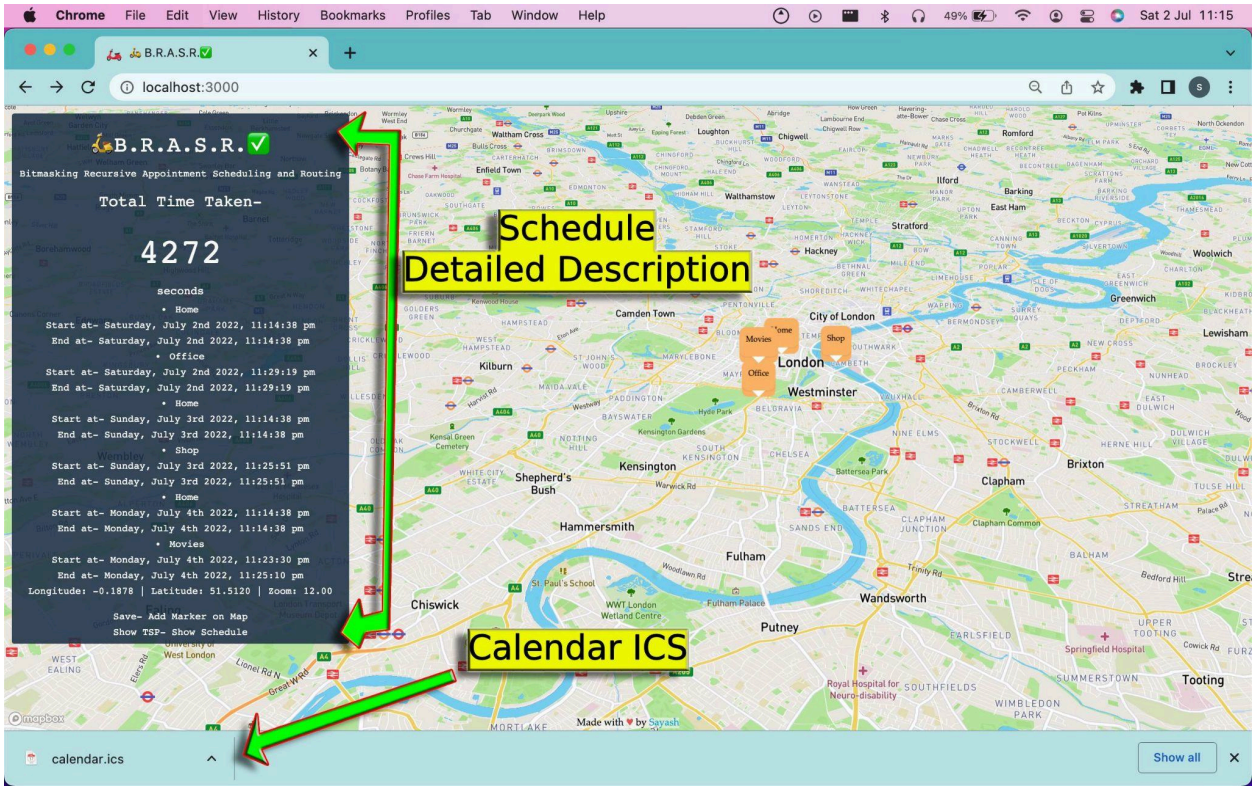
Sayash Raaj, **IIT Madras**, https://www.linkedin.com/in/sayashraaj/

Linear Programming Demand Supply Optimiser with EV Tour Optimisation

Fast, Accurate. Winner. LOG PIONEERING RESPONSIBLE ENERGY



Idea Validated, Credentials- Shell AI, EXL

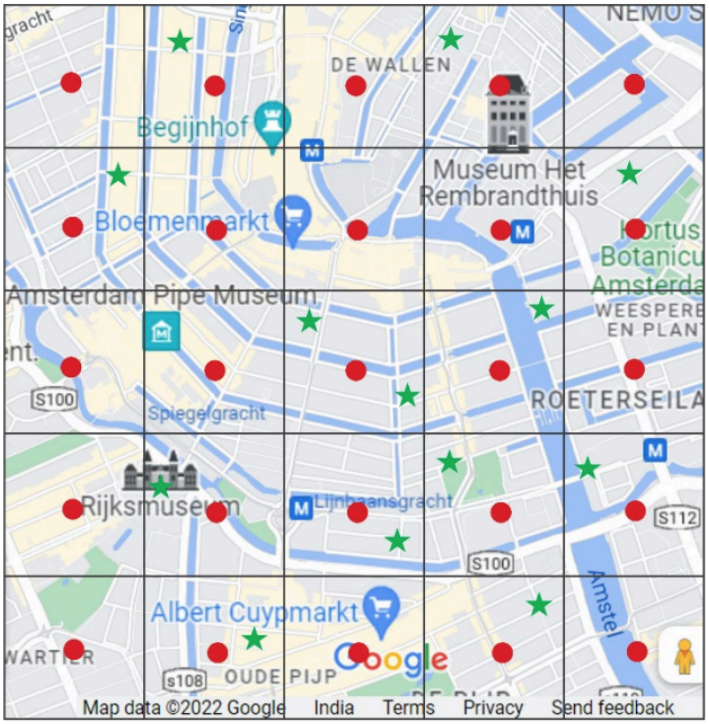
Two **Award-winning projects** modified and combined to create a winning project for this challenge, the **Linear Programming Demand Supply Optimiser with EV Tour Optimisation**

1) **Shell**- **Winner** of the University track of the Shell AI global hackathon, the first part focuses on creating a matrix with **demand supply forecasting for fast charging and slow charging stations (FCS and SCS)**. The forecasts then calculate the optimal fast and slow chargers required.
2) **EXL**- **Winner** of the global challenge winning the **$2000** first prize, the route optimisation algorithm for solving **operators' operational dynamics and charging**.

Objective with ALL Prerequisites covered, promises more ✅

A. Provide logic for a heat map of Public Fast charging for robust demography ✅

The solution uses **Demand-Supply matrix** to calculate the number of both fast charging and slow charging stations required including existing ones with **optimal objective function** using **constraints**



B. Create a matrix to deploy 1000 electric cargo 3W vehicles (GB/T connector) to estimate how many new fast chargers are required (incl the existing chargers) to cover the entire city with radius of 50kms. ✅

1. Bangalore/Delhi- Works for **ANY** city ✅
2. Range of the vehicle: 60kms- Works for **ANY** range ✅
3. Average km run – 300kms- Works for **ANY** avg km run ✅
4. Maximum usable SOC: 90%- Works for **ANY** maximum constraint ✅

C. Create a heat map of public fast charging for Cargo 3W cargo based on fleet operators' operational dynamics and type of charging. ✅- Done using **custom made linear programming algorithm** providing solutions in **seconds**

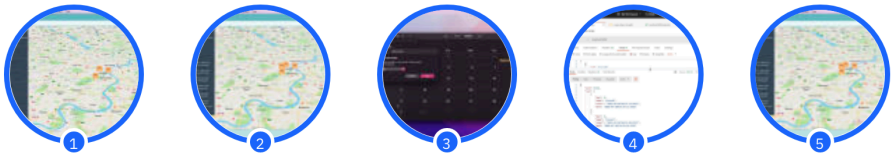
Prerequisite Study A (Forecasting techniques) - Understand forecasting of EV charging demand and EV infrastructure optimization for all EV segments ✅- Forecasting is done using **multiple polynomial regression techniques**

Prerequisite Study B (Heat mapping) - Derive at the demand & supply of vehicles & chargers, demand-supply constraints and objective ✅- included in the **Demand-Supply matrix**

Part A: Business EV Route Optimiser

Minimal travel time and waiting time for EV owners

HOW IT WORKS



1. Add Destinations with 1 click

   Vehicles add pre-defined destinations for optimisation

2. Automated Route Optimisation

   No input from the user, sit back and relax

3. Add and share to ANY calendar with 1 click

   🛵B.R.A.S.R.✅ Can be integrated with ANY calendar on Earth (Outlook, Teams, Google, Yahoo, Apple, etc)

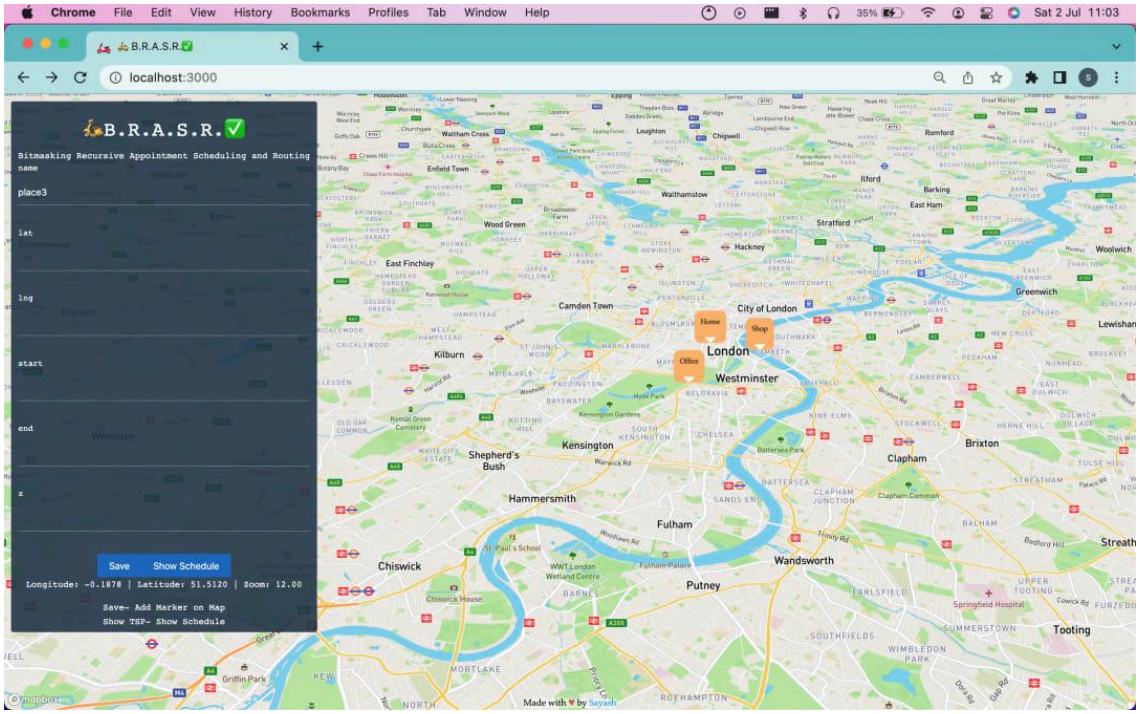
4. Open API available for businesses and developers

   (Optional) While a UI based approach is ideal for the majority of users, business integration and development can use an API wrapper for 🛵B.R.A.S.R.✅

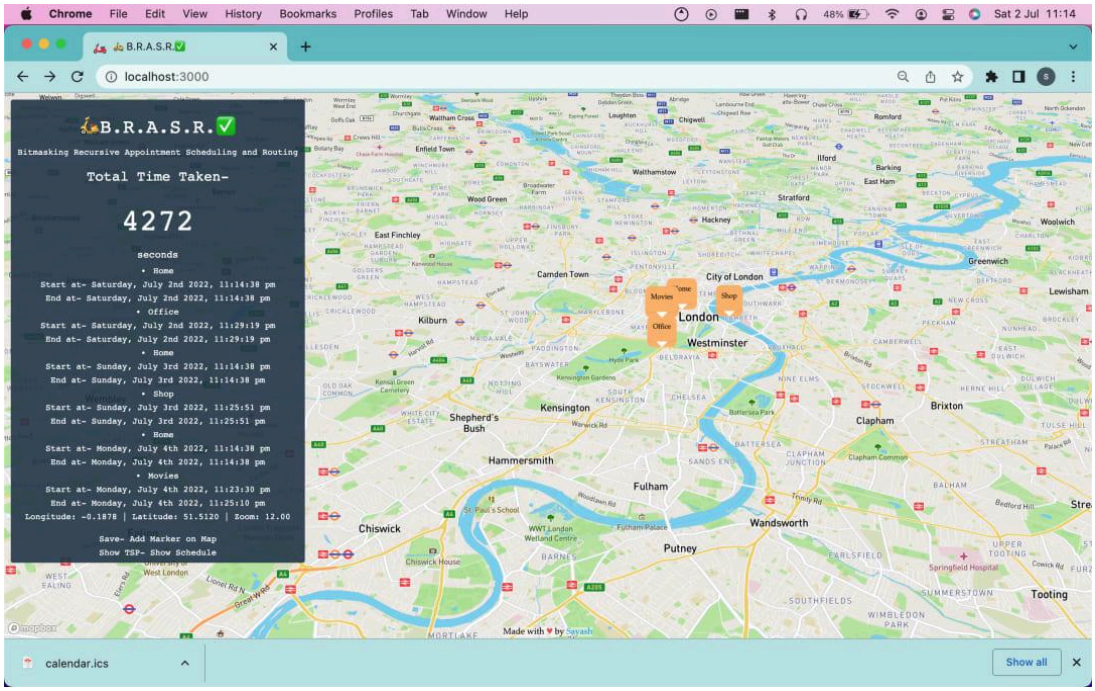
5. Rest and Watch The Sun Rise On a Grateful Universe

   With just 1 click, you have helped reduce Carbon emissions by 60%

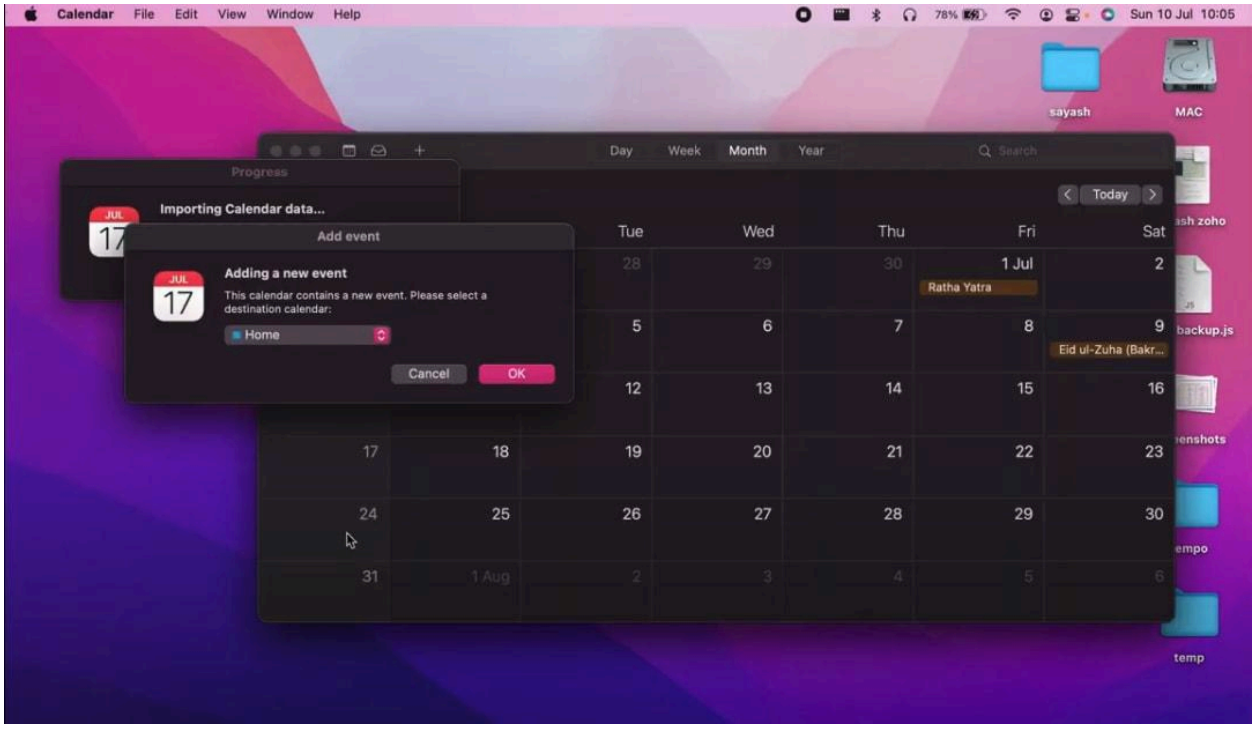
EV owner enters destinations including automated fetching of destinations



The minimal time required to route through all nodes is provided within seconds using the custom made algorithm



Can be integrated with any calendar using ICS files



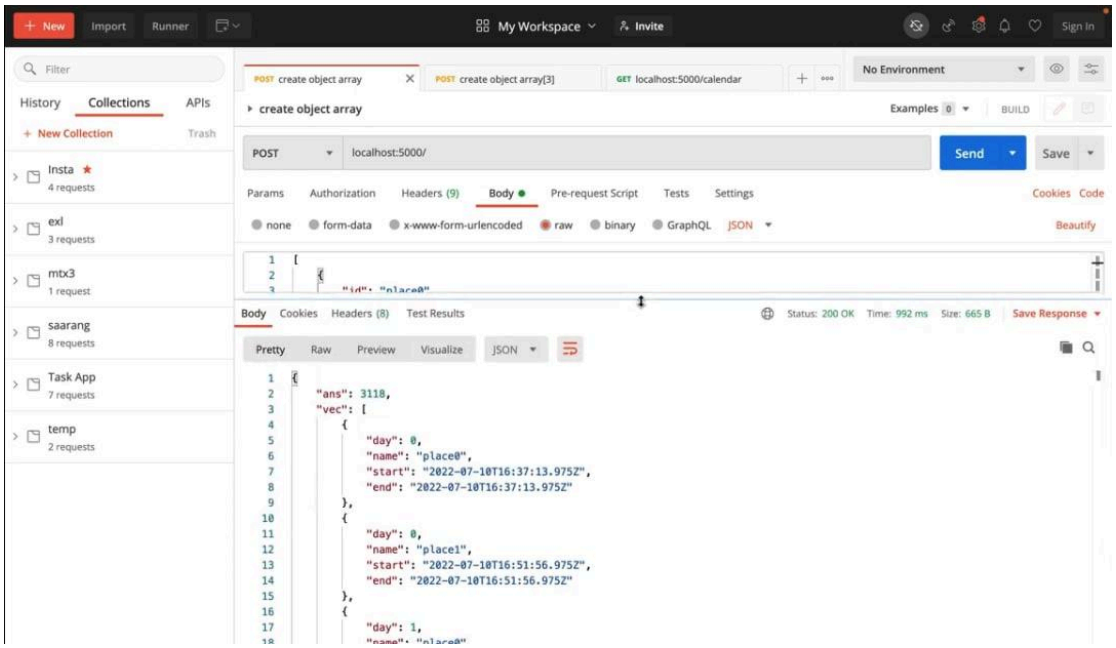
Calendar Integration Explained:

For **last mile** delivery operations, coordination is key, and is often a pain point for trip generation and mapping delivery destinations minimizing route cost.

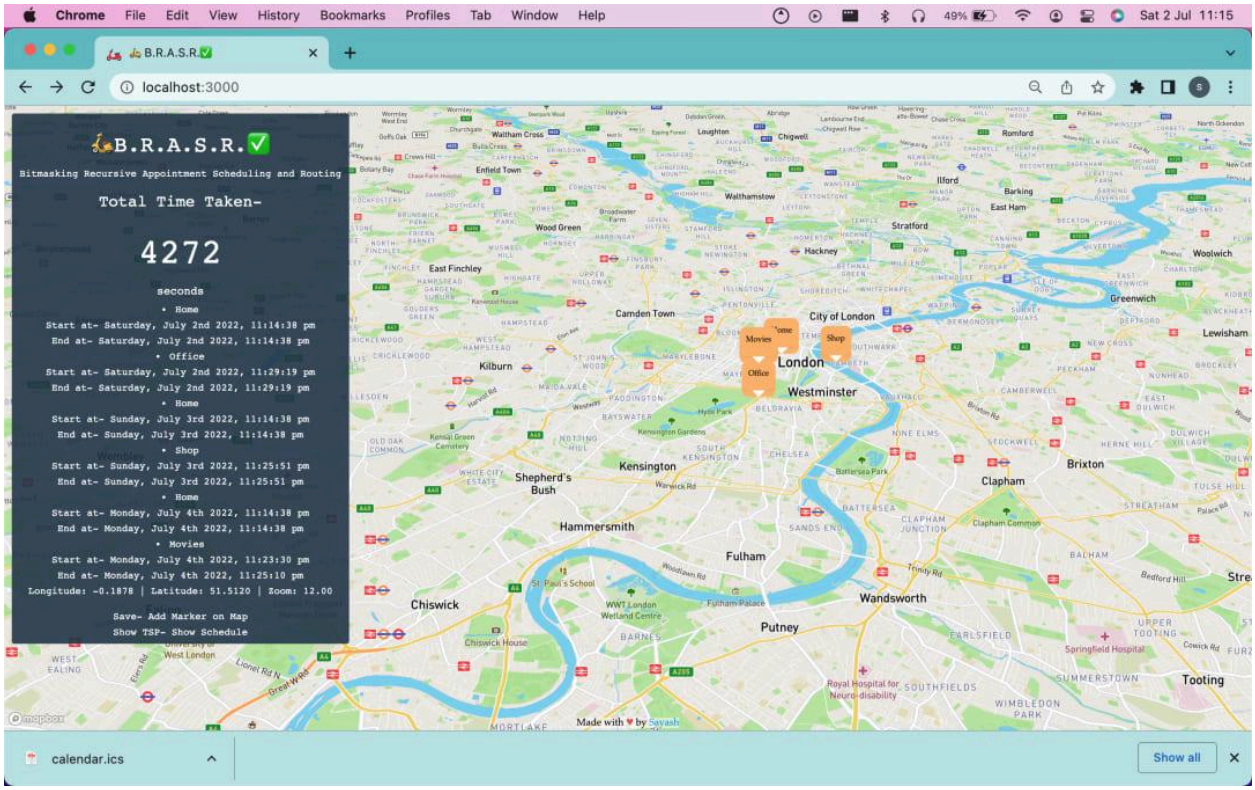
The Calendar Integration feature makes it possible for the fleet manager to coordinate with delivery executives in real-time and receive updates. The calendar events include details about each delivery executive's **forecasted** arrival time, waiting time, trip route and the order of deliveries for an optimal travel path minimizing route cost.

The calendar is an ICS file output, which makes it easy to integrate with ANY calendar application with just one click. The ICS file is **shareable**, which makes it easier to facilitate coordination amongst the fleet manager, last mile delivery executives, support team and agents in **real time**, eliminating information delays.

API based approach available for businesses



Maps based UI approach available for end user



Charging time can be varied, dynamic and the solution is robust to demographic changes. All data points are fetched in real time from **trusted** API sources including **Google Maps API** wrappers.

Part B: Linear Programming for Demand Supply

Acknowledging the **mentorship** and **guidance** of **Mr. Chinmay Shukla**, Log9, I have used **data** procured for the city of **Bangalore** in developing the solution and helping it align closely to a **real world solution**. Given below is a snippet of Bangalore EV data

Vehicle Number	Start Time	End Time	Start time	End time	Total Distance(Km)	Total Running Time	SOC Range	Energy Consumption	Total Halt Time
KA 51 AH 1277	4th Nov 2022, 12:54:35 AM	4th Nov 2022, 12:12:02 PM	12:54:35 AM	12:12:02 PM	23.97	3h 30m	100% - 52.9%	134.47	9h 32m 56s
KA 51 AH 1277	4th Nov 2022, 12:31:34 PM	4th Nov 2022, 5:25:42 PM	12:31:34 PM	5:25:42 PM	34.69	0s	100% - 30.6%	121.34	1h 51m 22s
KA 51 AH 1277	4th Nov 2022, 5:49:22 PM	4th Nov 2022, 7:58:11 PM	5:49:22 PM	7:58:11 PM	32.68	0s	99.1% - 23.7%	92.93	4m 40s
KA 51 AH 1277	4th Nov 2022, 11:28:11 PM	5th Nov 2022, 11:46:29 AM	11:28:11 PM	11:46:29 AM	15.68	4h 17m 1s	100% - 67.1%	104.46	10h 43m 46s
KA 51 AH 1277	5th Nov 2022, 12:01:09 PM	5th Nov 2022, 3:02:40 PM	12:01:09 PM	3:02:40 PM	20.9	0s	100% - 52.9%	86.95	42m 18s
KA 51 AH 1277	5th Nov 2022, 3:21:11 PM	5th Nov 2022, 8:39:50 PM	3:21:11 PM	8:39:50 PM	41.79	7m 38s	100% - 30.6%	90.51	1h 12m 35s
KA 51 AH 1277	6th Nov 2022, 12:56:51 AM	7th Nov 2022, 11:39:02 AM	12:56:51 AM	11:39:02 AM	18.94	3h 12m	99.1% - 23.7%	99.12	1d 8h 53m 23s
KA 51 AH 1277	7th Nov 2022, 11:53:32 AM	7th Nov 2022, 3:22:46 PM	11:53:32 AM	3:22:46 PM	37.12	0s	100% - 67.1%	78.57	50m 44s
KA 51 AH 1277	7th Nov 2022, 4:00:08 PM	7th Nov 2022, 7:15:54 PM	4:00:08 PM	7:15:54 PM	43.96	18m 54s	100% - 52.9%	116.49	53m 33s
KA 51 AH 1277	7th Nov 2022, 10:29:04 PM	8th Nov 2022, 12:45:33 PM	10:29:04 PM	12:45:33 PM	24.49	1h 20m	100% - 30.6%	115.23	12h 27m 27s
KA 51 AH 1277	8th Nov 2022, 1:00:03 PM	8th Nov 2022, 4:18:48 PM	1:00:03 PM	4:18:48 PM	33.68	0s	99.1% - 23.7%	132.64	1h 5m 4s
KA 51 AH 1277	8th Nov 2022, 5:00:52 PM	8th Nov 2022, 7:06:53 PM	5:00:52 PM	7:06:53 PM	17.71	0s	100% - 67.1%	78.57	29m 40s
KA 51 AH 1277	8th Nov 2022, 8:26:53 PM	9th Nov 2022, 1:53:18 PM	8:26:53 PM	1:53:18 PM	28.8	0s	100% - 52.9%	116.49	15h 21m 24s
KA 51 AH 1277	9th Nov 2022, 2:04:58 PM	9th Nov 2022, 4:28:55 PM	2:04:58 PM	4:28:55 PM	29.21	2h 47m 1s	100% - 30.6%	115.23	24m 24s
KA 51 AH 1277	9th Nov 2022, 4:51:35 PM	9th Nov 2022, 6:05:01 PM	4:51:35 PM	6:05:01 PM	9.62	1m 20s	57.7% - 34.5%	132.64	11m 16s
KA 51 AH 1277	9th Nov 2022, 6:31:01 PM	9th Nov 2022, 8:31:22 PM	6:31:01 PM	8:31:22 PM	24.48	52m	95.2% - 40.4%	123.12	3m 38s
KA 51 AH 1277	9th Nov 2022, 11:18:23 PM	10th Nov 2022, 12:29:58 PM	11:18:23 PM	12:29:58 PM	29.93	1h 25m 31s	57.7% - 34.5%	109.34	10h 44m 10s
KA 51 AH 9867	10th Nov 2022, 1:01:10 PM	10th Nov 2022, 1:02:50 PM	1:01:10 PM	1:02:50 PM	0.35	1m 10s	95.2% - 40.4%	94.29	0s
KA 51 AH 9867	4th Nov 2022, 12:53:01 AM	4th Nov 2022, 5:44:36 PM	12:53:01 AM	5:44:36 PM	33.83	10h 43m 46s	57.7% - 34.5%	75.92	13h 4m 22s
KA 51 AH 9867	4th Nov 2022, 6:10:21 PM	4th Nov 2022, 7:13:02 PM	6:10:21 PM	7:13:02 PM	6.66	42m 18s	95.2% - 40.4%	81.76	16m 21s
KA 51 AH 9867	4th Nov 2022, 7:49:02 PM	5th Nov 2022, 2:22:55 PM	7:49:02 PM	2:22:55 PM	41.15	1h 12m 35s	57.7% - 34.5%	70.84	14h 29m 22s
KA 51 AH 9867	5th Nov 2022, 2:40:04 PM	5th Nov 2022, 6:36:08 PM	2:40:04 PM	6:36:08 PM	49.26	1d 8h 53m 23s	95.2% - 40.4%	77.6	59m 33s
KA 51 AH 9867	5th Nov 2022, 6:57:58 PM	5th Nov 2022, 8:15:12 PM	6:57:58 PM	8:15:12 PM	6.53	50m 44s	57.7% - 34.5%	85.07	35m 54s
KA 51 AH 9867	5th Nov 2022, 9:16:13 PM	7th Nov 2022, 9:28:05 AM	9:16:13 PM	9:28:05 AM	0.02	53m 33s	95.2% - 40.4%	81.58	1d 12h 7m 2s
KA 51 AH 9867	7th Nov 2022, 9:28:15 AM	7th Nov 2022, 2:52:35 PM	9:28:15 AM	2:52:35 PM	42.85	12h 27m 27s	100% - 39.8%	72.56	1h 55m 27s

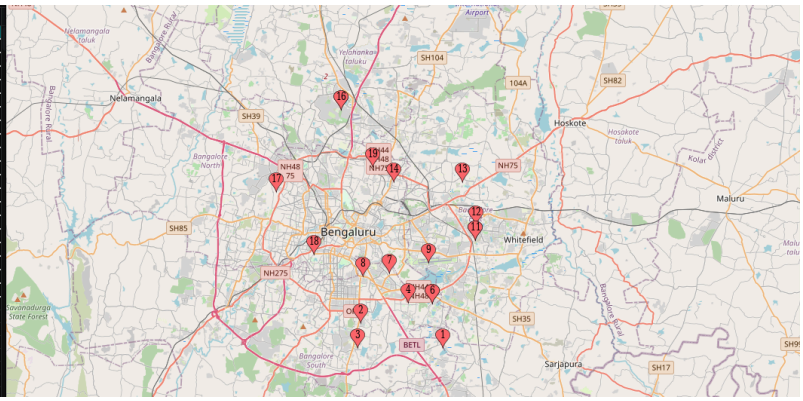
Crowdsourcing for Obtaining Existing EV Infrastructure Data

Following the success of the project where my college, The Indian Institute of Technology Madras developed a crowdsourcing initiative to collect real time information on areas affected by waterlogging or flooding during the 2021 northeast monsoon https://www.thehindu.com/news/cities/chennai/iit-m-revamps-website-to-crowdsource-waterlogging-information-in-city/article66100489.ece

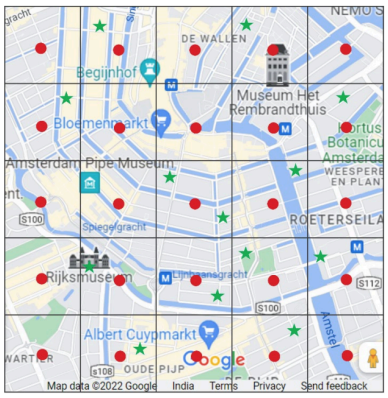
A crowdsourcing initiative to obtain details about the existing infrastructure pertaining to EVs including Fast and Slow chargers is a possible feasible solution with the IIT Madras project proving its feasibility and success.

A visual representation using a satellite imagery of Bangalore and the data procured, a possible grid division for further analysis is given:

Sl no	City	Location /. Area	Client	Lat Long
1	BLR	Meenakshi Layout	Flipkart	12.88168, 77.67627
2	BLR	Bannergatta Rd	Flipkart	12.9036752, 77.6015803
3	BLR	Arekere	Flipkart	12.8812125, 77.5986406
4	BLR	HSR	Amazon	12.9217447, 77.6445854
5	BLR	Bellandur	Amazon	12.9207375, 77.6666719
6	BLR	Bellandur	Amazon	12.9207375, 77.6666719
7	BLR	Neelasandra	Amazon	12.9475095, 77.6271889
8	BLR	Adugodi	Amazon	12.9452546, 77.6033552
9	BLR	Murgesh Palya	Amazon	12.957321, 77.663007
10	BLR	Dhoddenakundi	Amazon	12.9769194, 77.7057606
11	BLR	Dhoddenakundi	Dhelivery	12.9769194, 77.7057606
12	BLR	Hoodi KEB Colony	Dhelivery	12.990761, 77.70648
13	BLR	Nisarga Layout	Dhelivery	13.0286257, 77.6937871
14	BLR	HBR Layout	Dhelivery	13.02878, 77.6311604
15	BLR	Yelahanka	Dhelivery	13.0931291, 77.583606
16	BLR	Yelahanka	Dhelivery	13.0931291, 77.583606
17	BLR	Peenya	Dhelivery	13.0201083, 77.5242247
18	BLR	Deepanjali Nagar	Dhelivery	12.9641006, 77.5583794
19	BLR	Nagawara	Flipkart	13.042255, 77.6123005



Which is transformed into a grid network using the range of EV as a parameter, complete with the corresponding demand matrix from the data for each cell using hub and fleet operations and their delivery demands



The output includes the demand matrix and the existing FCS and SCS through crowdsourcing, forming the existing supply matrix.

Forecasting using polynomial regression models provides data for predicting growth and demand increase for each part of Bangalore according to grid divisions, given below is a brief on the methods used

PRE-INNOVATION

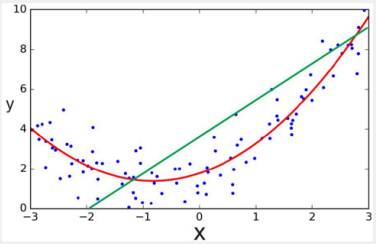
- Problem formulation
  - Customised personalised equation for fitting each trend line using multiple polynomial and exponential fitting equations
  - Yes, instead of heuristics and approximate solutions or imprecise algorithms, I used linear programming using the simplex method for solving an integer programming problem

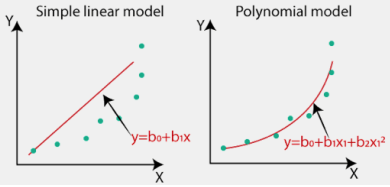
Regression Statistics	
Multiple R	0.924444
R Square	0.854596
Adjusted R Sq	0.847672
Standard Erro	8.717463
Observations	23

ANOVA

	df	*SS*	*MS*	*F*	*gnificance F*
Regression	1	9379.601	9379.601	123.4253	2.98E-10
Residual	21	1595.877	75.99416		
Total	22	10975.48			

	Coefficients	*tandard Err*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*
Intercept	10.3467	5.277558	1.960509	0.063337	-0.62858	21.32198	-0.62858	21.32198
0.8	96.27574	8.665922	11.10969	2.98E-10	78.25396	114.2975	78.25396	114.2975





Optimization using ILP

The objective of the EV infrastructure optimisation problem is to minimize the overall cost which comprises of total 3 components.

1. Cost of Customer Dissatisfaction ($Cost_{CD}$): This cost depends on how far customers have to travel to satisfy their EV charging demand.

$$Cost_{CD} = \sum_{i,j} Dist_{ij} \times DS_{ij}$$

2. Cost of Demand Mismatch ($Cost_{DM}$): This cost depends on the demand mismatch arising due to incorrect demand forecast as compared to the true values.

$$Cost_{DM} = \sum_{i} abs(D_{forecast,i} - D_{true,i})$$

3. Cost of infrastructure ($Cost_{IF}$): This cost depends on the operational, maintenance and amortised capital cost of owning EV infrastructure.

$$Cost_{IF} = \sum_{j} (SCS_j + r \times FCS_j)$$

Where, $r = 1.5$ is the ratio of the cost of fast charging station to that of slow charging station.

So, overall cost $(Cost) = a \times Cost_{CD} + b \times Cost_{DM} + c \times Cost_{IF}$. Here, $a = 1$, $b = 25$ and $c = 600$ are constants.

Aiming to lower range anxiety and minimizing travel time using $Cost_{CD}$, Forecasting error correction using $Cost_{DM}$, and optimal infrastructure cost using $Cost_{IF}$

We use a consolidated tri-objective minimization approach to find the global minima of the optimisation integer linear programming solution **within seconds**

The following constraints have been employed, a total of **1,000,000+** constraints:

All values of the number of slow (SCS_j) and fast charging stations (FCS_j) must be a non-negative integer.

Sum of slow (SCS_j) and fast charging stations (FCS_j) must be less than or equal to the total parking slots (PS_j) available at each j^{th} supply point.

You can only build incremental EV infrastructure on top of the 2018 infrastructure. That means, SCS_j and FCS_j must increase or stay constant year-on-year at each j^{th} supply point.

(Sum of fractional) Demand satisfied by each j^{th} supply point must be less than or equal to the maximum supply available.

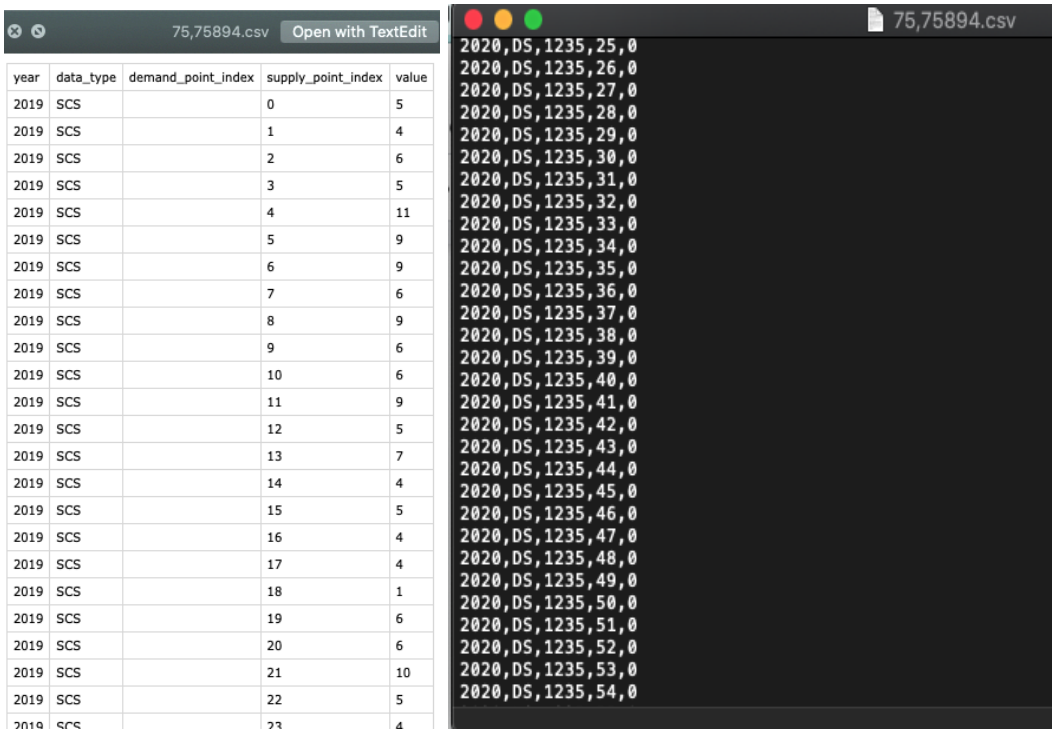
$$\sum_i DS_{ij} \leq Smax_j$$

(Sum of fractional) Forecasted demand at each i^{th} demand point must exactly be satisfied.

$$\sum_j DS_{ij} = D_{forecast,i}$$

Proof of Work: Coming to the Numbers

A snippet of the final output from the previous step of crowdsourcing: (Infra: R, Demand: L)

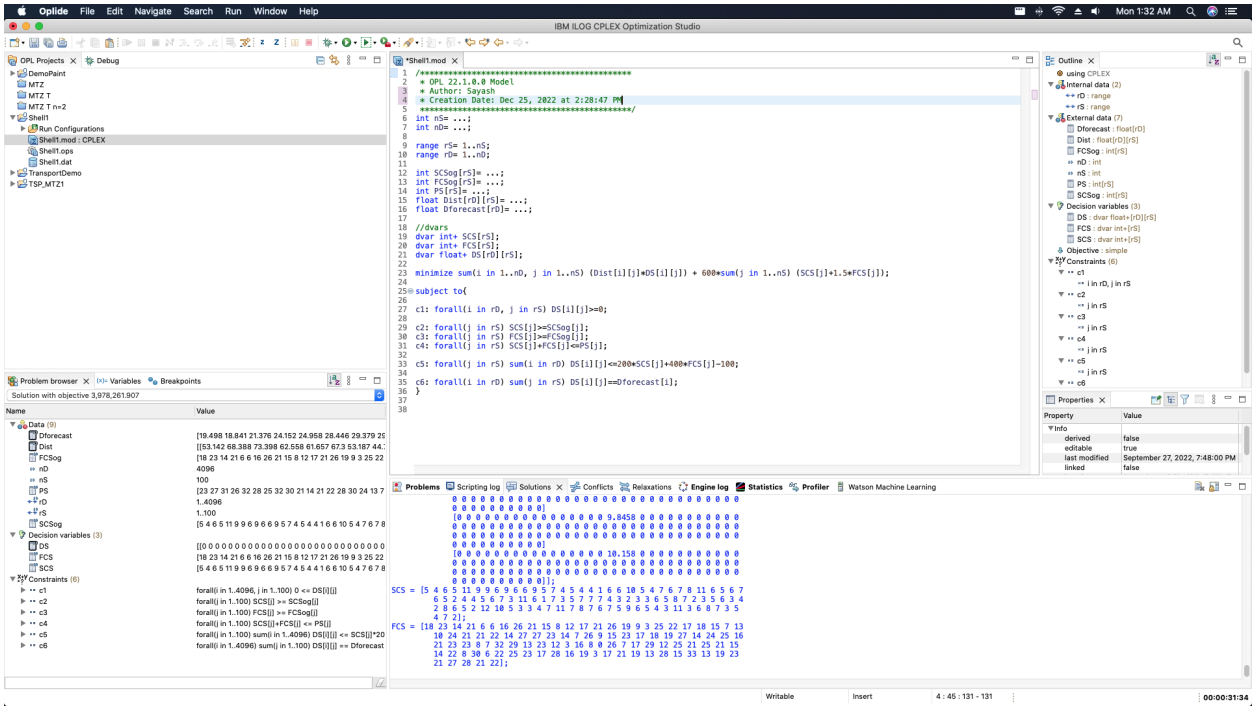


75,75894.csv Open with TextEdit

year	data_type	demand_point_index	supply_point_index	value
2019	SCS		0	5
2019	SCS		1	4
2019	SCS		2	6
2019	SCS		3	5
2019	SCS		4	11
2019	SCS		5	9
2019	SCS		6	9
2019	SCS		7	6
2019	SCS		8	9
2019	SCS		9	6
2019	SCS		10	6
2019	SCS		11	9
2019	SCS		12	5
2019	SCS		13	7
2019	SCS		14	4
2019	SCS		15	5
2019	SCS		16	4
2019	SCS		17	4
2019	SCS		18	1
2019	SCS		19	6
2019	SCS		20	6
2019	SCS		21	10
2019	SCS		22	5
2019	SCS		23	4

75,75894.csv

```
2020,DS,1235,25,0
2020,DS,1235,26,0
2020,DS,1235,27,0
2020,DS,1235,28,0
2020,DS,1235,29,0
2020,DS,1235,30,0
2020,DS,1235,31,0
2020,DS,1235,32,0
2020,DS,1235,33,0
2020,DS,1235,34,0
2020,DS,1235,35,0
2020,DS,1235,36,0
2020,DS,1235,37,0
2020,DS,1235,38,0
2020,DS,1235,39,0
2020,DS,1235,40,0
2020,DS,1235,41,0
2020,DS,1235,42,0
2020,DS,1235,43,0
2020,DS,1235,44,0
2020,DS,1235,45,0
2020,DS,1235,46,0
2020,DS,1235,47,0
2020,DS,1235,48,0
2020,DS,1235,49,0
2020,DS,1235,50,0
2020,DS,1235,51,0
2020,DS,1235,52,0
2020,DS,1235,53,0
2020,DS,1235,54,0
```



Code snippet for Integer Linear Programming ILP

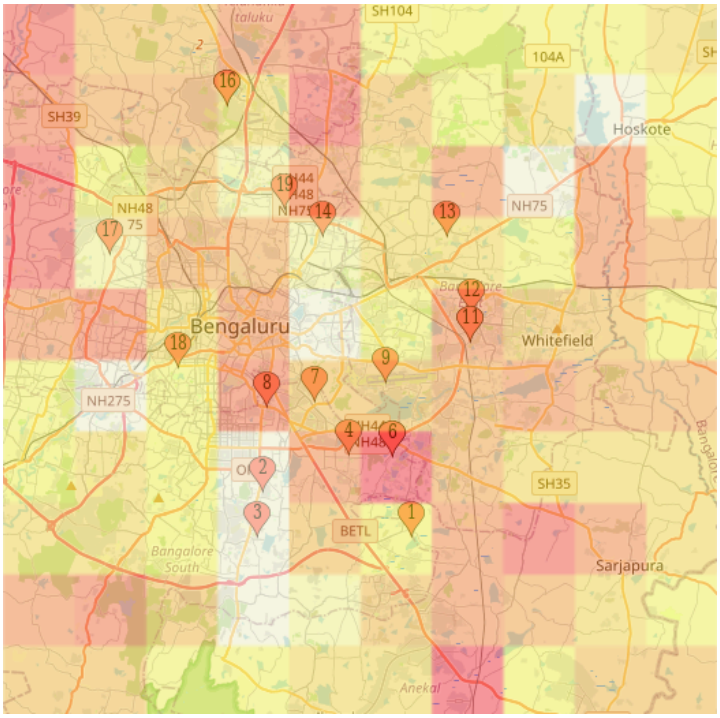
*Shell1.mod | Value for DS | Value for FCS | Value for SCS

rD (size 4096)	1	2	3	4	5	6
108	0	0	0	0	0	0
109	0	0	0	0	0	0
110	0	0	0	0	0	0
111	0	0	0	0	0	0
112	67.04...77308	0	0	0	0	0
113	0	0	0	0	0	0
114	0	0	0	0	0	0
115	0	0	0	0	0	0
116	0	0	0	0	0	0
117	0	0	0	0	0	0
118	0	0	0	0	0	0
119	0	0	0	0	0	0
120	0	0	0	0	0	0
121	0	0	0	0	0	0
122	0	0	0	0	0	0
123	0	0	0	0	0	0
124	0	0	0	0	0	0
125	0	0	0	0	0	0
126	0	0	0	0	0	0
127	0	0	0	0	0	0
128	0	0	0	0	0	0
129	0	0	0	0	0	0

*Shell1.mod | Value for DS | Val

rS (size 100)	Value
1	18
2	23
3	14
4	21
5	6
6	6
7	16
8	26
9	21
10	15
11	8
12	12
13	17
14	21
15	26
16	19
17	9
18	3
19	25
20	22
21	17
22	18
23	15
24	7
25	13
26	10
27	24

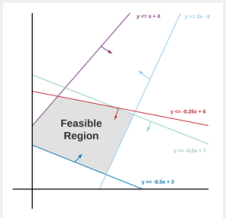
Actual Output after optimizing: Heatmap

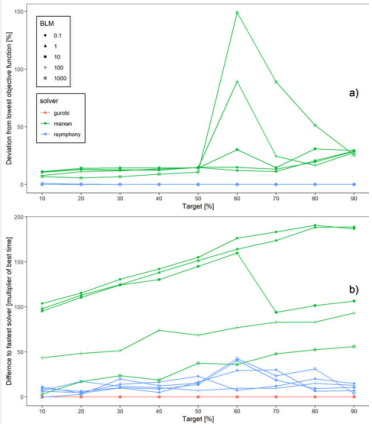


INNOVATION- THE UNIQUE SELLING POINT

- Converted floating point decision variables to 1e6 multiple integer decision variables for cent percent accuracy with complete precision, as opposed to the theoretical 1e7 precision for the classic floating point solution, complying with multiple imposed constraints on the decision variables

- Exponential performance improvement
  - As mentioned, I have converted the floating point decision variable solution to one with integer decision variables for ensuring precision, accuracy and an exponential increase in speed. Integer solutions are definite, 100% accurate, constraint verification
  - The solution submitted took less than **5 minutes** end-to-end, thus proving the efficiency and power of the consolidated pipeline I have created

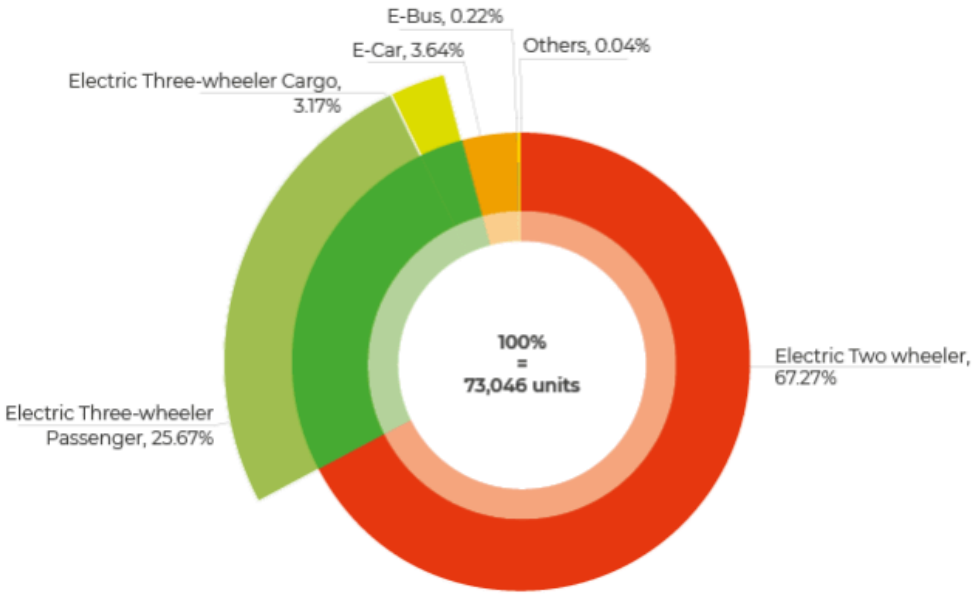




Conclusion

Solution Reliability

EV market division and solution benefit



3 Wheelers make a significant part of the EV market in India. Capturing, building and optimizing focusing on this segment of the EV market is bound to profit and increase revenue exponentially following a similar growth in sales of 3 Wheeler EVs.

My solution works on any and all commercial EV, including 3W EVs. The potential is infinite, and the solution acknowledges the uncertainty in this unexplored and nascent niche technology field, but believes **uncertainty brings promise**.

Data Generation and Procurement

Fetching data for existing EV charging networks including FCS and SCS. **Crowdsourcing** of data costs are minimal with a centralized basic CRUD enabled website. The main costs go towards publicizing the system.

IIT Madras, my college, has **successfully** used the project with minimal investment tying up with the **Government**

https://www.thehindu.com/news/cities/chennai/iit-m-revamps-website-to-crowdsource-waterlogging-information-in-city/article66100489.ece

Solution Part A: Business EV Route Optimiser
Uses a custom built highly optimized algorithm which has virtually light system requirements and involves hosting the site on a centralized server, costs are minimal to the order of INR 1000/month and can be integrated with the main website to further save server costs.
The costs can be recouped by launching the website as a freemium subscription model based semi business. The API method has already been included in the solution details and enables B2B integration on any platform, ideal to generate revenue through subscriptions.

Solution B: Linear Programming for Demand Supply
Uses Integer Linear Programming and is an extension of centralized server hosting and integration into the existing company platform. Costs are minimal to the order of INR 1000/month

The solution uses Linear programming with more than 1,000,000 constraints to converge to the optimal solution for constrained slow and fast charging infrastructure requirements.

The Demand Supply matrix is generated using the constrained ILP on forecasts made using multiple polynomial regression techniques.

The Demand Supply matrix is used to generate a heat map using float transformations, giving complete control to the business, freedom to do with the data as they please.

The heatmap generated by superimposing the 2D matrix output on the visual satellite imagery of the location is dynamic and robust to demographic changes, with calculations based on multiple polynomial regression models.

The TSP based EV tour optimisation algorithm, custom made using recursive programming techniques provides demographically robust solutions to optimal EV tour and charging requirements

The solution does everything, more than required and goes beyond the hackathon's requirements. Being the sole member of the team, working is something I will enjoy for this hackathon, changing the world into a better place one step at a time

Spending sleepless nights while balancing my academics at IIT Madras, I have worked on this project with the motivation for solving an important real world problem which will help everyone directly or indirectly. I thank the Jury for providing me with an opportunity to help make a significant positive impact on the world.

Sayash Raaj

IIT Madras

Harvard HPAIR Core Tech Associate

Amazon USA Intl' App Contest Winner

SMEC Indo-Australian Scholarship'22

Kalidas Madhavpeddi IITM Scholarship'22

https://www.linkedin.com/in/sayashraaj/