The Bengaluru Last Mile Challenge, 2025

Description of the datasets

This document describes the various datasets that have been made available for the hackathon.

Some general remarks:

The live location data tends to be noisy with some null, incorrect or missing values. Other static data such as details of stops and routes may be incomplete or even inconsistent in some cases. You may choose to ignore such errors or infer/impute the probable values based on heuristics. In any case, there is a sufficient amount of good data to find a reasonable solution for the given tasks.

1. stops.csv

This file provides a list of BMTC bus stops in the transit system. Each row represents a physical location where a vehicle stops. The following table shows the fields in **stops.csv**.

The file contains 12,274 unique stop_id values and corresponding stop_names and location information. Note that, many of these stop_ids may not occur in route_to_stop_sequence.csv.

columns in stops.csv:

Field	Description
stop_id	Unique identifier for the stop
stop_lat	Latitude of the stop
stop_lon	Longitude of the stop
stop_name	Name of the bus stop

Table. Description of fields of stops.csv

Example :

stop_id	stop_lat	stop_lon	stop_name
22482	13.09943	77.58817	Sheshadripuram College
33142	12.76839	77.77011	Anekal Road Junction
33020	13.24270	77.82258	I Basapura

2. routes.csv

This file provides information about each transit route in the transportation system. In our specific case, the route in which the bus is travelling is specified by the route_id. Please note that, in general, there will be several trips during the day for a given route. Each such trip will have a different trip_Id, but they will follow the same route. You will find the route_id in the live location data, which specifies the route in which the bus is travelling. This file contains the details of 10,719 unique route_id values. All of these route_ids may not occur in the live data. The following table explains the fields in the **routes.csv**.

columns in routes.csv

Field	Description
agency_id	We set this value to 38
route_desc	Description of the route.
route_id	Unique identifier for the route.
route_long_name	Descriptive name for the route.
route_short_name	Short name (e.g., bus route number)
route_type	Numeric type of transportation (1 = Subway/Metro,3 = Bus)

Table. Description of fields in routes.csv

Examples:

agency_id	route_desc	route_id	route_long_name	route_short_name	route_type
38	Central Silk Board - Yeshawanthapura Bus Station	3040	CSB-YBS	500-DJ	3
38	Sri Ramapura - Kempegowda Bus Station	2676	286-D SRP-KBS	286-D SRP-KBS	3

3. route_to_stop_sequence.csv

This file provides information about the sequence of stop_ids that are traversed in a route_id. It has 7,509 route_ids and corresponding stop_id sequences.

Field	Description
route_id	Unique route identifier (from routes.csv)
stop_id_list	sequence of stop_id s (from stops.csv) seen in the corresponding route_id

Table. Description of fields in route_to_stop_sequence.csv

In the following table we provide two route_ids and the corresponding stop_id sequences associated with it. Note that, the total number of route_ids is 10,179 (routes.csv), but route_to_stop_sequence.csv has 7,509 route_ids and the corresponding stop_id sequences. In your analysis please consider the route_ids for which the stop_id sequence can be obtained from route_to_stop_sequence.csv.

Example:

route_id	stop_id_list
10019	['20647', '20650', '24363', '35428', '21748', '22406', '22264', '22162', '32058', '32059', '22735', '34658', '28331', '32260', '34372']
10020	['34372', '24480', '22733', '26582', '30196', '30052', '30050', '22162', '22265', '22407', '24302', '20646', '20652', '20651', '20649']

Table. Sample rows from route_to_stop_sequence.csv

4. Vehicle live location data

This data displays the current position and route identifier (route_id) of a bus if available. The data is in parquet format. In this data, the most useful fields are the vehicle label (the registration number), latitude, longitude and timestamp fields. The remaining fields may not be updated as frequently as expected. Using the vehicle label and the latitude-longitude series ordered by timestamp, you can trace the bus path. Please note that this data is noisy.

The data provided is for the period July 30, 2025 to August 19, 2025. The data is divided into seven zipped files, each covering three days starting July 30, 2025. The daily data is stored in chunks of six parquet files for convenience. In many cases, the fields in the data packets may have null values and duplicates. You may ignore such data packets or use them after cleansing them appropriately.

Example:

Field	Description
id	Storage identifier, can be disregarded.
system_time	API polling time for internal reference, can be disregarded.

current_status	Operational status of the bus [0: INCOMING_AT, 1: STOPPED_AT, 2: IN_TRANSIT_TO].
stop_id	Th next stop_id the vehicle will be reaching
vehicle_timestamp	Packet transmission time by the vehicle **(this is in the Unix Timestamp use the code below to get the actual time stamp)**
trip_id	Unique identifier for a trip taken by vehicle
start_time	Start time of the trip in HH:MM:SS format.
start_date	Start date of the trip in YYYYMMDD format.
schedule_relationship	Status of the trip schedule [e.g., ADDED, CANCELED, SCHEDULED].
route_id	route_id followed by the vehicle
latitude	latitude
longitude	longitude
bearing	Denotes the GPS Angle of the bus from true north in degrees.
vehicle_id	Unique identifier for the vehicle.
label	Vehicle registration number/license plate number of the bus corresponding to this observation.
date	Used for internal data file sorting, can be disregarded.

IMPORTANT: Please use time_stamp attribute in live data as the data transmitted time from the vehicle. Do not consider the system_time attribute in the live data. Note that this is a Unix timestamp.

##Use the following to get the readable time stamp from Unix to Date time

```
from datetime import datetime

    start_dt = datetime.utcfromtimestamp(timestamp)
```

Note that the above function converts the timestamp to UTC (GMT). You can continue to work with UTC, as long as you are consistently using it.

Example:

Field	Description
id	6060
system_time	2025-07-19T20:00:06.00544

current_status	1
stop_id	38058
vehicle_timestamp	1752955153
trip_id	65945431
start_time	20:00:00
start_date	20250719
schedule_relationship	SCHEDULED
route_id	1890
latitude	12.795304
longitude	77.659775
bearing	22.0
vehicle_id	27067
label	KA51AK0972
date	2025-07-19

Please use vehicle_timestamp is the time at which vehicle shares the live location, route_id, route_id, latitude, longitude are the attributes that are mostly used in this task.

The value current_status = 1 (STOPPED_AT) indicates that the vehicle is at a bus stop. In practice, this does not always mean the vehicle is completely stationary — small changes in latitude/longitude can occur due to GPS drift or noisy observations.

The schedule_relationship field indicates whether a trip is valid for consideration. Data should only be used when schedule_relationship is either ADDED or SCHEDULED. If a trip is marked CANCELLED, it may have stalled or may be returning to the depot for some reason. All data points for that trip_id should be ignored.

The stop_id field indicates whether the bus is stopped or arriving at the stop_id. Please do not rely on this information as it may not be updated frequently. It may also be wrong.

Note:

- There is a trip ID (trip_Id) associated with each trip of a route on any given day. The trip_ids change every day and you may not be able to find the same trip_id in the live-location data of successive days. However, you can correlate the trips across days, albeit approximately, by making use of the route-id and the time of day when the trip is happening.
- You may not observe all the route_ids mentioned in the routes.csv in live data.

- You may observe NaN/null values for route_id in live data intermittently or for a complete day for a vehicle.
- The stop sequence in route_to_stop_sequence.csv represents the official order of stops that vehicles follow. In normal operations, buses adhere strictly to this sequence. Deviations are rare and typically only occur under special circumstances such as road repairs, construction, or other disruptions.

5. Autorickshaw activity and availability data

This zipped dataset consisting of 14 files contains information about autorickshaw rides in Bengaluru over a 14-day period (from July 30, 2025 to August 12, 2025). Each file is stored in parquet format and consists of 385 columns.

Spatial Dimension

- The city of Bangaluru is divided into a hexagonal grid pattern using the **Uber H3 index** at Resolution 7. (Note: At this resolution, each hexagon covers roughly 5.14 sq km.) For more information on the UBER H3 index please refer to https://github.com/uber/h3. H3 provides a consistent way to represent space as hexagons, which makes it easier to compare different parts of the city at the same spatial scale. When combined with **OpenStreetMap (OSM)** data, this grid can be used to link roads, traffic, and activity patterns to specific cells, enabling scalable analysis.

- The first column in the dataset is **h3_index**, which uniquely identifies the hexagon.
- Each row corresponds to one hexagonal region in the city. If no activity is observed in a hexagon, the row corresponding to this hexagon will not appear in the parquet file. Each day's data has 240 - 250 rows, which implies that approximately an area of 1300 sq km is covered. For example, in the following figure we show the coverage of auto rides on August 2, 2025 that has 254 hexagons. Note that, this is not a heat map for the number of auto rides in a hexagon. It simply shows the area covered by the data.

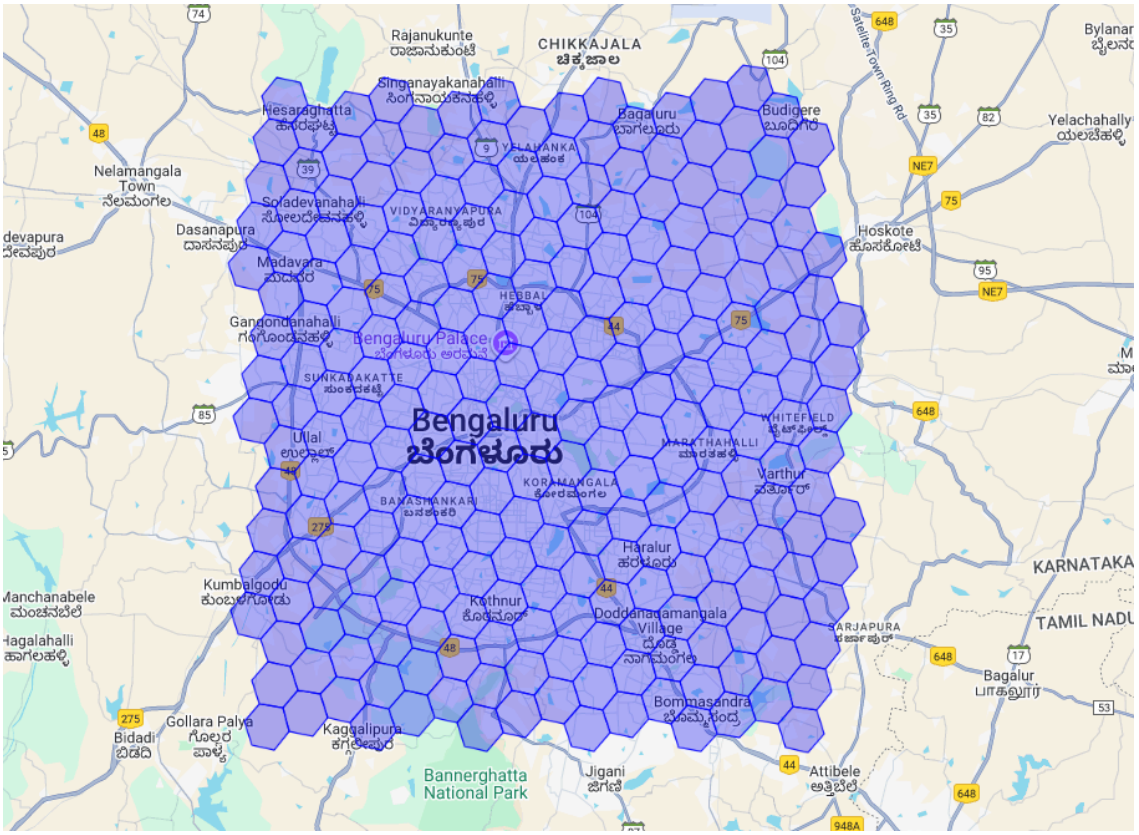


Fig. Autorickshaw ride coverage area on August 2, 2025.

Temporal Dimension

- Any given day (24 hours) is divided into **96 time intervals** of **15 minutes each**.
- The intervals are indexed from **0 to 95**.
    - Example:
        - index 0 corresponds to 00:00 – 00:15
        - index 1 corresponds to 00:15 – 00:30
        - …
        - index 95 corresponds to 23:45 – 24:00

Column Groups

1. *start_0 … start_95*
    - Represents the **number of autorickshaw rides that started in the given hexagon and ended in a different hexagon** (h3_index) during the corresponding time slot.

2. *end_0 … end_95*
    - Represents the **number of autorickshaw rides that started outside the hexagon and ended in the given hexagon** (h3_index) during the corresponding time slot.

3. *start_end_0 … start_end_95*
    - Each cell contains a **tuple (n, t)**:
        - n - **number of auto rides that started and ended within the same hexagon** in that time slot
        - t - average time taken (in minutes) to complete those rides

4. *idle_0 … idle_95*
    - Each cell contains a **list of values**: Each value in the list represents the **duration (in minutes)** that an auto remained **idle** in that hexagon during the given time slot.

Note:

1 Currently, the Namma Yatri has a time out of 3 minutes for each request. If no autorickshaw is available within 3 minutes, it will cancel the trip automatically.

2. Each request is broadcast to drivers that are approximately within 1.84 km, which covers an area of approximately 10 sq km. Therefore, for availability, you will need to take into consideration the availability in the neighbouring hexagons as well.

Files

- There are 14 parquet files, one for each day.
- File naming convention, h3_2025_07_30.parquet corresponds to data for July 30, 2025

Note: In h3_2025_07_30.parquet, **all idle columns contain empty lists**. This is because this data is not available for this day. We may provide this later. In all other parquet files, these columns contain actual idle time values. Also, the data is available from 5:30 AM onwards on July 30, 2025. For the remaining days, the data is available for the whole day.

6. Autorickshaw speed data

This dataset consists of 14 files in parquet format compressed into a single ZIP archive, each representing autorickshaw speed data in kmph on various OSM IDs for a particular day. For example, the file “2025-07-30_speed_onlyosm_speed.parquet” will contain the data for July 30, 2025. Each row in a parquet file contains the speed data for one OSM ID. The first column contains the OSM ID, the second and third columns contain the source and target nodes (u,v) of the road segment from OpenStreetMap. The remaining columns contain the average speed

observed on the OSM ID over 15 minute intervals starting at 00:00 hours (12 AM) on that day and ending at 24:00 hours. There will be 96 such columns, one corresponding to each 15 minute interval.
Each column corresponds to a specific 15-minute time slot, with the column name representing the time slot index. (Please see description of “Autorickshaw activity and availability data” for details on the slot indices.)
Missing values indicate that no speed data was recorded for that interval. Please note that not all OSM IDs in the map will be covered by the speed data. If a particular OSM ID (road segment) was not traversed by any autorickshaw on a particular day, the row corresponding to that OSM ID will be missing in the file.

NOTE: We do not have speed data for the period 00:00 hours - 05:30 hours on July 30, 2025.

The following figure shows a colour-coded map of speeds on road segments of Bengaluru for the 12:00 PM - 12:15 PM slot on Wednesday, July 30, 2025, on the left, and a histogram of speeds for the same time slot on the right.

