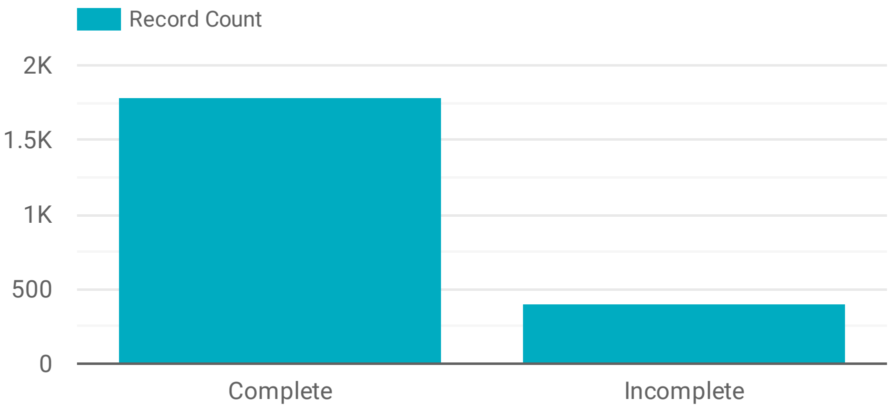
Analysis of Gateway test data(GPS data)

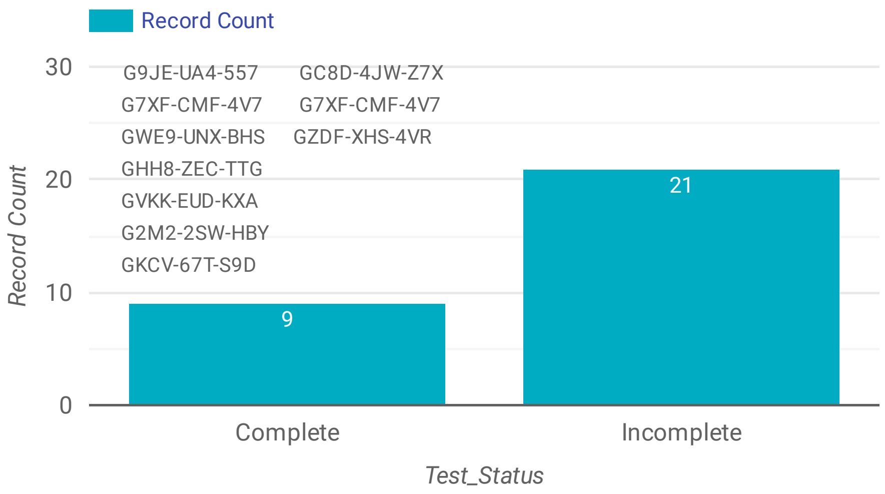
The below analysis is about finding the issues with gateway devices manufactured by Samsara. The gateway devices are tested by testing the quality check of the incoming GPS data. In a GPS system, data is sent from satellites (at least 12 satellites are connected at a time). The below analysis takes into account two important factors of the GPS system: TTFF (time to first fix) and SNR (signal to noise ratio) which are also important to determine whether a gateway is working properly or not.

Total number of non missing (Complete Tests) Vs missing (Incomplete Tests) values across SNR (Graph 1)



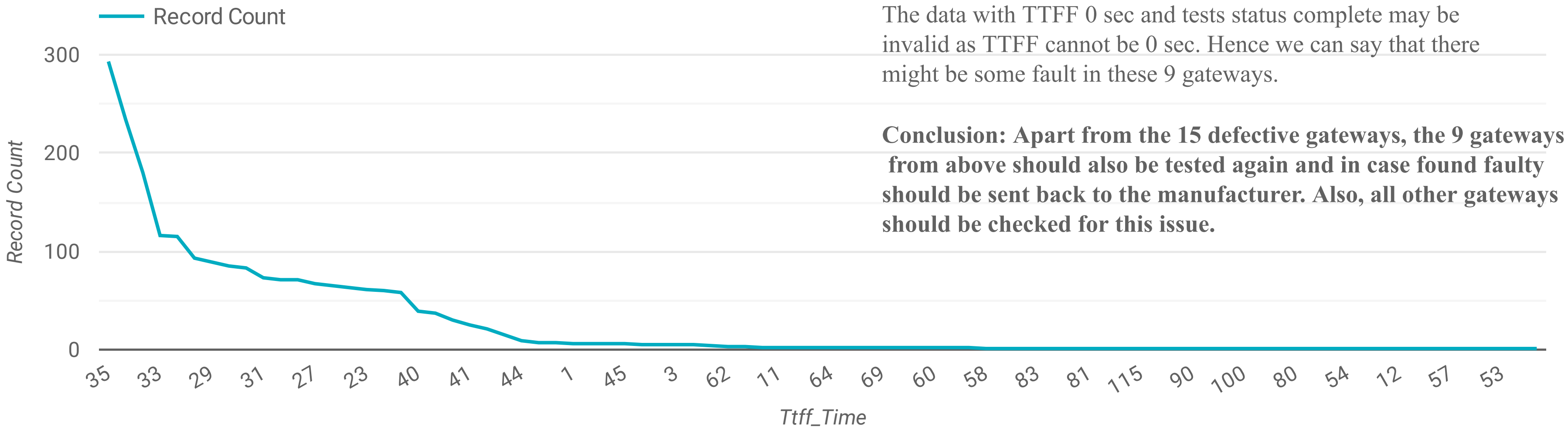
As seen from the above graph 1, 1779 tests are completed tests and 401 incomplete tests(have missing SNR values) from the data.

Total number of Complete vs Incomplete Test whose TTFF value is 0 sec (Graph 3)



As seen from the above graph 3, the data has 9 observations with TTFF 0 sec and where tests were complete and 21 observations with TTFF 0 sec and where tests were incomplete. The data with TTFF 0 sec and tests status complete may be invalid as TTFF cannot be 0 sec. Hence we can say that there might be some fault in these 9 gateways.

Conclusion: Apart from the 15 defective gateways, the 9 gateways from above should also be tested again and in case found faulty should be sent back to the manufacturer. Also, all other gateways should be checked for this issue.



TTFF time of all observations (Graph 2)

TTFF(Time to first fix) is the time a measure of the time required for a GPS navigation device to acquire satellite signals and navigation data, and calculate a position solution (called a fix).

As seen from the above graph 2, most of the tests had 35 sec of TTFF and the average of TTFF is 37.4 sec. Every GPS device to be tested has 3 types of starts: Hot, warm and cold. A hot start can have TTFF from 0.5 to 20 sec, warm start can have minimum TTFF of 30 sec and Cold start has the highest TTFF of all which can last up to 12.5 min.

Path of the average of SNR test throughout the data transmission (Graph 4)

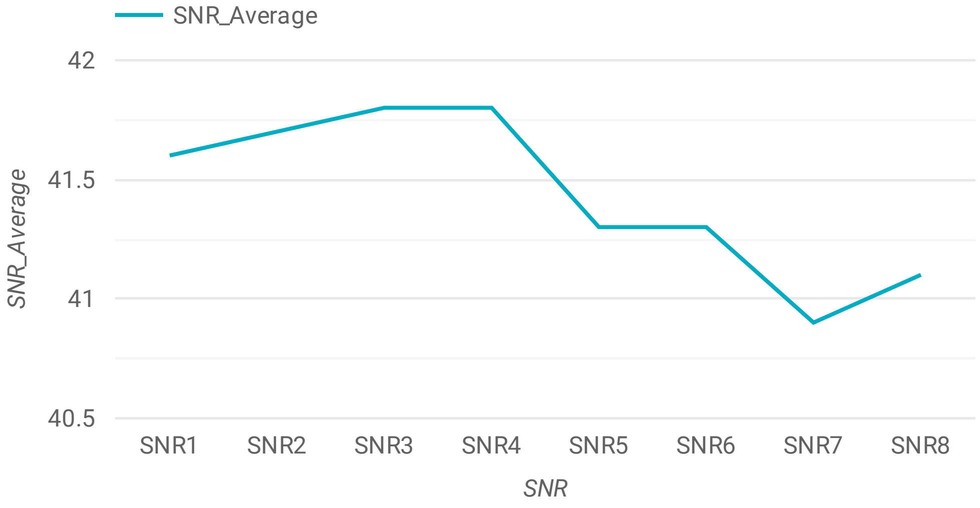


Table of the average of all SNR tests

	SNR ▲	SNR_Average
1.	SNR1	41.6
2.	SNR2	41.7
3.	SNR3	41.8
4.	SNR4	41.8
5.	SNR5	41.3
6.	SNR6	41.3

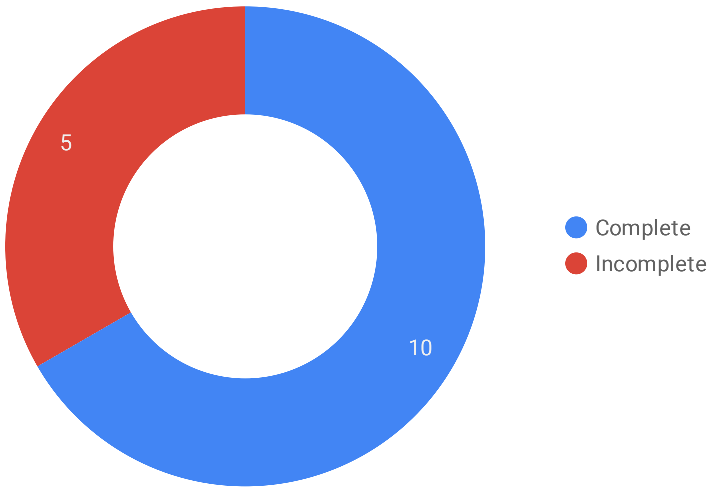
1 - 8 / 8 < >

Signal to noise ratio (SNR) is the strength of the incoming signal. This ratio is higher the better because that means that this has less noise.

As seen from the above graph 4, we can see that the average SNR decreases, but it does not change much during the transmission of the data. This means that the SNR value is stable and the quality or strength of the signal does not change much throughout its transmission period.

The mean SNR in the data is around 41 units, which is appropriate (strong) signal strength.

Complete Vs Incomplete Tests of Defective observations(Graph 5)

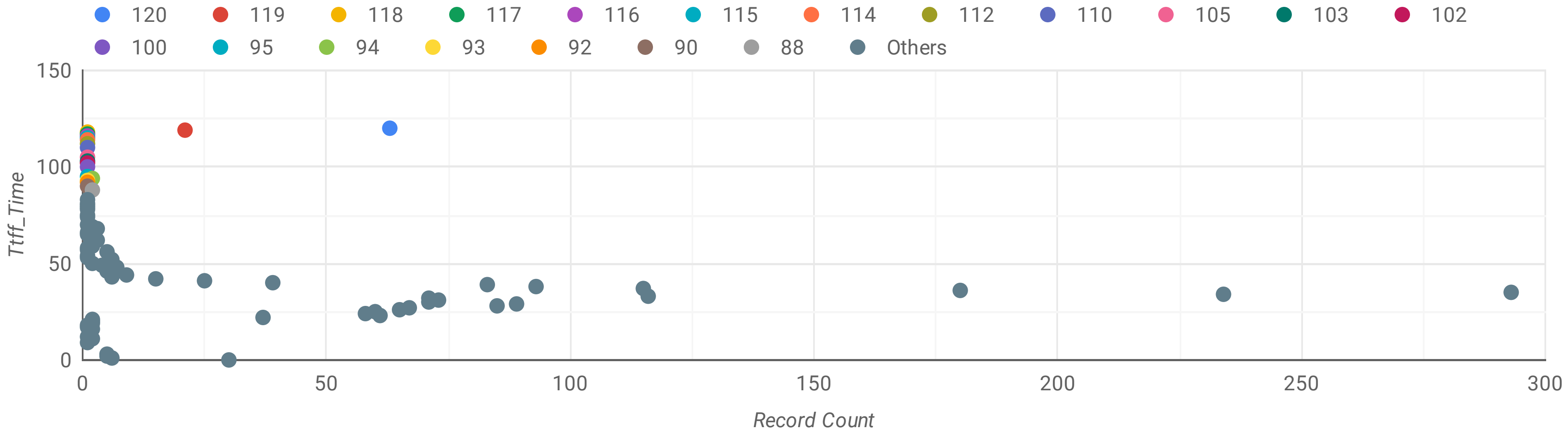


As seen from the pie chart 5, 5 out of 15 defective pieces have incomplete tests(missing SNR), this may mean that the tests were not conducted properly. These 5 gateways need to be rechecked by conducting complete tests.

Also, a pattern has been observed in the data that most of the SNR missing are in a group i.e either the first 4 are missing together in a test or the last 4 are missing together. This pattern needs to be checked further.

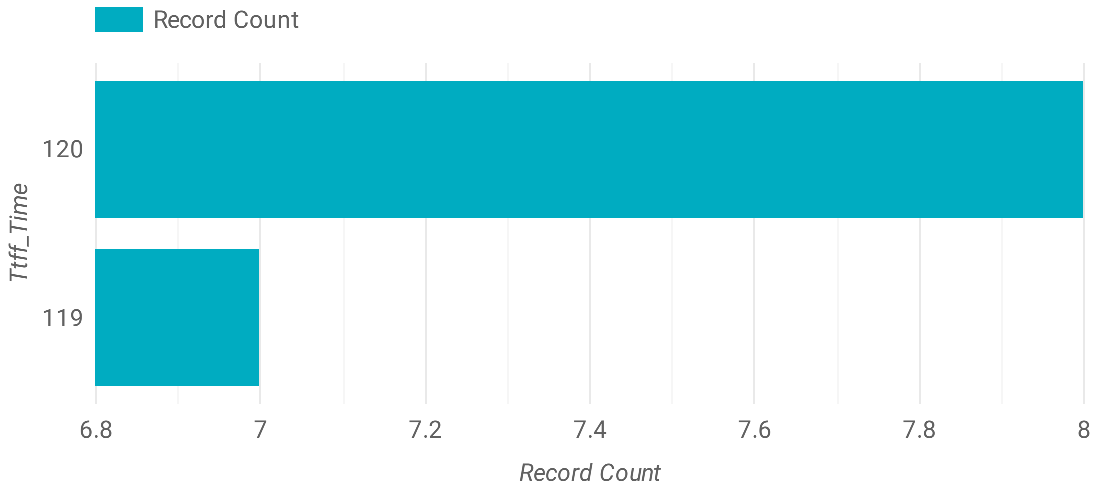
Also, the average SNR for the defective pieces is stable throughout the transmission of data.

TTFF of all records Vs Count of Observations (Graph 6)



As seen from the above graph 6, approximately 4 % of observations lie above 100 sec of TTFF. This may be because of the type of gateway start (cold start) or abnormality in the data.

TTFF of defective gateway data (Graph 7)



Conclusion: Hence, the gateways which have TTFF higher than 100 sec should be checked and tested for any abnormalities and if they are functioning correctly.
Also, all the 15 defective pieces are present in LI5-1834411M unit. All the gateways available for this unit should be rechecked and the manufacturer must be notified about the same.

The defective gateways have TTFF ranging from 119 sec and 120 sec as seen from the above graph 7 showing the TTFF for all the defective gateways.

A way to test invalid TTFF values is to check if any value is 10 times the mean TTFF value.
Mean value ~ 37 sec, hence an invalid value would be TTFF around 370.
The maximum value of TTFF in the data is 120 sec, which is not near to 370 sec, hence we can say that there no other invalid TTFF outliers in the data.

Annexture:

Python code to convert all log files into a CSV file(Jupyter notebook attached in the mail):

```
import os
import numpy as np
import pandas as pd
pd.set_option('display.max_columns', 30)
def clean_df(df):
    cleaned_df = pd.DataFrame(index=np.arange(1), columns=np.arange(15))
    cleaned_df.columns = ['Date', 'Input_Sn', 'Detect_Ok_Time',
                          'Gps_Use_Time', 'Ttff_Time', 'SNR1',
                          'SNR2', 'SNR3', 'SNR4', 'SNR5', 'SNR6',
                          'SNR7', 'SNR8', 'Pass_Test_Time', 'Total_Time']
    date = np.nan
    input_sn = np.nan
    detect_ok_time = np.nan
    gps_use_time = np.nan
    ttff_time = np.nan
    snr1 = np.nan
    snr2 = np.nan
    snr3 = np.nan
    snr4 = np.nan
    snr5 = np.nan
    snr6 = np.nan
    snr7 = np.nan
    snr8 = np.nan
    pass_test_time = np.nan
    total_time = np.nan
    for value in df.values[0]:
        if value.find('\t') != -1:
            date, input_sn = value.split('\t')
            input_sn = input_sn.split(':')[1]
        elif value.find('Detect OK!') != -1:
            detect_ok_time = value.split('=')[1]
        elif value.find('Enable GPS Module OK!') != -1:
            gps_use_time = value.split('=')[1]
        elif value.find('TTFF') != -1:
            ttff_time = value.split('=')[1]
        elif value.find('SNR1') != -1:
            snr1 = value.split('=')[1]
        elif value.find('SNR2') != -1:
            snr2 = value.split('=')[1]
        elif value.find('SNR3') != -1:
            snr3 = value.split('=')[1]
        elif value.find('SNR4') != -1:
            snr4 = value.split('=')[1]
        elif value.find('SNR5') != -1:
            snr5 = value.split('=')[1]
        elif value.find('SNR6') != -1:
            snr6 = value.split('=')[1]
        elif value.find('SNR7') != -1:
            snr7 = value.split('=')[1]
        elif value.find('SNR8') != -1:
            snr8 = value.split('=')[1]
        elif value.find('GPS Signal Test PASS!') != -1:
            pass_test_time = value.split('=')[1]
        elif value.find('Total Time') != -1:
            total_time = value.split(' ')[2]
    cleaned_df['Date'] = date
    cleaned_df['Input_Sn'] = input_sn
    cleaned_df['Detect_Ok_Time'] = detect_ok_time
    cleaned_df['Gps_Use_Time'] = gps_use_time
    cleaned_df['Ttff_Time'] = ttff_time
    cleaned_df['SNR1'] = snr1
    cleaned_df['SNR2'] = snr2
    cleaned_df['SNR3'] = snr3
    cleaned_df['SNR4'] = snr4
    cleaned_df['SNR5'] = snr5
    cleaned_df['SNR6'] = snr6
    cleaned_df['SNR7'] = snr7
    cleaned_df['SNR8'] = snr8
    cleaned_df['Pass_Test_Time'] = pass_test_time
    cleaned_df['Total_Time'] = total_time
    return cleaned_df
all_df = []
for root, dirs, files in os.walk('GPS Test Data/'):
    for file in files:
        if file.endswith(".txt"):
            path_file = os.path.join(root,file)
            file_df = pd.read_csv(path_file, index_col=None,
header=0).T.reset_index()
            file_df = clean_df(file_df)
            all_df.append(file_df)
df = pd.concat(all_df, axis=0, ignore_index=True)
df.to_csv('Clean.csv', index=False)
```