

# Improving LM Accuracy

- Steps
- Terminology Used
- Experiments
  - L8 Geohash with only OSM data
    - Steps
    - Results
      - Neighbor level 3
        - Observations
      - Neighbor level 5
    - Iterations
  - L8 Geohash with pings data relayed over OSM data
    - Steps
    - Results
      - Neighbor level 3
        - Observations
      - Neighbor level 5
      - Neighbor level 3 to 5
  - L8 Geohash with only pings data
    - Results
      - Neighbor level 5
      - Neighbor level 3 to 5
  - L7 Geohash with only OSM data
    - Results
      - Neighbor level 5
  - L7 Geohash with pings data relayed over OSM data
    - Results
      - Neighbor level 5
  - L50 Geohash with OSM data
    - Results
- Action Items (Discussed in meeting with Product)
  - Use reached time instead of delivered time
    - L8 Geohash with only OSM data
      - Results
        - Neighbor level 5
        - Neighbor level 3 to 5
        - Neighbor level 4 to 5
        - Neighbor level 2 to 5
        - Neighbor level 5 to 5
        - Neighbor level 4 to 7
    - L8 Geohash with only pings data
      - Results
        - Neighbor level 3 to 5
        - Neighbor level 4 to 7
        - Neighbor level 5 to 7
    - L8 Geohash with pings data relayed over OSM data
      - Results
        - Neighbor level 4 to 7
  - Experiment with Custom Geohash (50mX50m)
    - L50 Geohash with only OSM data
      - Results
        - Neighbor level 5
        - Neighbor level 3 to 5
    - L50 Geohash with only pings data
      - Results
        - Neighbor level 3 to 5
    - L50 Geohash with pings data relayed over OSM data
      - Results
        - Neighbor level 3 to 5
  - Filter PINGS data based on accuracy
    - L8 Geohash with only pings data
      - Neighbor level 4 to 7
  - PINGS data based on time of day
- Relevant Logics in code
  - Last mile time calculation from distance

## Steps

- ☒ For a last 7 days trips that occurred in Koramangala zone, find out OSM time for the corresponding lat-lng and compare it with Lucifer predicted LM and actual LM. 02 Feb 2018
  - Following are the stats for Koramangala zone orders between '2018-01-26 00:00:00' and '2018-02-02 00:00:00'
  - Total: 38620
  - Non-compliant: 9670 (25%)
  - Non-accurate: 23488 (60%) (+5/-5 minutes)
- ☒ For a Koramangala Zone, get all the Geohashes (Custom L50) and for each Geohash, find distance and time given by OSM for all the neighbors and for their neighbors till 4-5 levels. 02 Feb 2018
- ☒ Once this is done, predict LM using A\* algorithm and compare it with the trips that occurred for last 7 days. 05 Feb 2018
- ☐ Build a feedback mechanism to calculate time between 2 geohashes based on the pings that we get from DEs and update the cached entry based on entries
  - ☒ Have a rough time-slots to start with: BreakFast, Lunch, Dinner 05 Feb 2018
  - ☐ Apply aging to give more weight to latest data. **05 Feb 2018**
- ☐ Compare LM with the trips that occurred for last 7 days. **06 Feb 2018**
- ☐ Tune the parameters: aging, number of lat-longs in a Geohash etc to optimize to accuracy. Compare the predicted LM with actual LM and iterate this step by tuning the parameters. **09 Feb 2018**

## Terminology Used

- Accuracy: Prediction is accurate if **(actual\_lm - 5 minutes) < prediction < (actual\_lm + 5 minutes)**
- Compliance: Prediction is compliant if **prediction >= actual\_lm**

## Experiments

### L8 Geohash with only OSM data

#### Steps

- For a Koramangala zone, get all L8 Geohash entries and for each L8 Geohash find neighbors till 3 levels.
- For each neighbor pair, find OSM distance and calculate time from the distance
  - We can't use time given by OSM as it returns time in minutes and for most of the L8 geohash pairs, it would return time as 0 minutes.
  - Use mins\_per\_km\_80\_perc field from sla\_area\_data\_v3 table to calculate time from distance.
- Get all the orders from Koramangala zone for a day.
- For each order, calculate shortest time between restaurant and customer L8 Geohash by using Dijkstra's algorithm provided by [JGraphT](#) library.

## Results

### Neighbor level 3

Total number of orders: 8278

Number of valid orders: 3839

Valid order: Path exists between source to destination

Method	Accurate	Compliant
Existing	2383 (62.1%)	<b>2412 (62.8%)</b>
New	<b>2429 (63.3%)</b>	1173 (30.1%)

### Observations

- Compliance is affected by a great margin. That means predicted LM is less than the actual value for most of the times.
- For few orders, graph is not reachable. That has reduced the number of valid orders from 8278 to 3839
- On debugging, it was discovered that graph has more than one components, Mostly due to restaurant or customer from out of Koramangala zone (or from the border).

- Also, to improve compliance, check the result by using time = 10 for pairs with time < 10.

Following are the results by applying above observations:

Total number of orders: 4306

Method	Accurate	Compliant
Existing	<b>2652 (61.6%)</b>	<b>2638 (61.3%)</b>
New	2633 (61.1%)	1777 (41.3%)

### Neighbor level 5

Total number of orders: 4432

Method	Accurate	Compliant
Existing	2729 (61.6%)	<b>2718 (61.3%)</b>
New	<b>2753 (62.1%)</b>	1763 (39.8%)

### Iterations

1. Complete the graph by adding missing entries.
2. Repeat by increasing number of neighbors per Geohash.
3. Explore other options to get time from distance.

## L8 Geohash with pings data relayed over OSM data

### Steps

- Collect the pings data from the de\_locations table (For duration: 26/01 to 02/02) and parse the locations column to find out the time required to travel from one geohash to another.
- Divide the data into Breakfast, Lunch and Dinner.
- If multiple values exist for the same geohash pair and time of day, remove outliers and take the maximum one.
- Add this info on top of the OSM time data that is collected in the earlier experiment.

### Results

#### Neighbor level 3

Total number of orders: 4306

Method	Accurate	Compliant
Existing	<b>2652</b>	<b>2638</b>
New	1429	50

#### Observations

- Both accuracy and compliance is affected. As number of non-compliant orders are equal to number of orders, prediction is always less than actual time.
- Maybe we don't have enough data points. Let's retry the experiment by collecting ping logs for longer duration (1 month) and geohash neighbors till level 5.
- Also, experiment by using Level 7 geohash instead of Level 8.

#### Neighbor level 5

Total number of orders: 4432

Method	Accurate	Compliant
Existing	<b>2729</b>	<b>2718</b>
New	787	30

### Neighbor level 3 to 5

Skipping immediate neighbors till level 2

Total number of orders: 4432

Method	Accurate	Compliant
Existing	<b>2729 (61.57%)</b>	<b>2718 (61.33%)</b>
New	2541 (57.33%)	628 (14.16%)

## L8 Geohash with only pings data

### Results

#### Neighbor level 5

Total number of orders: 4385

Method	Accurate	Compliant
Existing	<b>2708 (61.76%)</b>	<b>2689 (61.32%)</b>
New	1344 (30.65%)	41 (1%)

### Neighbor level 3 to 5

Skipping immediate neighbors till level 2

Total number of orders: 4371

Method	Accurate	Compliant
Existing	2700 (61.77%)	<b>2680 (61.31%)</b>
New	<b>2772 (63.42%)</b>	1485 (34%)

## L7 Geohash with only OSM data

### Results

#### Neighbor level 5

Total number of orders: 6198

Method	Accurate	Compliant
Existing	<b>3642 (58.8%)</b>	<b>3847 (62.1%)</b>
New	3569 (57.5%)	2819 (45.5%)

## L7 Geohash with pings data relayed over OSM data

## Results

### Neighbor level 5

Total number of orders: 6198

Method	Accurate	Compliant
Existing	<b>3642 (58.8%)</b>	<b>3847 (62.1%)</b>
New	3412 (55.1%)	3553 (57.3%)

### L50 Geohash with OSM data

## Results

Total number of orders: 900

Method	Accurate	Compliant
Existing	36.66%	
New	50.0%	

## Action Items (Discussed in meeting with Product)

1. Use `reached_time` instead of `delivered_time` to calculate actual LM. So actual LM will become ***reached\_time - pickedup\_time***
2. Repeat experiment with L50 (50mX50m custom geohash). Having a bigger geohash will help in reducing pings inaccuracy.
3. Filter pings based on accuracy. Experiment with different threshold numbers.
4. Use GDMA instead of OSM for base values.
5. Divide the available pings data in different segments based on time.
  - Start with B, L and D
  - Use 24 segments based on each hour of the day.

## Use reached time instead of delivered time

### L8 Geohash with only OSM data

## Results

### Neighbor level 5

Total number of orders: 3272

Method	Accurate	Compliant
Existing	1564 (47.78%)	<b>2430 (74.27%)</b>
New	<b>2203 (67.32%)</b>	1686 (51.52%)

### Neighbor level 3 to 5

Total number of orders: 2896

Method	Accurate	Compliant
Existing	1375 (47.48%)	<b>2184 (75.41%)</b>
New	<b>1997 (68.95%)</b>	1472 (50.82%)

### ***Neighbor level 4 to 5***

Total number of orders: 2896

Method	Accurate	Compliant
Existing	1375 (47.47%)	<b>2184 (75.41%)</b>
New	<b>1994 (68.85%)</b>	1480 (51.10%)

### ***Neighbor level 2 to 5***

Total number of orders: 2896

Method	Accurate	Compliant
Existing	1375 (47.47%)	<b>2184 (75.41%)</b>
New	<b>1998 (68.99%)</b>	1469 (50.72%)

### ***Neighbor level 5 to 5***

Total number of orders: 2896

Method	Accurate	Compliant
Existing	1375 (47.47%)	<b>2184 (75.41%)</b>
New	<b>1972 (68.09%)</b>	1561 (53.90%)

### ***Neighbor level 4 to 7***

Total number of orders: 2897

Method	Accurate	Compliant
Existing	1376 (47.47%)	<b>2184 (75.41%)</b>
New	<b>2004 (69.17%)</b>	1471 (50.77%)

## **L8 Geohash with only pings data**

### **Results**

### ***Neighbor level 3 to 5***

Total number of orders: 3259

Method	Accurate	Compliant
Existing	1554 (47.68%)	<b>2421 (74.28%)</b>
New	<b>2251 (69.07%)</b>	1609 (49.37%)

### ***Neighbor level 4 to 7***

Total number of orders: 2891

Method	Accurate	Compliant
Existing	1372 (47.45%)	<b>2181 (75.44%)</b>
New	<b>2069 (71.56%)</b>	1218 (42.13%)

### ***Neighbor level 5 to 7***

Total number of orders: 2889

Method	Accurate	Compliant
Existing	1371 (47.45%)	<b>2180 (75.45%)</b>
New	<b>2035 (70.43%)</b>	1493 (51.67%)

## L8 Geohash with pings data relayed over OSM data

### Results

#### *Neighbor level 4 to 7*

Total number of orders: 2897

Method	Accurate	Compliant
Existing	1376 (47.47%)	<b>2184 (75.41%)</b>
New	<b>2013 (69.48%)</b>	740 (25.54%)

## Experiment with Custom Geohash (50mX50m)

## L50 Geohash with only OSM data

### Results

#### *Neighbor level 5*

Total number of orders: 3276

Method	Accurate	Compliant
Existing	1566 (47.80%)	<b>2431 (74.20%)</b>
New	<b>2225 (67.91%)</b>	1620 (49.45%)

#### *Neighbor level 3 to 5*

Total number of orders: 3276

Method	Accurate	Compliant
Existing	1566 (47.80%)	<b>2431 (74.20%)</b>
New	<b>2226 (67.94%)</b>	1637 (49.96%)

## L50 Geohash with only pings data

### Results

#### *Neighbor level 3 to 5*

Total number of orders: 3274

Filtering pairs on 50Secs

Method	Accurate	Compliant
--------	----------	-----------

Existing	1565 (47.80%)	<b>2431 (74.25%)</b>
New	<b>2247 (68.63%)</b>	886 (27.06%)

## L50 Geohash with pings data relayed over OSM data

### Results

#### *Neighbor level 3 to 5*

Total number of orders: 3276

Method	Accurate	Compliant
Existing	1566 (47.80%)	<b>2431 (74.20%)</b>
New	<b>2179 (66.51%)</b>	706 (21.55%)

## Filter PINGS data based on accuracy

## L8 Geohash with only pings data

#### *Neighbor level 4 to 7*

Total number of orders: 2859

Filtering PINGS with less that 25m accuracy

Method	Accurate	Compliant
Existing	1359 (47.53%)	<b>2162 (75.62%)</b>
New	<b>2047 (71.59%)</b>	1138 (39.80%)

## PINGS data based on time of day

## Relevant Logics in code

### Last mile time calculation from distance

- A cron runs Every 6 hours
- Gets last 7 days trips for the area id
- Calculates intercept and slope by plotting distance and time
- Intercept and slope is stored in sla\_area\_data\_v3
- Intercept and slope are used to calculate last mile time from distance as follow
  - $\text{time} = \text{intercept} + \text{slope} * \text{last\_mile\_distance}$