Group Project 2021

Intelligent Transportation System based on Connected Vehicles

Project Team

Pavan Idhole

BT17CME047

Shubham Chauhan

BT17ECE079

Pratik Patil

BT18CME041

Project Guide :

Dr. Neeraj Rao

Introduction

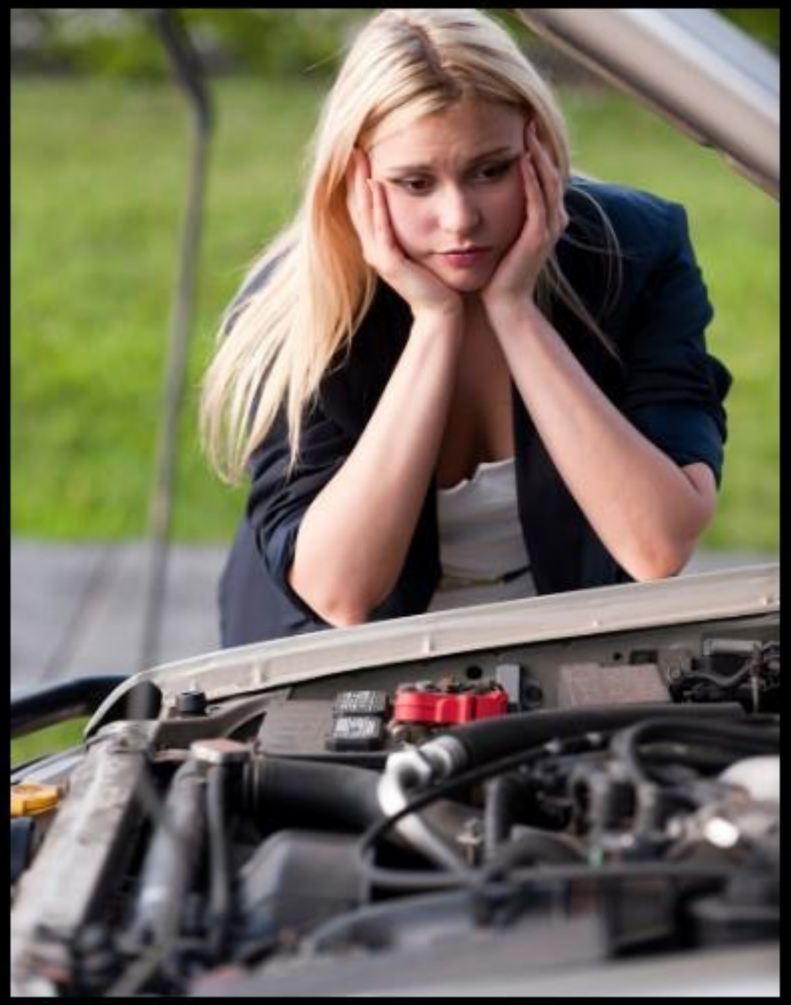
Problem Statement

To design an Intelligent Transportation System for a smart city using the IoT driven concept of connected vehicles to enhance the overall vehicular and mobility experience of people in city.

Connected Vehicles are smart vehicles that can interact with other vehicles as well as surrounding infrastructures to build better situational awareness and intelligent decision making.

Vehicular / Mobility related problems

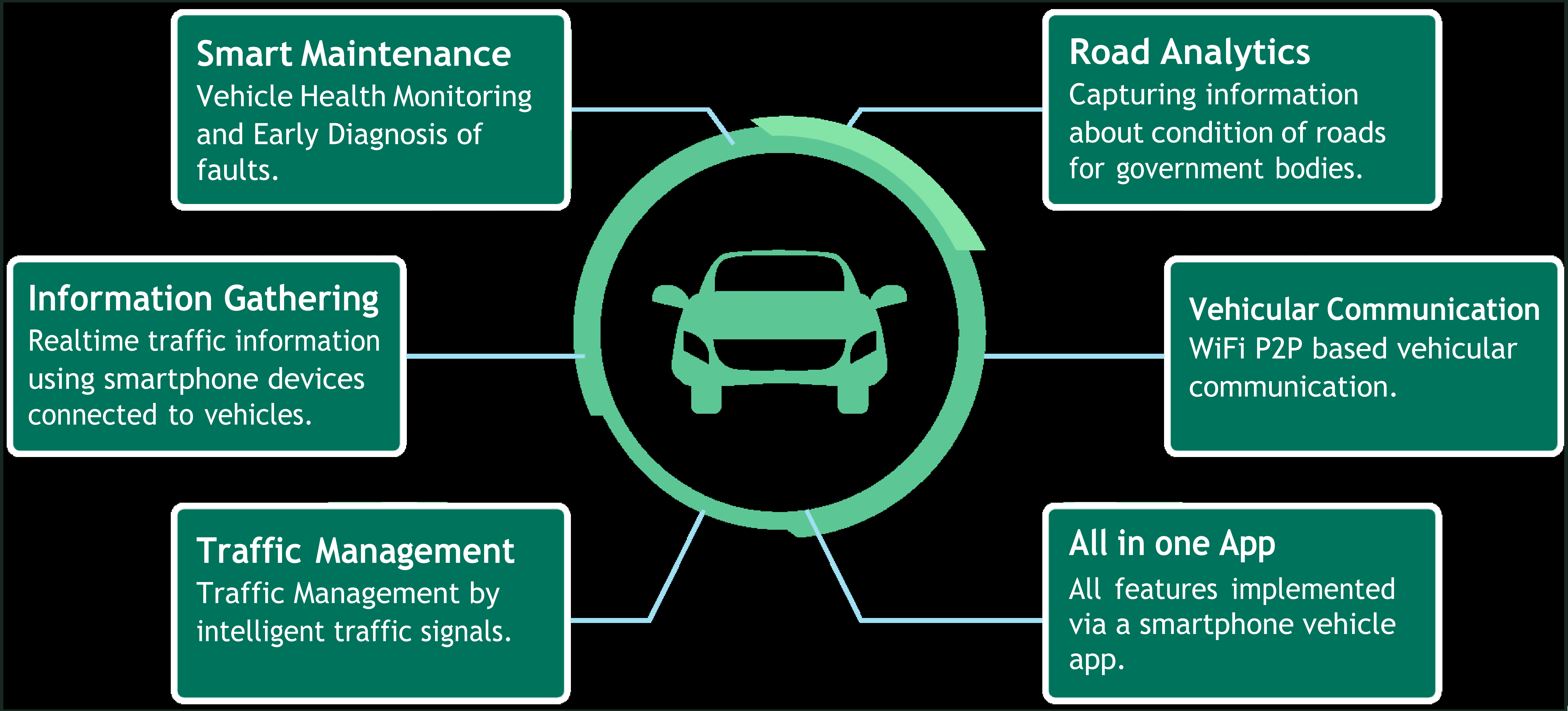
Poor Vehicle Maintenance



Inefficient Traffic Management

Poor Condition Damaged Roads

Proposed System



Smart Maintenance
Vehicle Health Monitoring and Early Diagnosis of faults.

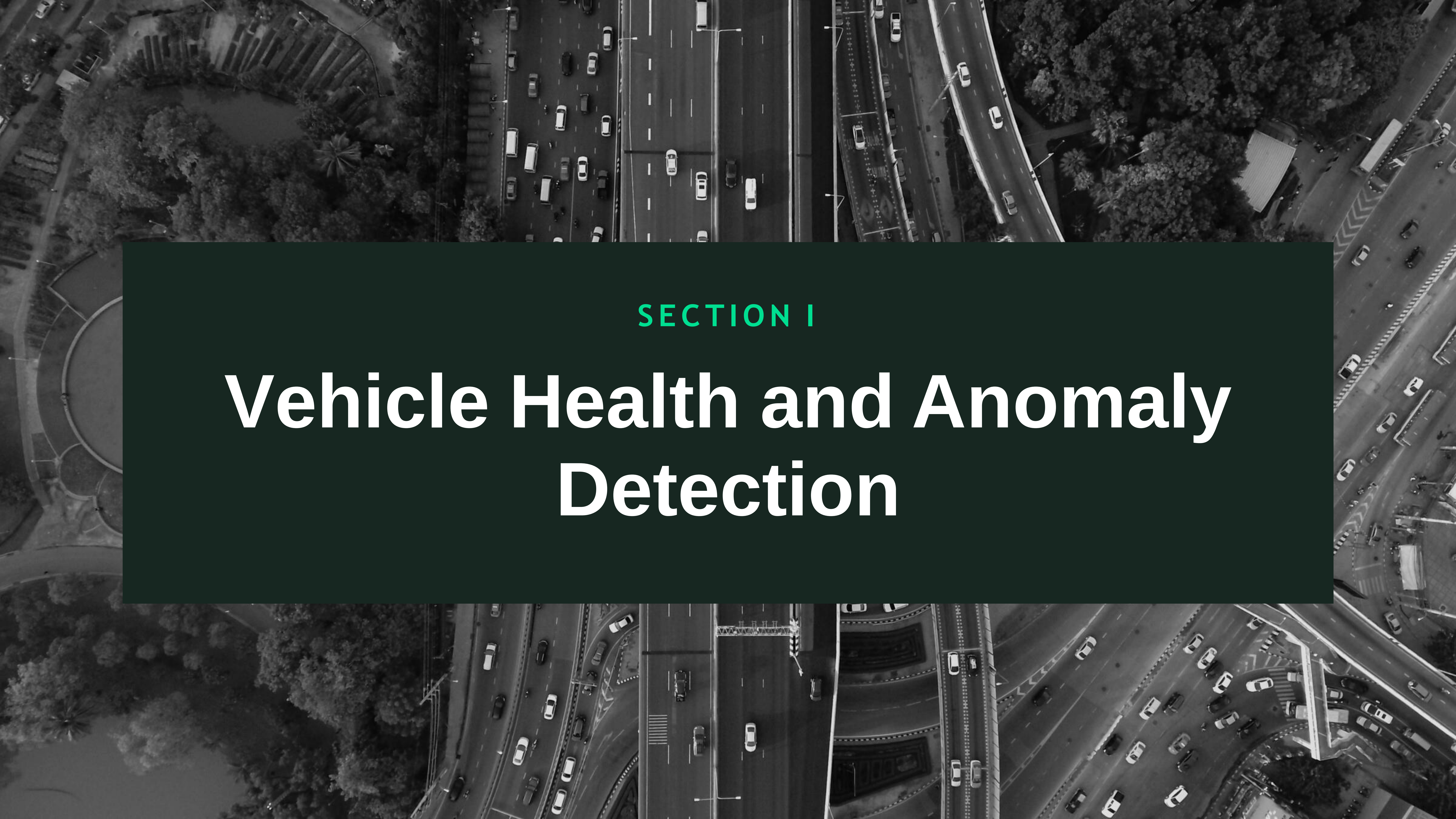
Road Analytics
Capturing information about condition of roads for government bodies.

Information Gathering
Realtime traffic information using smartphone devices connected to vehicles.

Vehicular Communication
WiFi P2P based vehicular communication.

Traffic Management
Traffic Management by intelligent traffic signals.

All in one App
All features implemented via a smartphone vehicle app.



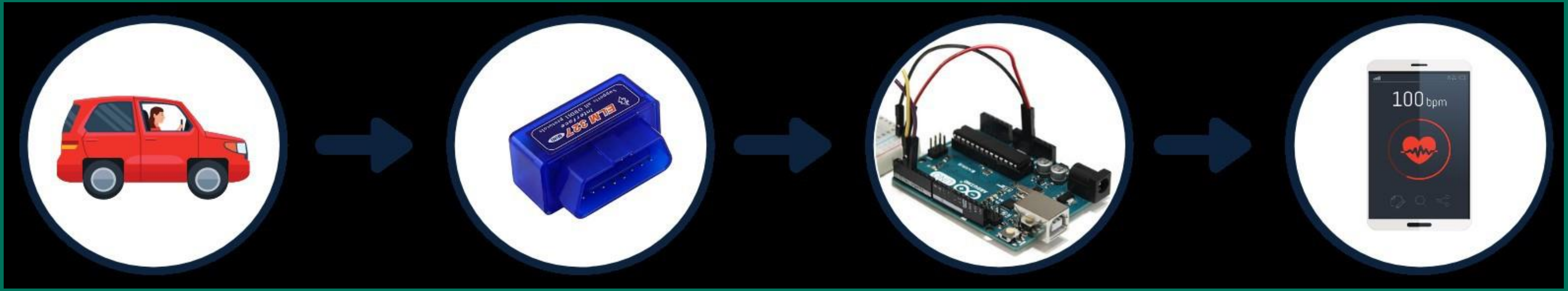
SECTION I

Vehicle Health and Anomaly Detection

Motivation

- Maintenance of vehicle is a tedious job for many vehicle owners.
- As per the motor laws, all vehicles are compliant to have On-Board Diagnostics (OBD) system installed in them.
- These OBD sensors monitor various health parameters of th vehicle. However this OBD data can be accessed only through technicians in the service centers.
- There is a need of simple solution for easy maintenance.

Approach



OBD compliant Vehicle

ELM 327 Interface

Arduino based circuit

Smartphone app for analysis

- OBD data from OBD compliant vehicle is collected from ELM 327 module connected to the OBD port.
- Arduino based circuit is used to collect data from ELM module and send it periodically to users smartphone via Bluetooth at regular time intervals.
- Anomaly detection algorithms deployed on users android device analyze the incoming data, and if anomalies are detected, notification is issued on both phone as well as the arduino circuit to notify the user.

Algorithms

Each data point contains 7 parameters and based on these paramters, we need to classify if the given data point has an anomaly in it or not.

Algorithms Used:

1. Local Outlier Factor Algorithm
2. Isolation Forest Algorithm
3. Detection using Neural Network

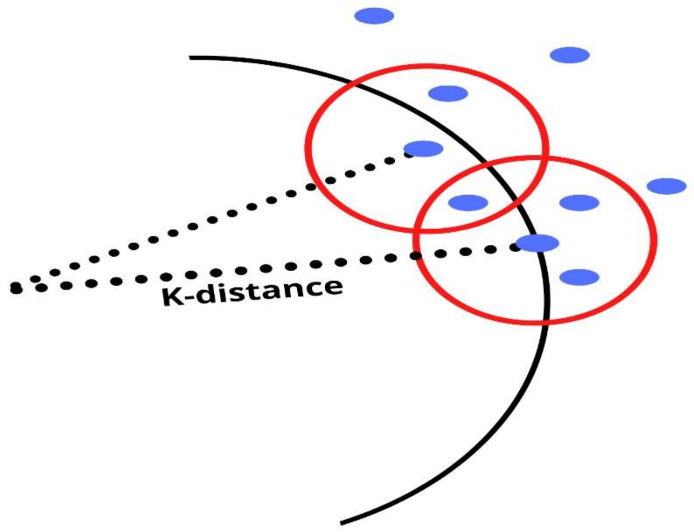
Parameters Monitored:

- BAROMETRIC PRESSURE
- ENGINE COOLANT TEMP
- ENGINE RPM
- INTAKE MANIFOLD PRESSURE
- MAF
- SPEED
- THROTTLE POSITION

Algorithms Description

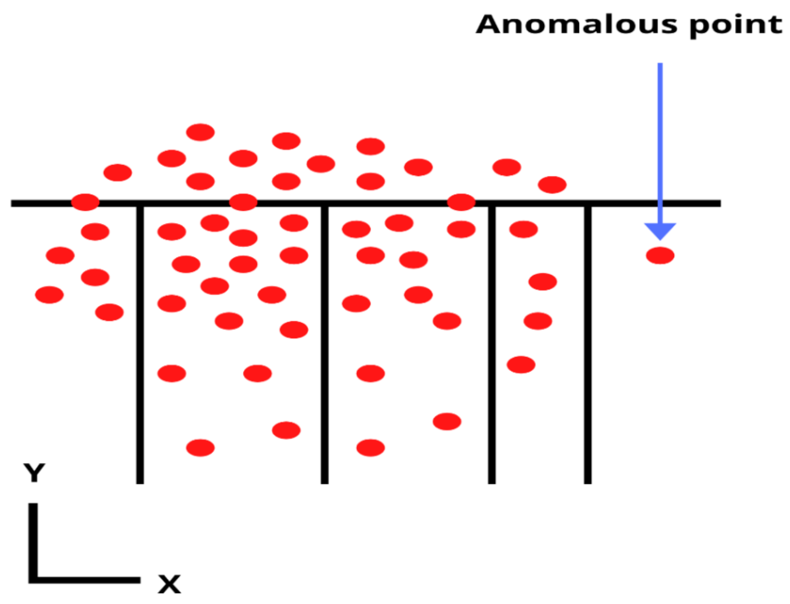
Local Outlier Factor (LOF)

- The local outlier factor is based on a concept of a local density.
- We can identify outlier points by comparing the local density of an object to the local densities of its neighbors.



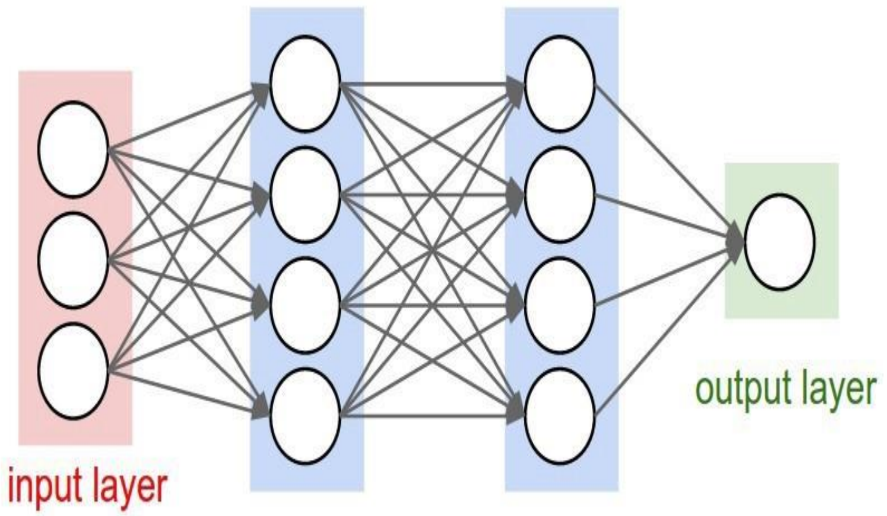
Isolation Forest Algorithm

- Isolation Forest recursively creates segments on dataset by randomly choosing an element and an arbitrarily split value for the feature.
- Abnormalities will be the points in the dataset that are closer to the root of the isolation tree.



Neural Network

- Neural Network model with an i/p, 4 hidden and o/p layer is built.
- Hyperparameters of the model are tuned to achieve high accuracy.
- Problem of overfitting is countered by setting dropout rate to 0.1.



Results Phase I

LOCAL OUTLIER FACTOR (LOF)

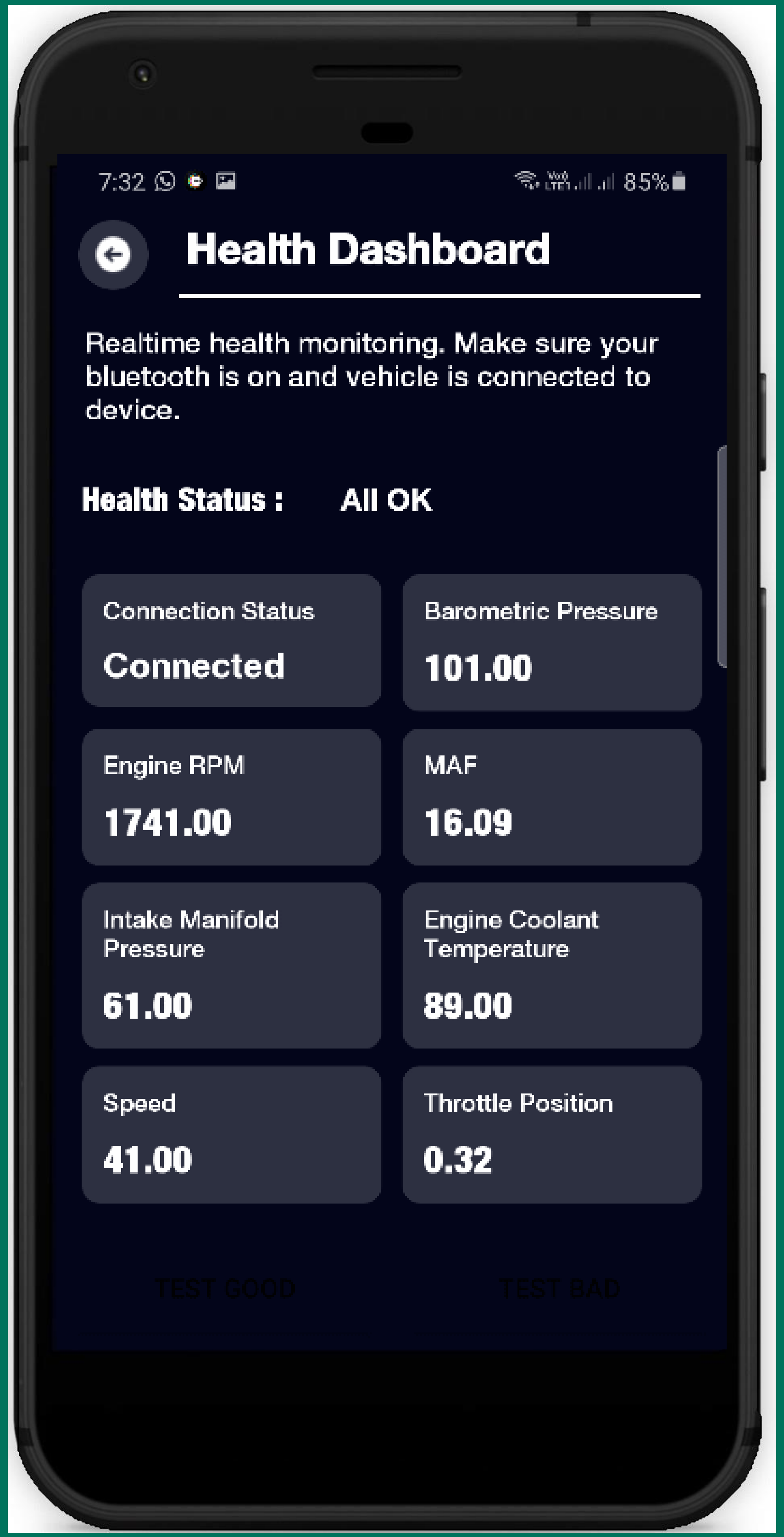
- Accuracy of Local Outlier Factor = **62.29%**
- Low Accuracy

ISOLATION FOREST

- Accuracy of Isolation Forest = **78.75%**
- Medium Accuracy

NEURAL NETWORK

- Accuracy of Neural Network = **99.79%**
- High Accuracy



Health Dashboard

Realtime health monitoring. Make sure your bluetooth is on and vehicle is connected to device.

Health Status : All OK

Connection Status
Connected

Barometric Pressure
101.00

Engine RPM
1741.00

MAF
16.09

Intake Manifold Pressure
61.00

Engine Coolant Temperature
89.00

Speed
41.00

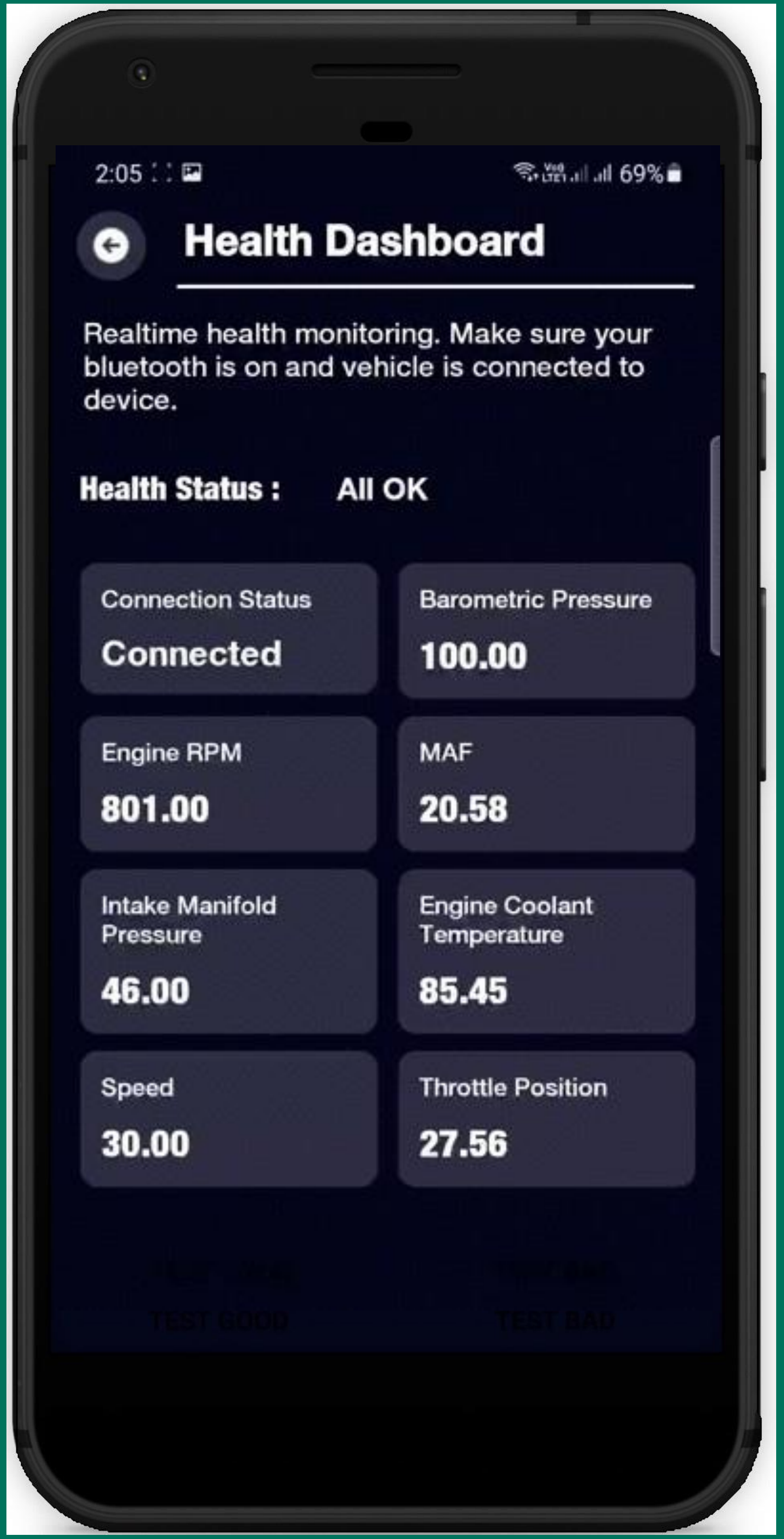
Throttle Position
0.32

Deployed in Android App

Results Phase II

NEURAL NETWORK

- Accuracy of Neural Network = **98.60%**
- High Accuracy
- Capable of identifying whether the anomaly is present or not and also detecting the senor with anomaly value.
- Slight decrease in accuracy as compared to previous one observed but it can be highly justified as prediciton classes have increased 16 times as compared to previous one.



2:05 69%

Health Dashboard

Realtime health monitoring. Make sure your bluetooth is on and vehicle is connected to device.

Health Status : All OK

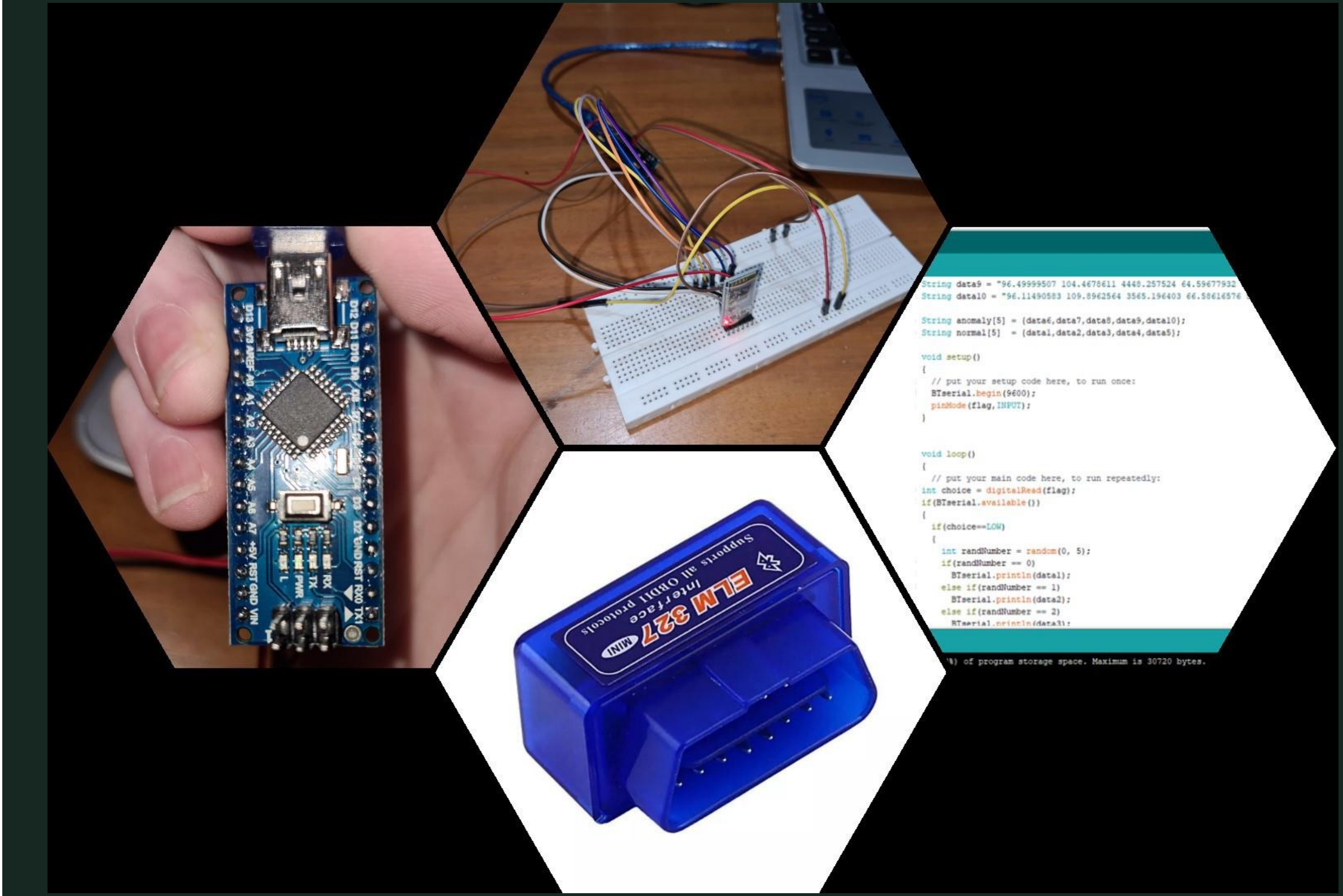
Connection Status Connected	Barometric Pressure 100.00
Engine RPM 801.00	MAF 20.58
Intake Manifold Pressure 46.00	Engine Coolant Temperature 85.45
Speed 30.00	Throttle Position 27.56

TEST GOOD

TEST BAD

Deployed in Android App

Testing Setup



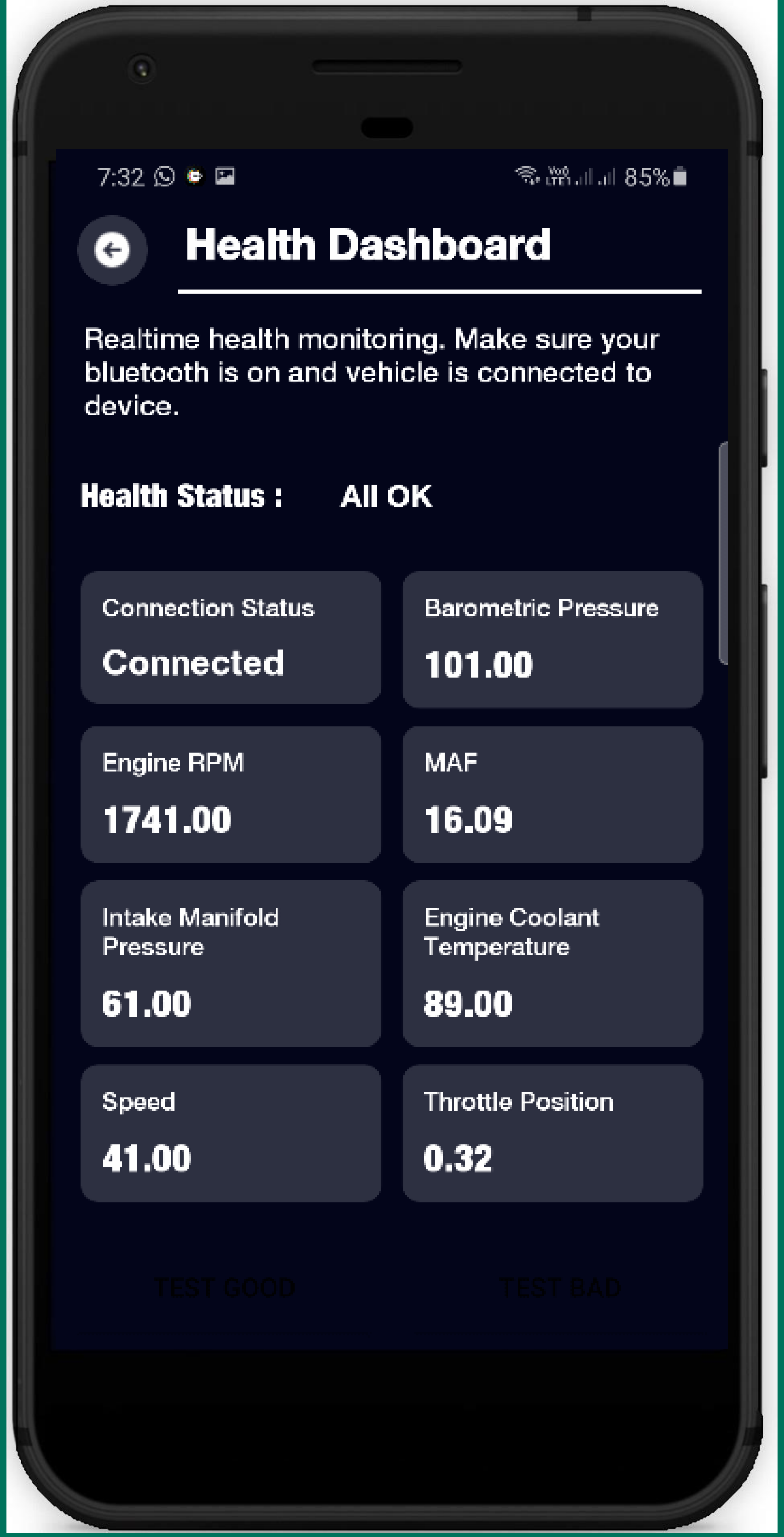
```
String data9 = "96.49999507 104.4678611 4448.257524 64.59677932
String data10 = "96.11490583 109.8962564 3565.196403 66.58616576

String anomaly[5] = {data6,data7,data8,data9,data10};
String normal[5]  = {data1,data2,data3,data4,data5};

void setup()
{
  // put your setup code here, to run once:
  BTserial.begin(9600);
  pinMode(flag,INPUT);
}


void loop()
{
  // put your main code here, to run repeatedly:
int choice = digitalRead(flag);
if(BTserial.available())
{
  if(choice==LOW)
  {
    int randNumber = random(0, 5);
    if(randNumber == 0)
      BTserial.println(data1);
    else if(randNumber == 1)
      BTserial.println(data2);
    else if(randNumber == 2)
      BTserial.println(data3);
```

```
%) of program storage space. Maximum is 30720 bytes.
```



Health Dashboard

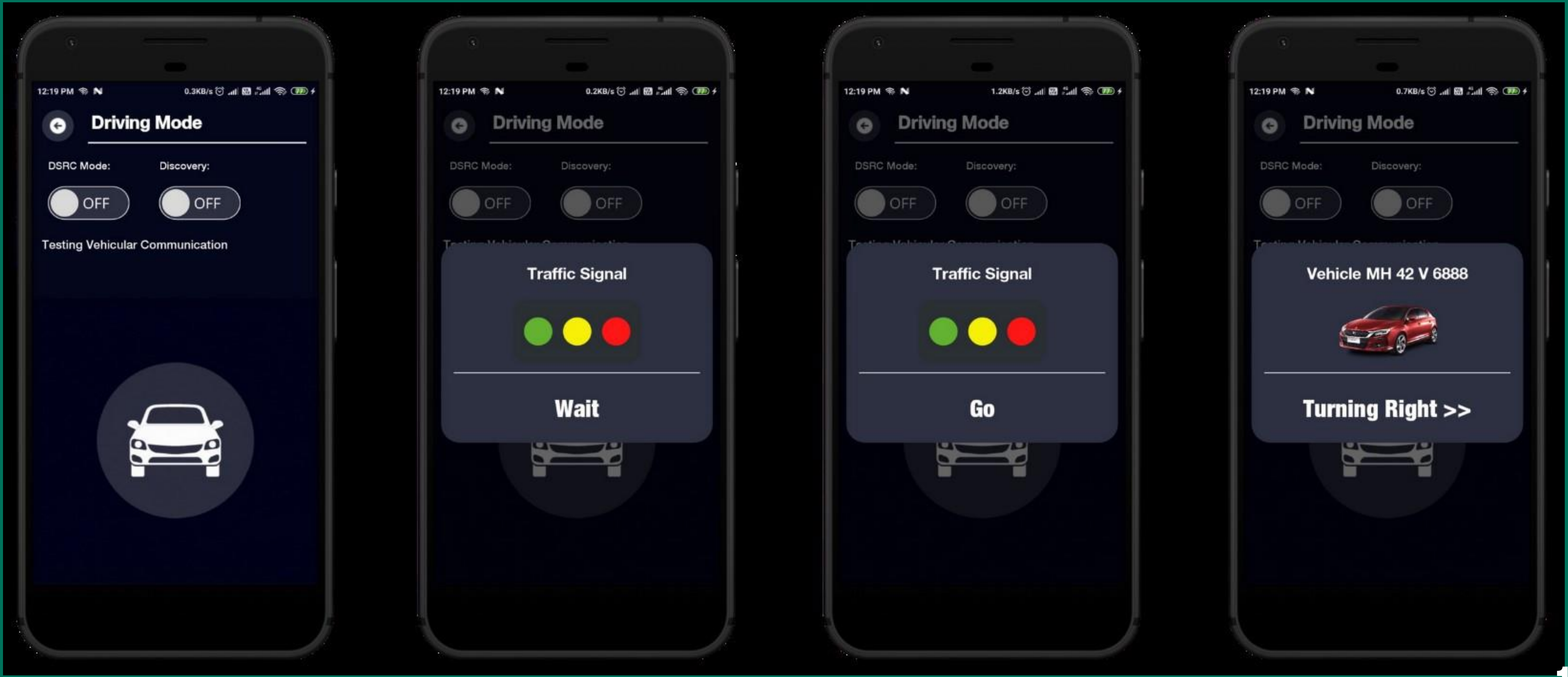
Realtime health monitoring. Make sure your bluetooth is on and vehicle is connected to device.

Health Status : All OK

Connection Status **Connected**	Barometric Pressure **101.00**
Engine RPM **1741.00**	MAF **16.09**
Intake Manifold Pressure **61.00**	Engine Coolant Temperature **89.00**
Speed **41.00**	Throttle Position **0.32**

Deployed in Android App

Example:



12:19 PM

Driving Mode

DSRC Mode: OFF

Discovery: OFF

Testing Vehicular Communication

12:19 PM

Driving Mode

DSRC Mode: OFF

Discovery: OFF

Traffic Signal

🟢 🟡 🔴

Wait

12:19 PM

Driving Mode

DSRC Mode: OFF

Discovery: OFF

Traffic Signal

🟢 🟡 🔴

Go

12:19 PM

Driving Mode

DSRC Mode: OFF

Discovery: OFF

Vehicle MH 42 V 6888

Turning Right >>

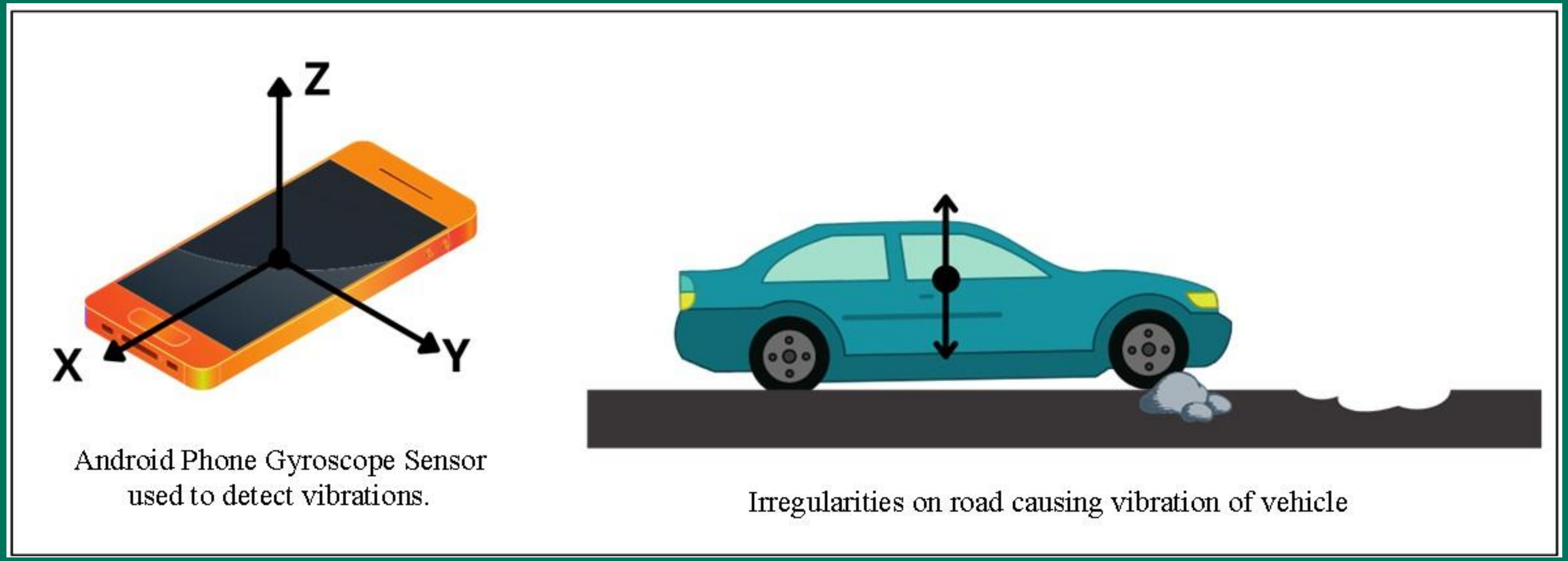
SECTION IV

Road Analytics using Smartphone App

Motivation

- In countries like India having a large network of roads, It’s a tedious task for government bodies to keep track of condition of roads in different parts of country.
- A centralized System that maintains realtime up-to-date status of condition of roads can be very helpful for planning developmental strategies.
- Motivation of Road Analytics lies here. In our proposed work, we developed as system that can collect this information automatically using smartphone based sensing technology.

Approach



Android Phone Gyroscope Sensor used to detect vibrations.

Irregularities on road causing vibration of vehicle

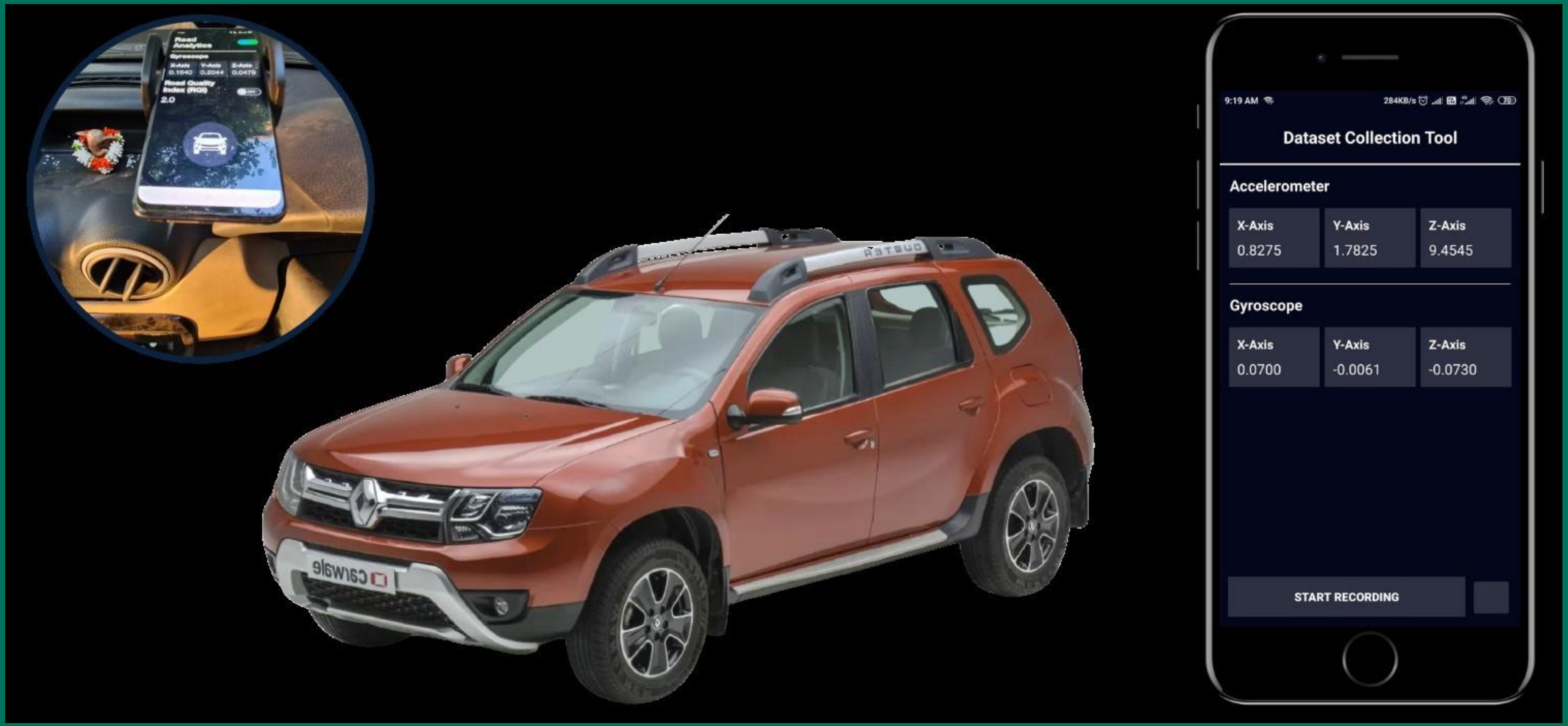
- Whenever vehicle moves on a damaged road, there are certain fluctuations or vibrations experienced inside the vehicle. These vibrations can be sensed using smartphone sensors.

- In the proposed work, gyroscope sensor in mobile phone was used to sense these vibrations and infer about the condition of roads.

Data Acquisition

Collection of labelled dataset

- An android app developed for data acquisition.
- App collected raw gyroscope readings along three axes and saved it in a csv file inside the internal storage.
- Three patches of roads were identified in outskirts of Mumbai city and dry run ofvehicle with our app was done manually.



9:19 AM

Dataset Collection Tool

Accelerometer

X-Axis	Y-Axis	Z-Axis
0.8275	1.7825	9.4545

Gyroscope

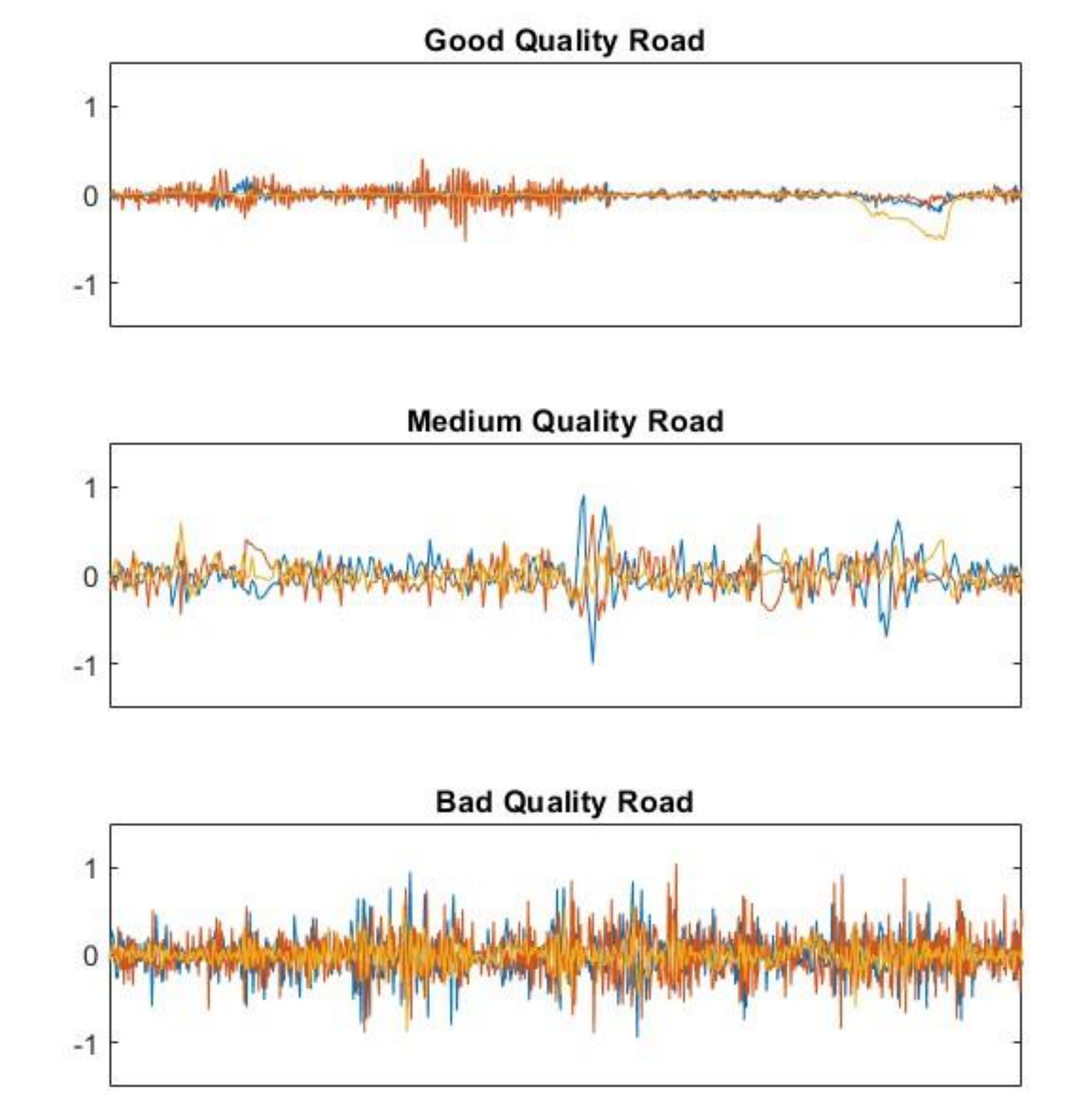
X-Axis	Y-Axis	Z-Axis
0.0700	-0.0061	-0.0730

Green Zone

Yellow Zone

Red Zone

Visualization of Collected Data



- Data saved in csv files collected for these three zones was plotted results are shown here.

- Motivation to continue work lies here.

Preprocessing

- Dataset contains raw Gx, Gy and Gz values

$$Geff = \sqrt{{G_x}^2 + {G_y}^2 + {G_z}^2}$$

- Filtering to remove noise – Moving average filter used.

$$\frac{1}{W} \quad \underset{i=-\frac{W}{2}+1}{\overset{W/2}{}} \quad G(t+i)$$

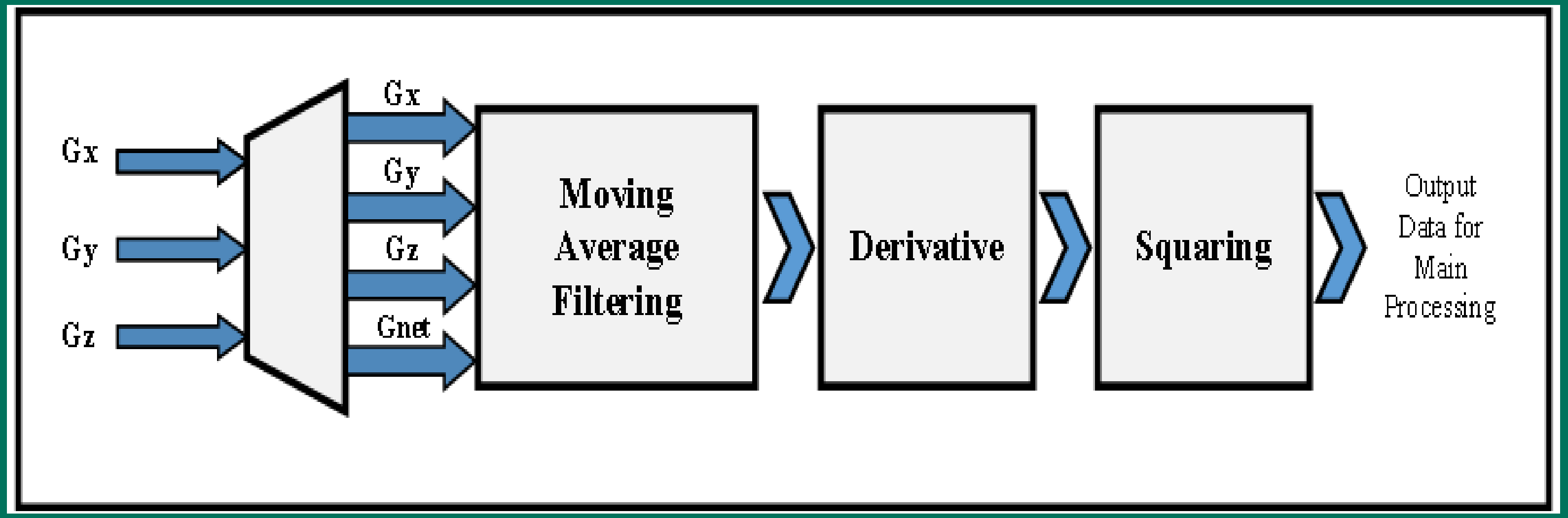
- Absolute values are of no interest and we are interested in rate of change in these values.

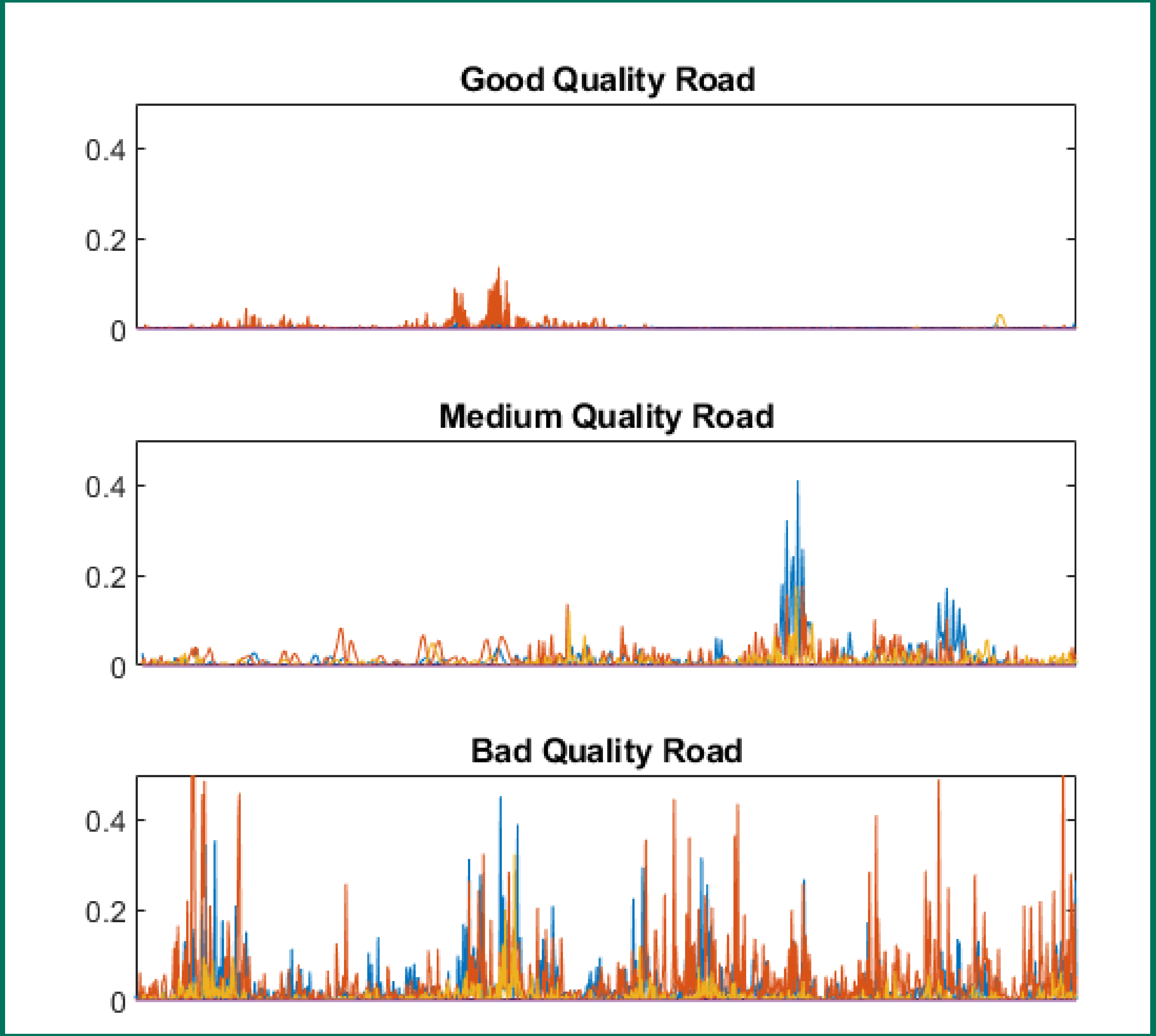
$$\frac{\partial G}{\partial t} = \frac{G(t+1)-G(t)}{\Delta t}$$

- Magnitude of change interested and not the sign

$$\left(\frac{\partial G}{\partial t}\right)^2$$

- Squaring also magnifies the difference between values.





Algorithms & Results

- Now the problem is simple classification problem. We have the data points, we need to classify them to one of the three zones –
  - Green Zone
  - Yellow Zone
  - Red Zone

- Three algorithms studied and implemented
  1. Thresholding
  2. K Nearest Neighbours
  3. Neural Network Model

Thresholding

- Simplest approach

- Two thresholds T1 and T2

  Green zone : $x < \tau_1$

  Yellow zone : $\tau_1 < x < \tau_2$

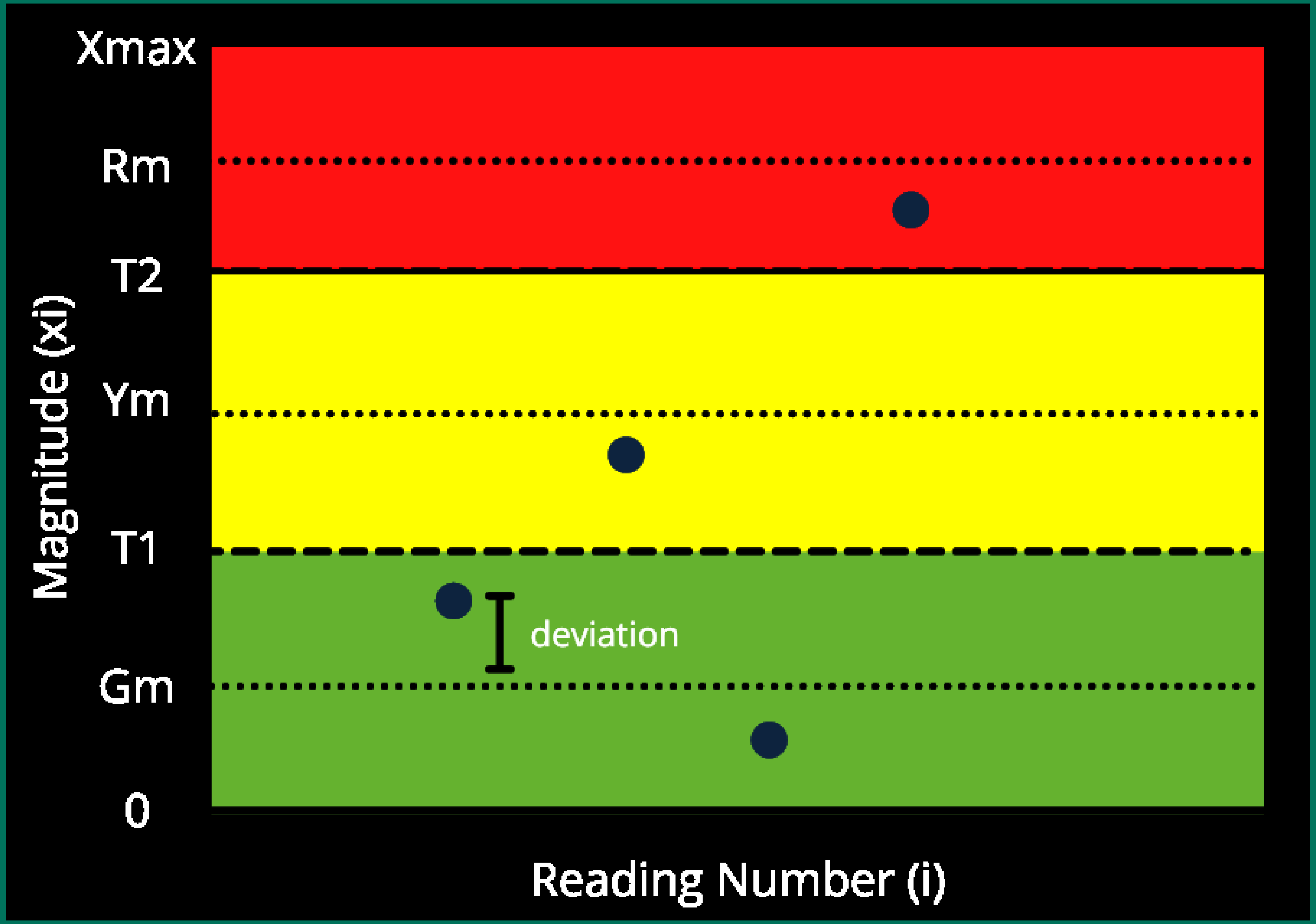
  Red zone : $x > \tau_2$

Total 4 features – G_x, G_y, G_z, G_{net}

Each feature will predict one out of three classes

Output is the majority of results predicted by 4 features.

In case of tie, mean deviation is compared to select the resultant class.



Results

Overall Accuracy : **72.29 %**

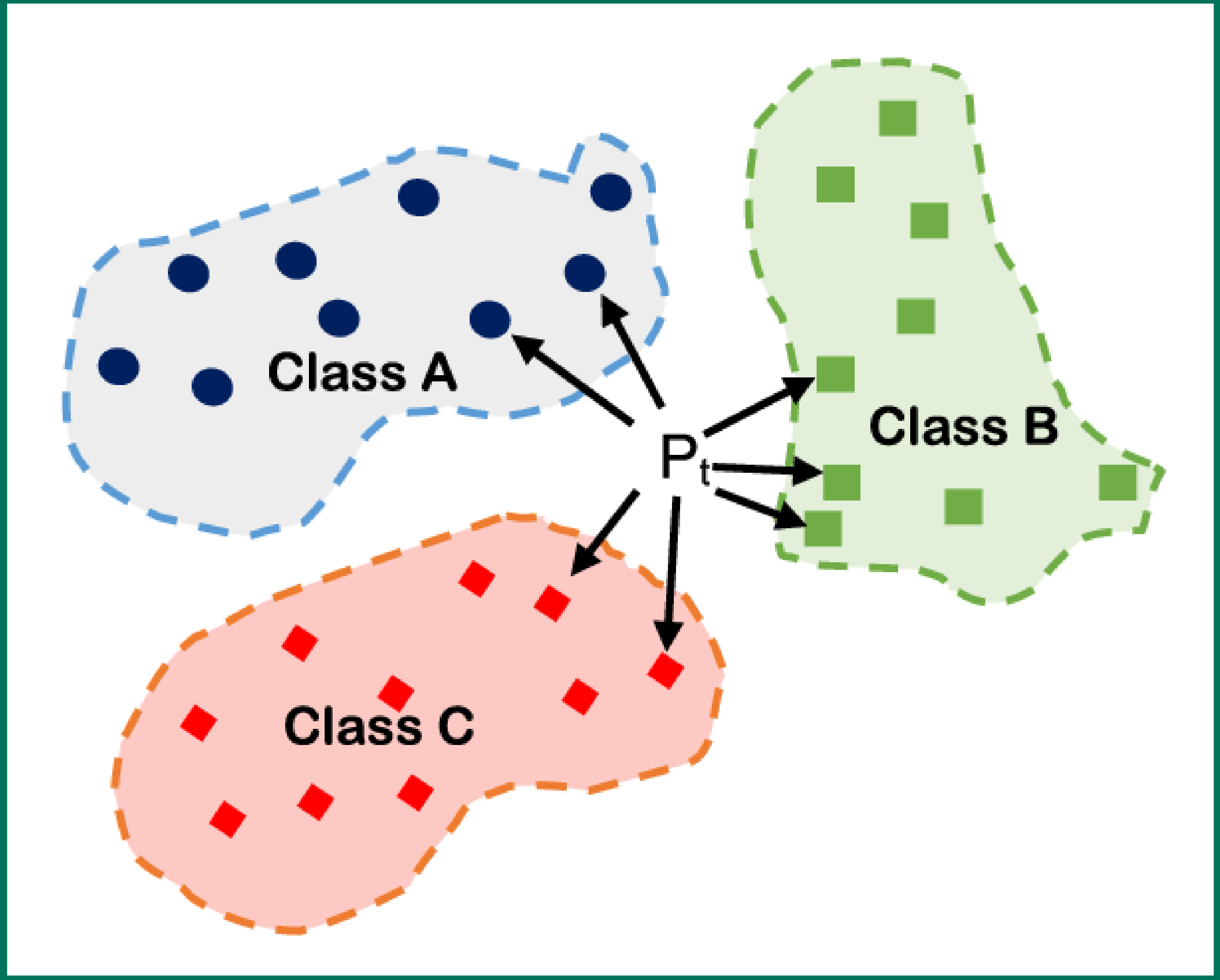
	True Green	True Yellow	True Red
Predicted Green	**98.33 %**	**46.99 %**	**9.29 %**
Predicted Yellow	**1.67 %**	**41.30 %**	**34.88 %**
Predicted Red	**0.00 %**	**11.71 %**	**55.83 %**

Good Accuracy in identification of green zone points

Poor Accuracy in identification of Red and Yellow zone points

K Nearest Neighbours

- Well known classification algorithm in literature.
- Works on the principle that similar things exist in close proximity
- Calculates the Euclidean distance of unlabelled data point which is to be classified from all other points in the training set and then select the K nearest points.
- Resultant class is majority of class to which these neighbours belong to.



Results

Overall Accuracy : **82.25 %**

	True Green	True Yellow	True Red
Predicted Green	**97.75 %**	**25.42 %**	**3.70 %**
Predicted Yellow	**0.73 %**	**46.15 %**	**46.15 %**
Predicted Red	**1.52%**	**28.43 %**	**91.09 %**

Good Accuracy in identification of Green zone and Red zone points

Poor Accuracy in identification of Yellow zone points

Neural Network Model

- Neural Network model is developed with an input layer, 4 hidden layers and an output layer.

- Hyperparameters of the model are tuned to achieve high accuracy.

- Problem of overfitting is countered by setting appropriate dropout rates.

- **Rectified Linear Activation Unit** or **ReLU** activation function was selected due to its linear behavior, computational simplicity, and the enhanced probability of optimization.

Results

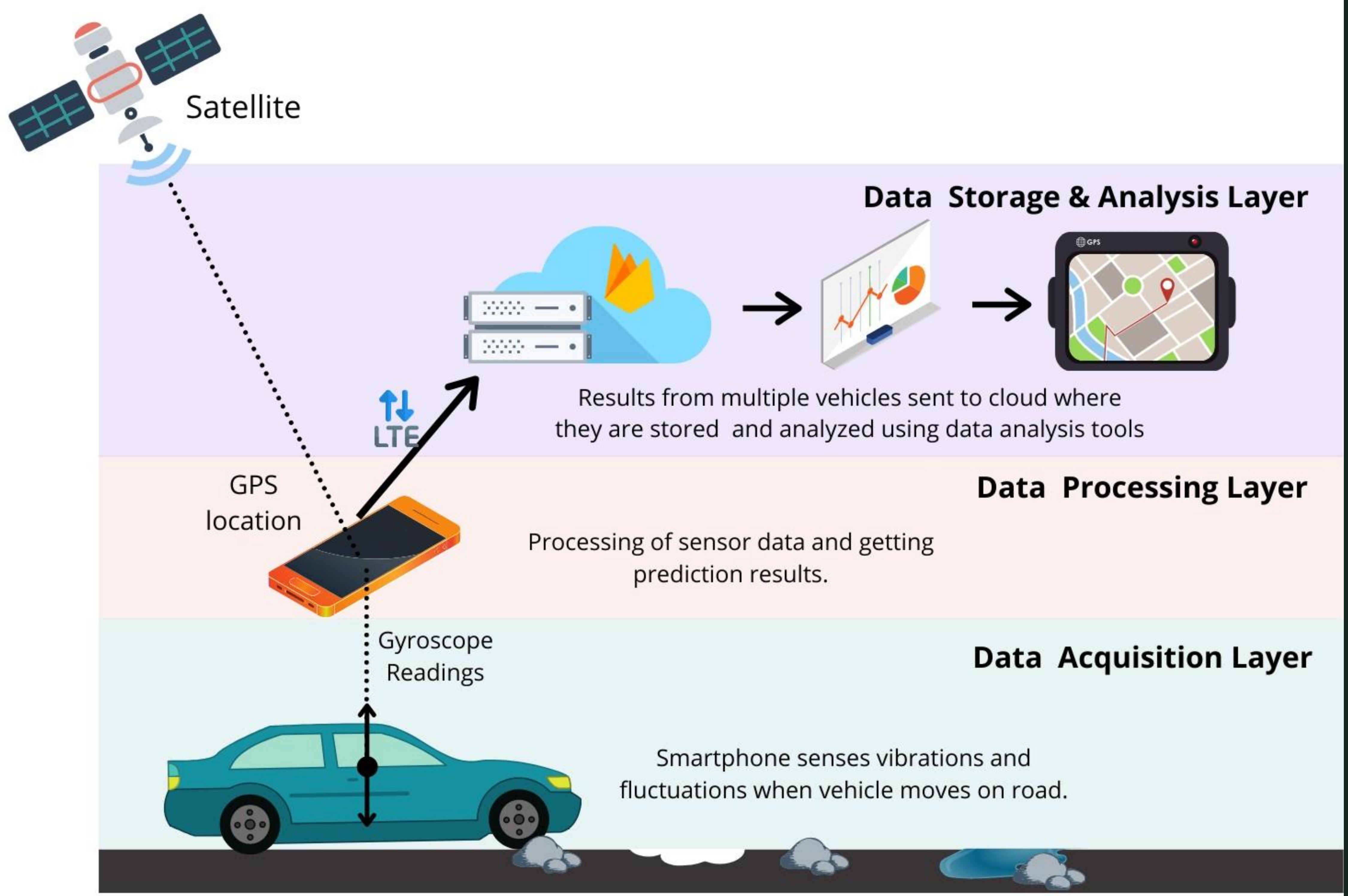
Overall Accuracy : **89.9 3%**

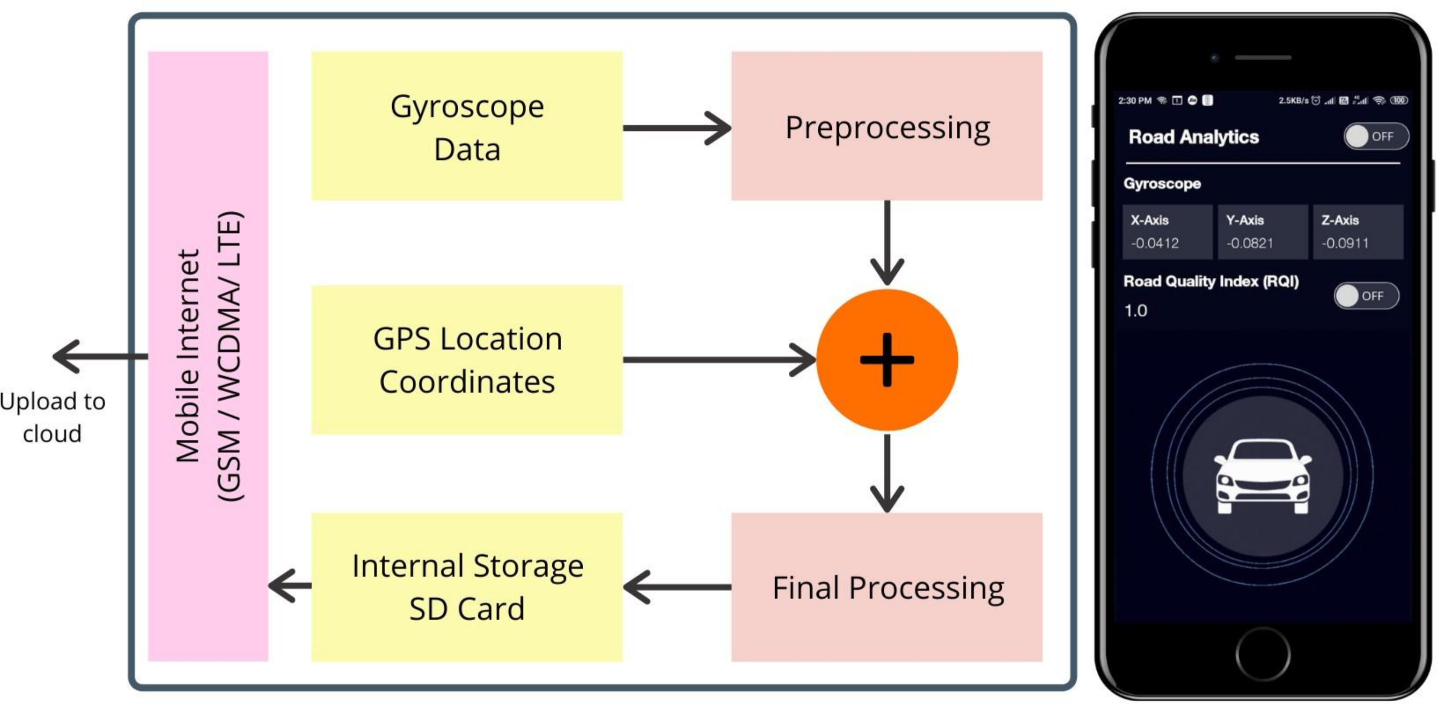
	True Green	True Yellow	True Red
Predicted Green	**95.04 %**	**0.89 %**	**0 %**
Predicted Yellow	**4.13 %**	**92.86 %**	**16.54 %**
Predicted Red	**0.82 %**	**6.25 %**	**83.46 %**

Good Accuracy in identification of Green zone , Red zone points as well as Yellow zone points.

Proposed System

Three layered System architecture proposed and prototype developed


Satellite
Data Storage & Analysis Layer
LTE
Results from multiple vehicles sent to cloud where
they are stored and analyzed using data analysis tools
GPS
GPS
location
Data Processing Layer
Processing of sensor data and getting
prediction results.
Gyroscope
Readings
Data Acquisition Layer
Smartphone senses vibrations and
fluctuations when vehicle moves on road.



Road Analytics OFF

Gyroscope

X-Axis	Y-Axis	Z-Axis
-0.0412	-0.0821	-0.0911

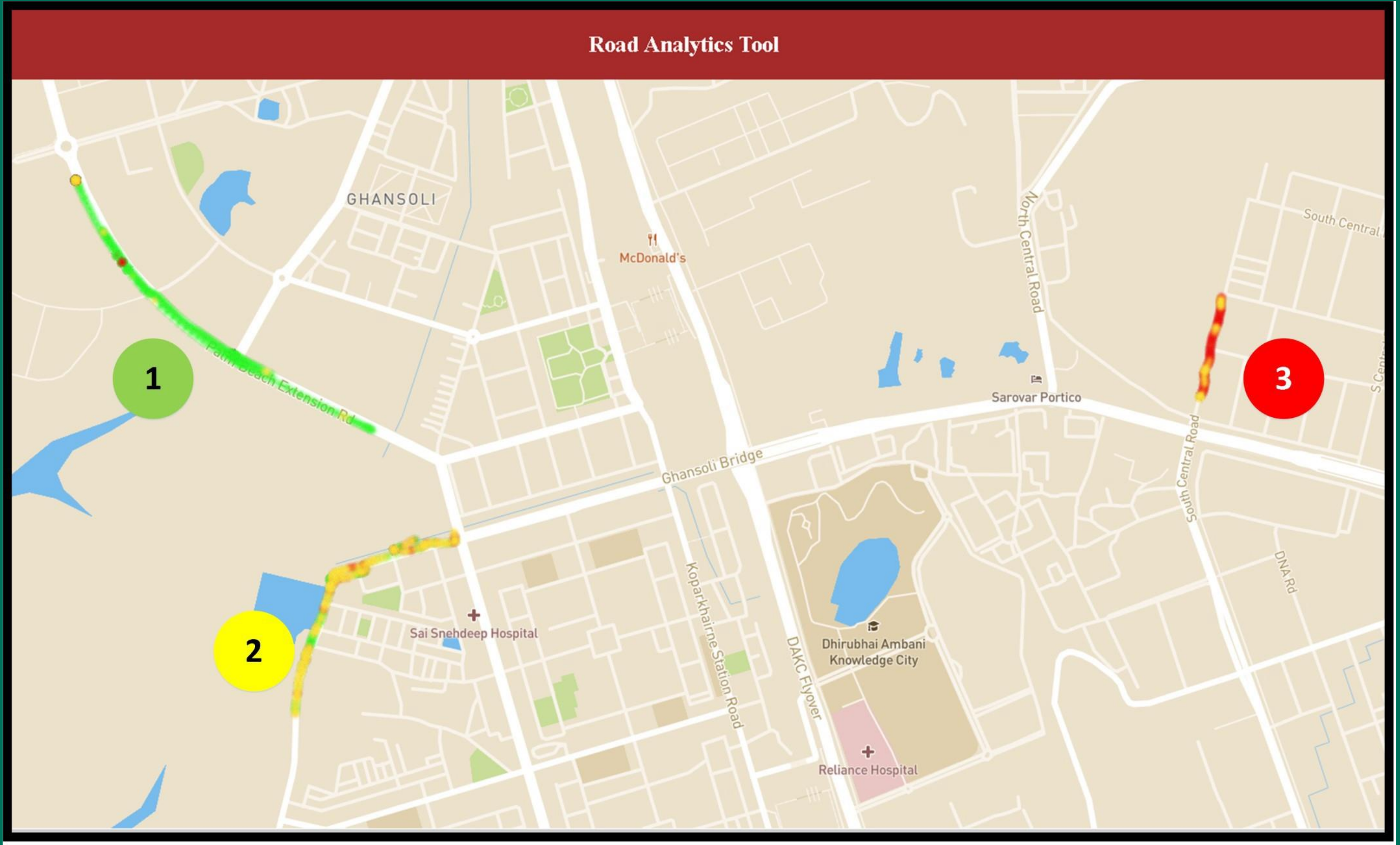
Road Quality Index (RQI) OFF

1.0

- Neural Network gave best accuracy and hence deployed in the android app.
- App works in background.
- Maps road analytics data to the gps location and sends it to the cloud where data is stored and can be analyzed using data analysis tools.
- Google Firebase cloud used as analytis cloud.

Proof of Concept

- On-field testing of the proposed system was carried out using a vehicle and the android device.
- The Neural Network model was deployed in the android application and the mobile device was placed inside the car.
- Test drive of the system was taken on a road of length about 2 km in the outskirts of Mumbai City as shown in figure.
- Route selected for testing was such that it had wide variations in road quality. The route was first traversed and regions of the good, medium, and bad quality were identified manually and noted along with their GPS coordinates.



Results observed on GUI screen

Green Zone

Yellow Zone

Red Zone

Results

Overall Accuracy : **84 %**

	True Green	True Yellow	True Red
Predicted Green	**97.91 %**	**22.60 %**	**0 %**
Predicted Yellow	**2.08 %**	**61.20 %**	**15.14 %**
Predicted Red	**0%**	**16.20 %**	**84.85 %**

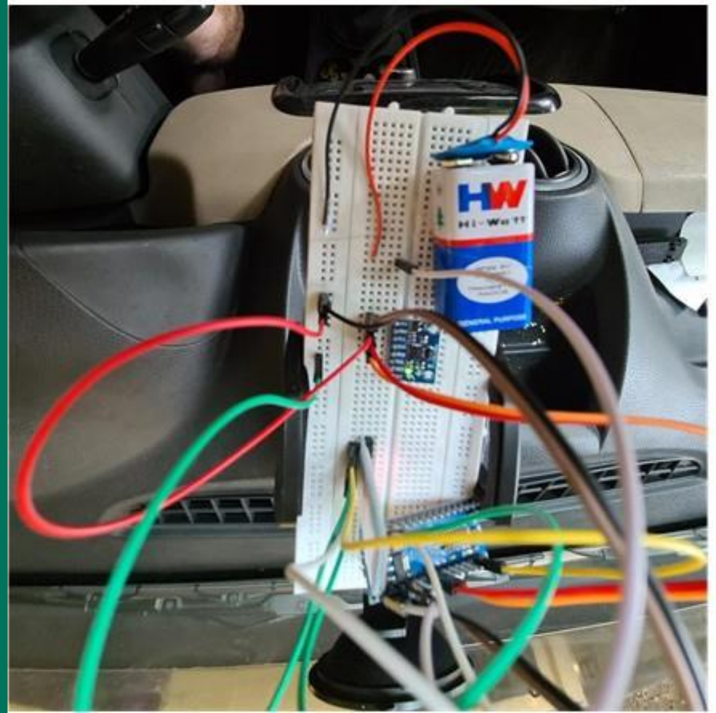
Overall good accuracy obtained

Hardware Implementation

Overall Accuracy : **86 %**

	True Green	True Red
Predicted Green	86.22 %	13.78 %
Predicted Red	13.68%	86.32 %

Hardware implementation of the system was carried out using arduino, MPU 6050 and battery.



IMU Setup

Green Zone

Red Zone

Novelty in Project

1

Unique Approach

Systems for vehicle health monitoring using smartphone application are not available in Indian markets.

2

Implementation

All algorithms, systems, applications involved in project are programmed, implemented, tested and deployed by us from scratch without any third party tool.

3

Dedicated Simulator

System is modelled and simulated using our own simulation tool developed from scratch using basic principles of computation modelling.

4

Supported by Results

Proposed system is no more a fictional system of Hollywood movies but now supported by satisfactory results for proof of concept.

Summary

1

Traffic Management

Almost 4 times reduction in waiting time and 3 times reduction in traffic congestion achieved by proposed system in simplest ratio algorithm.

2

Vehicular Communication

Vehicular communication via Smartphone WiFi P2P communication was achieved. Simple implementation on app.

3

Health Monitoring

Successful deployment of trained model in proposed app and detection of anomalies with accuracy of 99.79%

4

Smartphone Application

A dedicated android application for integrating all features together in a single app.

Appendix

Simulator Design

Orignal Source code of simulator design in JAVA programming language maintained up to date on:

https://github.com/Hrishi3331/IntersectionTrafficSimulator

Diagnostic Algorithms

Local Outlier Factor, Isolation Forest and Neural Network implemented in python and source code maintained at:

https://github.com/pushpak4166/Anomaly-Detection-new

Android App

Proposed application developed using android studio and source code maintained on :

https://github.com/Hrishi3331/ConnectedVehicleSmartphoneApp

Thank You!