem, the performance of our anomaly detection model is not assured. As a preliminary test, we verify if the labeled anomalous data coincides with instances when the mobile connection is lost. This comparison involves examining the distribution of 'mCellConn' values between subsets of anomalous and normal data. We aim to confirm whether anomalous instances have a mobile connection turned off, while normal instances display active cell connections during data streaming.

The kernel density plot as shown in Figure 13 demonstrates that the anomaly data samples display a notable peak when the mobile cell connection is at 0.

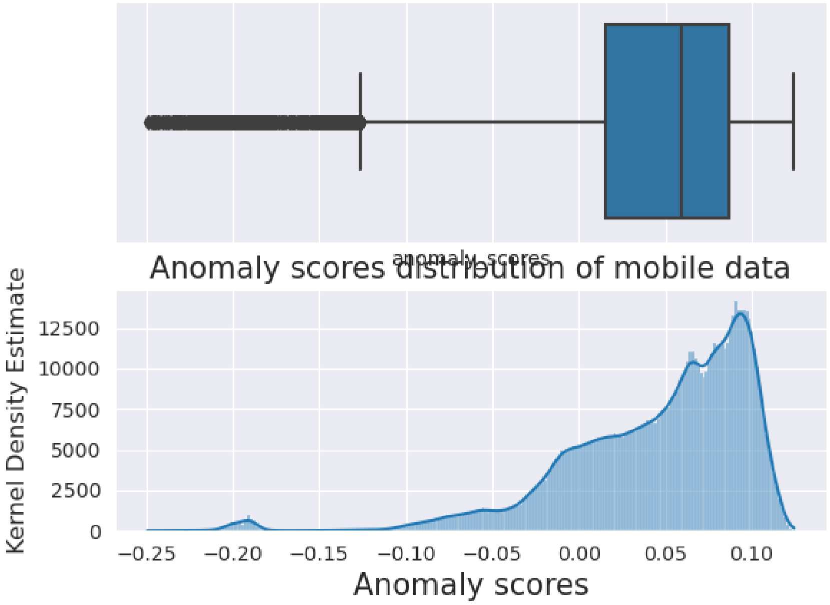


Figure 12: Anomaly distribution in mobile data

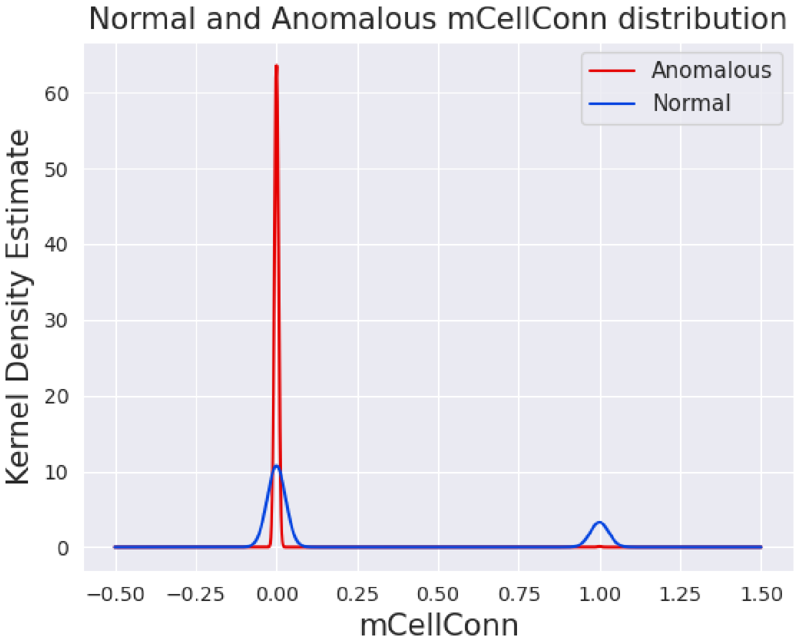


Figure 13: Normal and Anomalous data distribution

Conversely, for normal data samples, a higher peak is observed when the cell is connected (mCellConn = 1) compared to the anomalous data points. This alignment with our expectations supports the anticipated results.

We further check the correlation between anomaly scores and mobile data features. Highly correlated features with 'Anomaly Score' are:

Feature	**Correlation**
DEVICE_INFO_android_id	0.7
DEVICE_INFO_brand	0.6
DEVICE_INFO_buildtime	0.6
DEVICE_INFO_device	0.6
DEVICE_INFO_display	0.5
DEVICE_INFO_fingerprint	0.6
DEVICE_INFO_id	0.6
DEVICE_INFO_manufacturer	0.7
DEVICE_INFO_model	0.6
DEVICE_INFO_product	0.6
DEVICE_INFO_user	0.5
mBW	0.7
mSsRsrq	0.5

Table 5: Correlation Values with Anomaly Score

These correlation values provide insights into the degree and direction of the relationship between each listed feature and the 'Anomaly Score,' assisting in understanding which features are more strongly aligned with anomalies in the dataset. Features related to device information are important in anomaly detection.

2.3 Estimation of Cellular Tower Locations

Another interesting prospect for the given data set with 5G measurements is to estimate the coordinates of cellular towers. Due to the lack of ground truth, i.e., where the towers are located, we considered heuristics as the most suitable approach for this task. The following features were considered to be relevant for this task:

- *mCellID* - Cell Identity, a unique identifier for a cell in the mobile network.
- *mCellPci* - Physical Cell Identity, a unique identifier for a cell in LTE networks.
- *mCellSig* - Cell Signal Strength, indicating the strength of the signal from the cell tower.
- *mLocLon* - Location longitude, indicating the east-west position.
- *mLocLat* - Location latitude, indicating the north-south position.

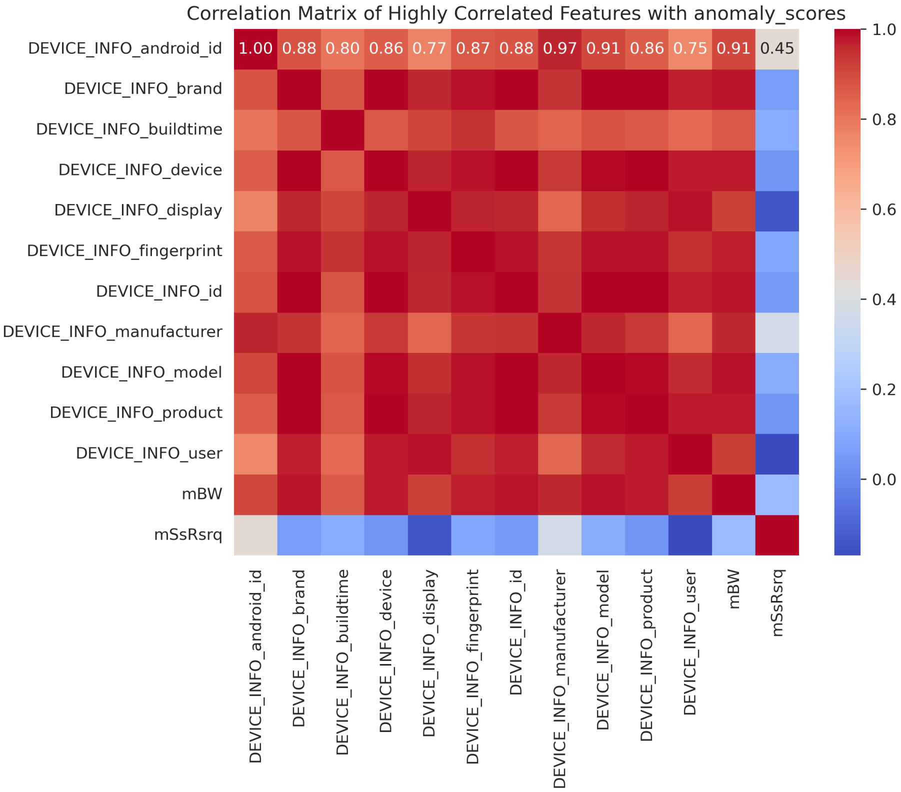


Figure 14: Correlation matrix of Anomaly

- *mPingAvg* - Average ping latency of the measurement.

We clustered the measurements based on *mCellID* or *mCellPci* and then calculated the average coordinate for every data point. We also added a variant that calculates weighted averages, where measurements with lower latency (*mPingAvg*) and better signal (higher *mCellSig*) are given higher weights according to this formula:

$$\bar{Lat} = \frac{\sum(Lat \times Signal \times (1/Ping))}{\sum(Signal \times (1/Ping))}$$

Note that the same formula but with longitude instead of latitude was also used. The estimations are then visualized as an interactive map that can be explored in a web browser, as shown in Figure 15.

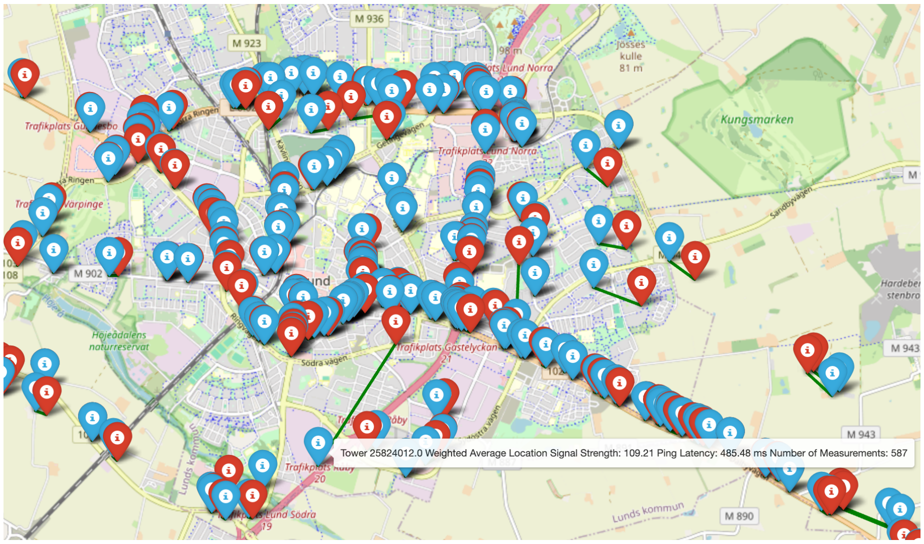


Figure 15: Cell tower estimations map. The locations in red are plain averages while the ones in blue are weighted averages.

2.4 Data Pruning and Statistical Analysis

As mentioned in the introduction, the data was collected from real measurements and the supplied file appears to be made out of several concatenated comma separated value files. Due to additional columns appearing and their order changing, the data had to be reshaped. In addition to reshaping, the data can be pruned. Multiple fields, such as the application build information, can be deemed irrelevant and could be removed for a more manageable data file. A Julia notebook is available, in the supplementary material, that prunes the data.

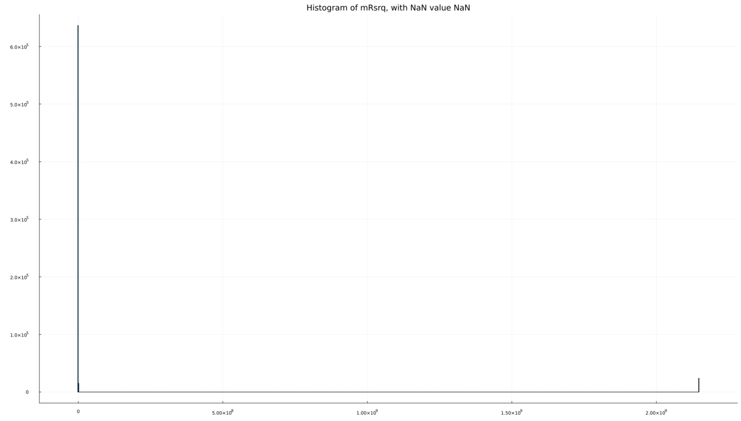


Figure 16: Histogram of *mRsrq*, with NaN values kept.

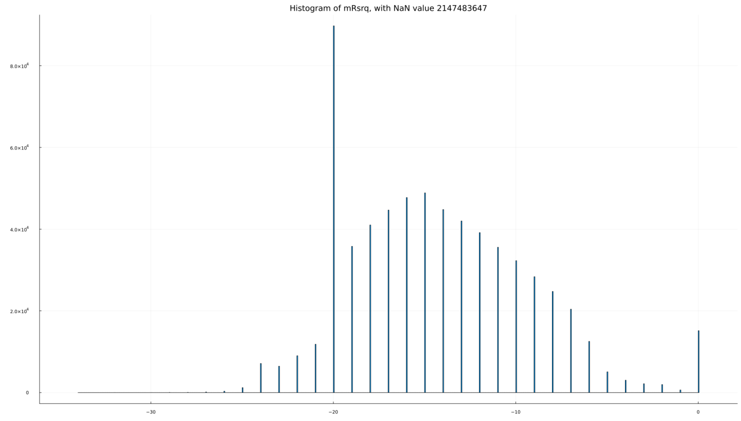


Figure 17: Histogram of *mRsrq*, with NaN values removed.

After being pruned, the data could be explored. The histogram of feature *mRsrq* is shown in Figure 16 and an immediate observation is that something is wrong; the received quality cannot be over 2 million dB. All values of 2147483647, the maximum value of a signed 32-bit integer, seem to represent missing data. A histogram with these values removed is shown in Figure 17, showing a more reasonable distribution. Around -20 dB there seems to be an abnormal amount of data, the cause is unknown.

The Pearson correlation coefficient is a normalized measure of correlation and used to here interchangeably with correlation. Correlation between two different fields across all time gave no interesting results, so how it changes over time was investigated instead. In Figure 18, the correlation between *mPingAvg* and *mRsrq* over time is shown. A sliding window of 10,000 samples is used, and the color represents the time interval in said window. The darker colors mean

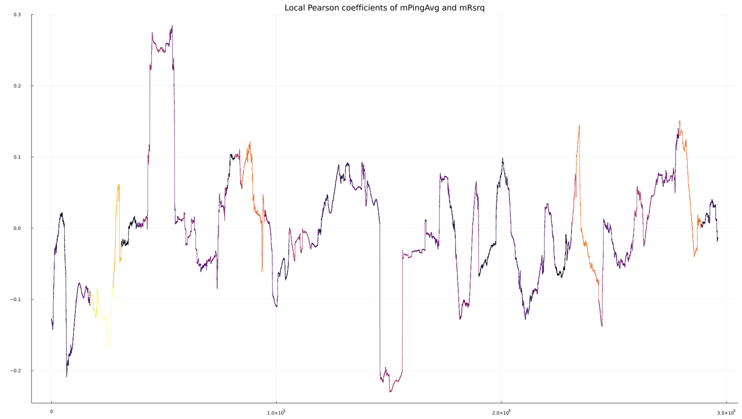
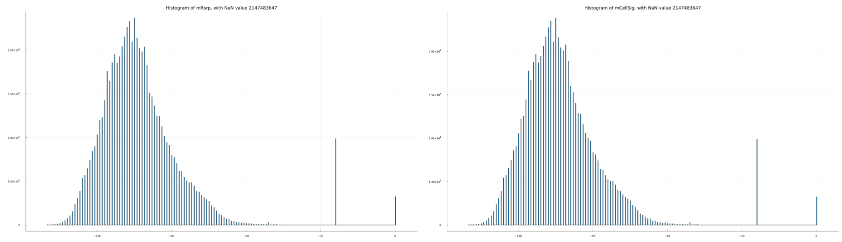


Figure 18: Pearson correlation coefficient between average ping *mPingAvg* and reference signal received quality *mRsrq*, with a sliding window length of 10,000 samples. A brighter color corresponds to a larger time difference between first and last data points, as such the darker colors represent data more relevant for short term usage.



(a) *mRsrp*

(b) *mCellSig*

Figure 19: The properties *mRsrp* and *mCellSig* have nearly identical histograms.

a shorter total time interval, i.e., no break between data collection sessions. It would be expected to be negative as lower signal quality should correspond to a higher ping. There are two interesting plateaus around samples 50,000 and 150,000, however the time interval reveals that these comprise of data from different collection sessions.

A Julia notebook for creating both the histograms and correlation plots is available in the supplementary material.

3 Evaluation / Conclusion

Looking at distribution of data and the visualizer website makes it clear that some properties are highly correlated. For example, *mRsrp* and *mCellSig* have almost identical distributions as shown in Figure 19, and when one of them is high, the other one is too. An interesting distinction between the two is how in *mRsrp*, there are 18227 occurrences of the NaN value 2,147,483,647, whereas in *mCellSig* there are only 525. If one wanted to detect the signal quality but not waste CPU cycles (and battery) collecting unnecessary data, then we would suggest not sampling properties that are so similar.

Looking at *mCellID* values, we can also see that there are more base stations inside a city than any of us had anticipated. This means that as you sit on a bus and surf on your phone, the phone must switch base station quite rapidly. It is impressive that it this happens so transparently to the end user. However, predicting the exact locations of the base stations and cell towers proved to be a tricky challenge. In the absence of the ground truth, heuristics such as assuming that a stronger cell signal and lower ping latency imply more accurate location was used. From the generated map, it could be seen that using the weighted averages was useful. For instance, using plain averages would result in some towers being estimated to be located on the water, while the weighted average would instead estimate them to be on land. Similarly, the weighted averages also tend to make reasonable estimations, such as being closer to popular areas of cities and near main roads, compared to plain averages. Another interesting observation is that the plain and the weighted average coordinates can vary significantly for some towers. This could indicate that for specific cell towers, the devices remain connected for longer compared to other towers that are switched between more frequently.

The AI/ML analysis showed some more useful results. One important finding was which properties are correlated with signal quality issues. See Section 2 for more information. These findings could be used to determine error correction bits for polar codes, as mentioned in Section1.3.

The data preprocessing phase involved converting the timestamp data from nanoseconds to seconds for better comprehension. Although the dataset spans six months, prolonged periods lack any recordings. Continuously observed data instances are sparse, usually lasting only minutes or hours within the entire six-month duration. Feature engineering was undertaken to derive new columns based on diverse time-related attributes and specific conditions within the dataset. 'ChangeIndicator CellID' column was generated to track variations in the 'mCellID' column, aiding in identifying transitions in cell IDs. Another feature, 'Loss,' was generated to indicate network packet loss.

In exploring the 'mCqi' field indicating Channel Quality Indicator, the presence of the maximum signed 32-bit integer value ('2,147,483,647') was noted frequently, implying errors, overflows, or missing data in the dataset. A target variable 'Error' was defined as 1 when 'mCqi' contained this maximum value and 0 otherwise.

Subsequent analysis uncovered highly correlated features associated with the 'Error' target variable. Features like 'app,' 'build,' 'type,' 'board,' 'bootloader,' 'hardware,' 'host,' 'mCellConn,' 'mCellID,' 'mCellPci,' 'mCellTac,' 'mCqi,' 'mEarcfn,' and 'mRssnr' exhibited strong correlations with the occurrence of communication errors indicated by 'mCqi' overflow conditions. These features encompassed diverse aspects such as device information, connectivity metrics, and cell identification.

An XGBoost model trained on specific features achieved impressive classification results for error prediction. Visualization of feature importance illustrated the significance of predictors in error prediction. Notably, certain features like 'mRssi' and 'mTimingAdvance' exhibited predictive patterns. The outlier 'mRssi' values (outside -130db to 0 db) and very low values (lower than -140 db) correlate positively with channel quality decline. 'mTimingAdvance,' indicating synchronization timing, showcases a decline in channel quality for higher values, suggesting potential issues like signal interference or synchronization problems. 'mPingAge's also related with channel quality. Lower signal strengths in 'mSsRsrp'(below -140) align with expected low channel quality. However, within the -140 to -60 range, predictions of good channel quality and no error seem reasonable, corresponding to robust signal strengths associated with better channel conditions. Moreover, 'ChangeIndicator CellID' aligns with expectations: during channel quality errors, alterations in cell IDs reflect a positive change indicator, consistent with anticipated behavior.

Another prediction task involved packet loss using an XGBoost model trained on specific features. The model demonstrated reliable classification results for predicting packet loss. Furthermore, an experiment aimed to predict changes in 'mCellID,' potentially indicating handover events. An XGBoost model was trained, achieving notable predictive performance in identifying cell ID changes. These analyses, employing machine learning models and feature importance assessment, provided substantial insights into channel quality, packet loss, and handover prediction based on the dataset's features.

Detecting anomalies within mobile network data was implemented using Isolation Forest model. To assess the anomaly detection model's performance, a comparison is made between labeled anomalous data and instances of mobile connection loss ('mCellConn'). The kernel density plot illustrates a clear distinction between normal and anomalous data, depicting a significant peak for anomalous data when the mobile cell connection is inactive (0), contrasting with normal data displaying higher peaks when the cell is connected (mCellConn = 1).

Examining the correlation between anomaly scores and mobile data features unveils highly correlated features with the 'Anomaly Score.' Notably, features associated with device information emerge as crucial in the anomaly detection process.

4 Acknowledgements

This work was performed as part of the 2023 instance of the WASP Project Course. We'd like to thank Ericsson for providing the dataset this work was based on. We'd also like to thank Johan Eker and Karl-Erik Årzén for supervising the work.

References

[1] E. Arikan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[2] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16. New York, NY, USA: ACM, 2016, pp. 785–794. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785

[3] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[4] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *2008 eighth ieee international conference on data mining.* IEEE, 2008, pp. 413–422.