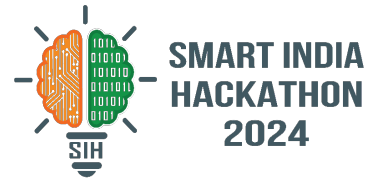


# SMART INDIA HACKATHON 2024



## HARIHACKERS

**Team Name :** HariHackers

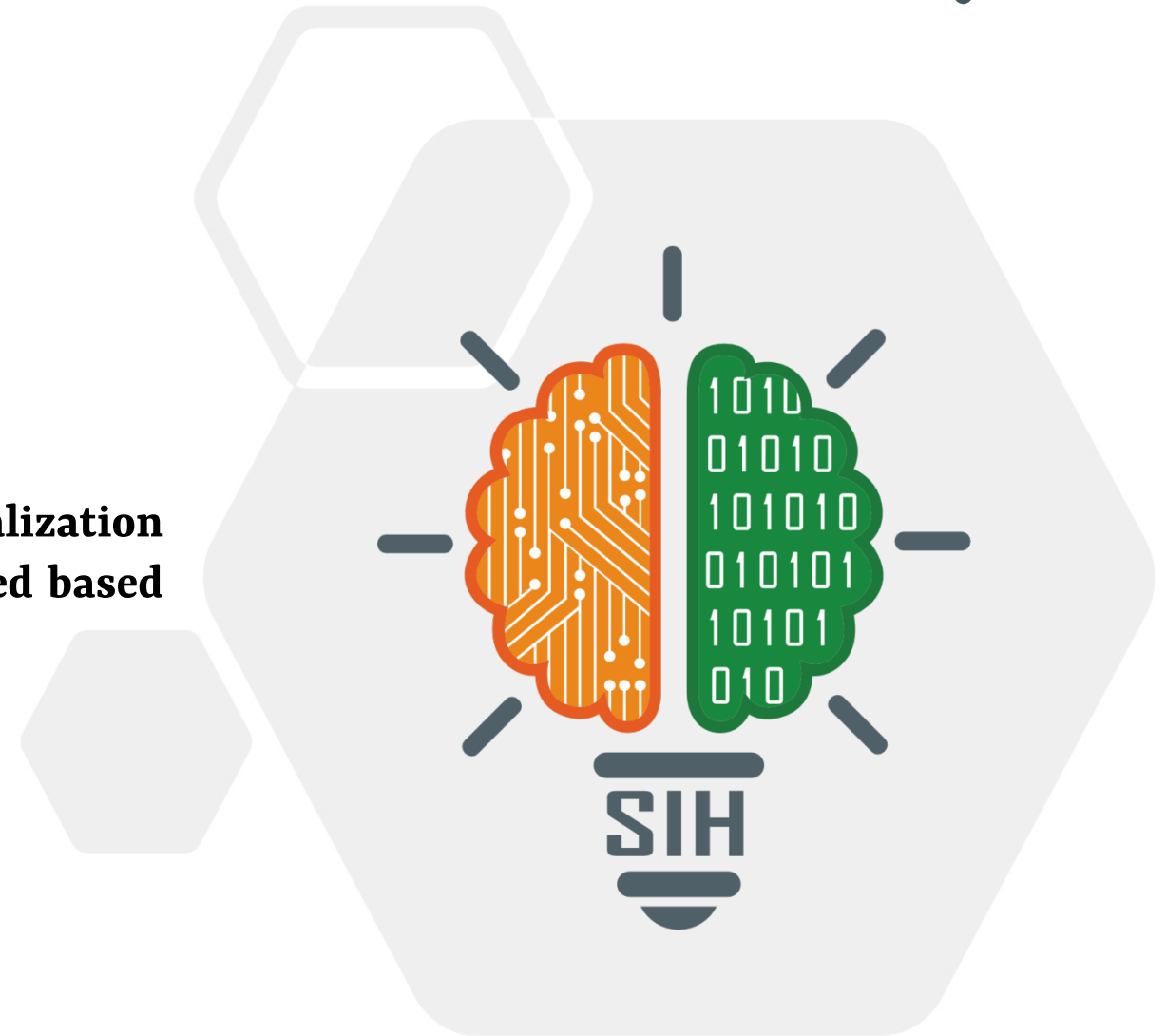
**Team ID :** 10

**Problem Statement Title :** Dynamic Route Rationalization Model based on Machine Learning/AI would be required based on Real-Time traffic and Road parameters.

**Problem Statement ID :** 1617

**Theme :** Smart Automation

**PS Category :** Software



# DYNAMIC ROUTE RATIONALIZATION MODEL

## The Problem Itself :

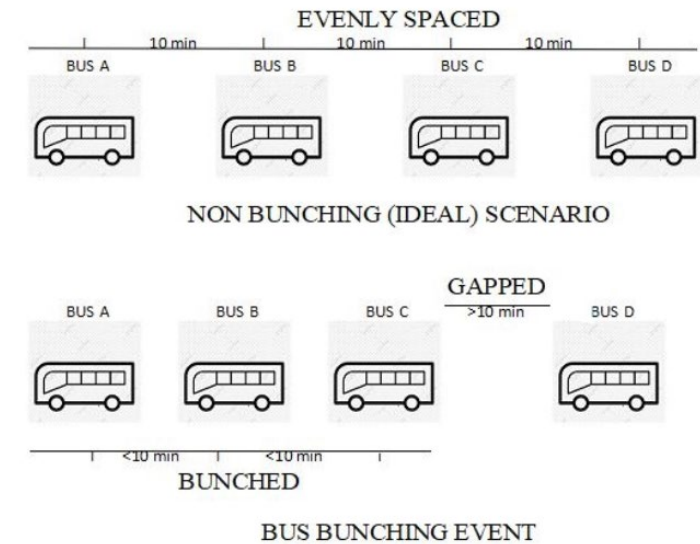
- Due to **Bus Bunching** the waiting time for the buses increases it leads to Decrease in **Passenger Satisfaction**.
- Also causes uneven number of passengers which leads to **higher operational cost**.

## Solution :

- For Bus bunching we will **train a Model** and will use real-time Data to predict bus bunching.
- If bus bunching is predicted, we will **optimize route**, and the schedule based on the Situations.
- We have two Strategies to tackle the Problem, first is **bus holding** and other is **stop skipping**.

## Innovation and uniqueness of the solution :

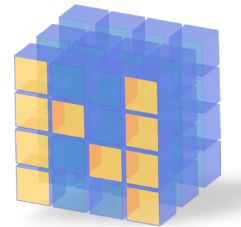
- The solution uses **machine learning algorithms** which in turn further uses **Real-time Data** and bus bunching prevention strategies to optimize and dynamically update the route and the schedule of the buses.
- At present very few technologies are used in **public transportation** in India and fixed schedules are used which makes it harder to locate and prevent bus bunching.



# TECHNICAL APPROACH

## Tools List :

- **Python** : Provides various **AI/ML** Libraries.
- **Pandas** : For working with CSV & Excel files.
- **NumPy** : For Mathematical & Scientific Calculations.
- **Matplotlib** : For Visualization & exploring the Data.
- **Scikit-learn** : Provides varies **ML Algorithms**.
- **Tensorflow** : Provides varies **DeepLearning/NLM** Algorithms.
- **Google Maps API** : For calculating the arrival time for each stops.
- **HTML/CSS/JavaScript** : For providing User Interface on the Web.



**HTML**



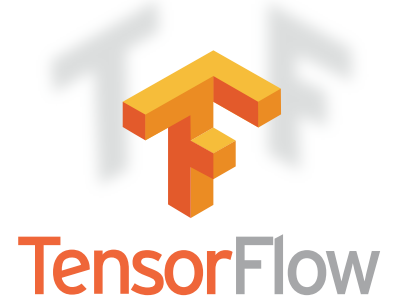
**CSS**



**JS**

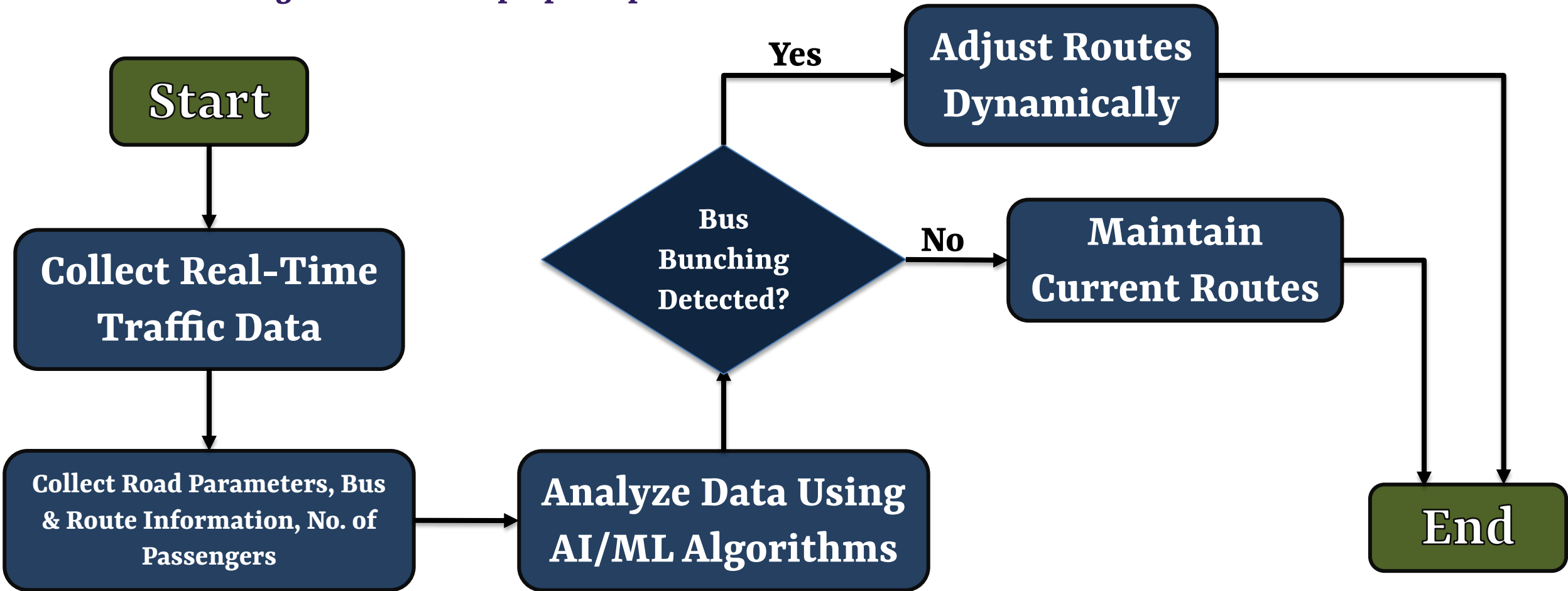


Google  
Maps



# TECHNICAL APPROACH

Flowchart describing the flow of the proposed problem :



# FEASIBILITY AND VIABILITY

- We will use Various predictive algo which uses real time data and patterns which leads to optimization of bus routes, thus making system feasible as the technology resources required is available and reliable.
- Getting accurate Real-time Prediction will depend on the quality of the data and the accuracy of prediction of the model. We will use Google Maps API for calculating arrival time for each buses on every remaining stops.
- Getting Predictions in an instant may not be possible as Real-time Prediction require high computational power. This may cause slight Delay's in the Output but it will not matter in this situation.

# IMPACT AND BENEFITS

**Decrease in Waiting time :** By preventing bus bunching the schedule of the buses on the route is maintained thus, waiting time is decreased and satisfaction of the passengers is increased.

**Increased Efficiency :** Preventing bunching ensures that buses are evenly distributed along their routes, optimizing the use of available buses and reducing idle time.

**Reduced Overcrowding :** Consistent spacing helps avoid situations where buses arrive so closely together that they are overcrowded, while others are nearly empty.

**Smoother Operations :** Drivers and transit authorities can manage schedules and operations more effectively when buses arrive in a predictable manner.

# RESEARCH AND REFERENCES

**Following is a credible reference which explains the bus bunching problem and a possible way to mitigate it :**

- <https://link.springer.com/article/10.1007/s42524-024-0297-1>
- Following is a visual representation of the problem :  
<https://setosa.io/bus/>