**Summary Document: FWA Time Series Forecasting Model**

1. **Overview:**

* The FWA Time Series Forecasting (TSF) model helps to predict how much internet data a user will use in the future on Fixed Wireless Access (FWA) devices.
* It uses a smart built-in forecasting method called ARIMA+, available directly in Google BigQuery. This means we can do all the predictions using simple SQL queries.
* The model predicts daily data usage for each user (IMSI) and helps to identify early signs of slow internet speed, especially when the user’s connection changes from high-speed (QCI 8) to lower-speed (QCI 9) — which usually happens when users use too much data.

1. **Why Forecasting is Needed?**

* To predict monthly data usage trends.
* Identify network throttling risks.
* Help the AIOps Insight Platform manage resources and alert potential performance drops.
* Specifically detect when a user’s QoS shifts from high priority (QCI 8) to low priority (QCI 9).

**3. Model Use Case**

The model forecasts future daily traffic for each FWA device (IMSI) to:

* Detect QCI transitions.
* Improve customer experience by avoiding unexpected slowdowns.
* Enable better capacity planning and network optimization.
  1. **Methodology – End-to-End Flow**

**Step 1: Raw Data Aggregation**

* Pull from from\_mobile\_bytes and to\_mobile\_bytes fields from SDR logs.
* Aggregate daily total traffic for each user (IMSI).

**Step 2: Cycle Start & Cumulative Calculation**

* Identify plan cycle start and end dates.
* Calculate cumulative usage per cycle and per QCI level.

**Step 3: Model Training (ARIMA+)**

* Train model using 120 days of historical traffic.
* Model built in SQL using BigQuery’s ML.FORECAST function.

**Step 4: Forecast Generation**

* Predict 30 days of future traffic per IMSI.

**Step 5: Post-Processing & QCI Throttling Detection**

* Merge historical + forecasted data.
* Calculate cumulative usage in future month.
* Compare with historical QCI threshold to predict throttling day.

**Flow diagram:**

**Predict QCI 8 → QCI 9 Switch Date**

- Identify When Throttling May Happen │

**Combine Historical + Forecast Data**

- Create Continuous Usage Timeline **│**

**Feature Engineering using SQL** - Identify Monthly Plan Cycle - Calculate Cumulative Traffic

**Create Forecasting Model (ARIMA+)**

- tsfModel in BigQuery

- Forecast Horizon: 30 Days

**Compare with Throttling Threshold**

- Derived from Previous QCI Change

**Raw Traffic Data (from GCP SDR) (Fields:** from\_mobile\_bytes, to\_mobile\_bytes)

**Calculate Monthly Cumulative Usage**

(Historical + Forecasted Traffic)

**Daily Aggregation Process** (IMSI-wise Total Traffic per Day)

**Generate Forecast using ML.FORECAST**

- Daily Forecasted Usage per IMSI

* 1. **Model Architecture and Key Parameters**

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Model Type | ARIMA\_PLUS (in BigQuery) |
| Forecast Horizon | 30 Days |
| Confidence Level | 95% |
| Data Frequency | Daily |
| Max Time Series Length | 60 Days |
| Clean Spikes/Dips | Yes |
| Holiday Region | US |
| Decompose Time Series | Yes |

* 1. **Input & Output Parameters**
* **Input Table (Aggregated Data):**

|  |  |
| --- | --- |
| **Name** | **Description** |
| servedimsi | User Identifier (IMSI) |
| date | Daily Data Timestamp |
| totalTraffic | Sum of data usage (upload + download) |

* **Output Table (Forecasted Data):**

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| forecast\_timestamp | Date of Forecast |
| forecast\_value | Predicted Data Usage |
| standard\_error | Error in Forecast |
| prediction\_interval\_lower\_bound | Lower bound of forecast range |
| prediction\_interval\_upper\_bound | Upper bound of forecast range |
| confidence\_interval\_lower\_bound | Lower bound of confidence interval |
| confidence\_interval\_upper\_bound | Upper bound of confidence interval |

* 1. **Model Training and Forecasting**

**Create Model SQL:**

CREATE OR REPLACE MODEL `tsfModel`

OPTIONS (

model\_type='ARIMA\_PLUS',

time\_series\_timestamp\_col='date',

time\_series\_data\_col='totalTraffic',

time\_series\_id\_col=['servedimsi'],

auto\_arima=True,

data\_frequency='DAILY',

max\_time\_series\_length=60,

horizon=30,

auto\_arima\_max\_order=5,

clean\_spikes\_and\_dips=True,

holiday\_region='US',

adjust\_step\_changes=True,

decompose\_time\_series=True

)

AS

SELECT servedimsi, date, totalTraffic FROM `aggregatedTable`;

**Forecast SQL:**

SELECT \* FROM ML.FORECAST( MODEL `tsfModel`, STRUCT(30 AS horizon, 0.95 AS = confidence\_level));

**Post-Processing – Predict QCI Change**

**Step 1: Detect Historical QCI Threshold**

* Analyze when QCI changed from 8 → 9 in past cycles.
* Identify cumulative traffic at that point (threshold).

**Step 2: Merge Historical + Forecast Data**

* Create a single table with both real and forecasted values.

**Step 3: Calculate Monthly Cumulative Forecast**

* Calculate total monthly usage (historical + predicted).
* Compare it with the historical QCI threshold to predict the next throttling event.

**-------------\*\*-------------**