Revolutionizing Urban Mobility: IoT-Enabled Smart Traffic Signal Optimization System

Abstract

Efficient traffic signal management remains a critical issue in modern urban environments. Traditional traffic signals operate on a fixed 60-second cycle, irrespective of the traffic density, often leading to inefficiencies and increased wait times. This paper introduces an innovative model that adjusts traffic signal intervals based on the real-time volume of vehicles on each road approaching an intersection. The core advantage of this system is its ability to significantly reduce driver wait times at traffic signals. This model utilizes clustering techniques derived from the K-Nearest Neighbors (KNN) algorithm to dynamically compute the necessary signal duration based on vehicle count inputs. Vehicle counts from each direction at an intersection are used to determine the precise signal timings needed. Case studies on traffic networks and real-time sub-networks have been conducted to assess the proposed model's performance.

Keywords: Internet of Things, traffic automation, K-Nearest Neighbors

Introduction

A pivotal component of the Internet of Things (IoT) in the realm of smart cities is the Intelligent Transportation System (ITS). ITS enhances both Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications, facilitating improved roadway usage without necessitating the expansion of road capacities or the construction of new routes. This advancement is made feasible through ITS, which leverages sophisticated information and communication technologies to mitigate traffic congestion and reduce the incidence of road accidents, particularly in densely populated urban areas.

Managing the timing of traffic signals is a crucial aspect of urban traffic control. Properly coordinated traffic signal timing can significantly diminish drivers' wait times, consequently reducing fuel consumption. This is achieved through the implementation of ITS. Within this framework, we employ Infrared (IR) sensors, which operate on the principles of the Infrared spectrum. An IR sensor comprises two main components: a transmitter and a receiver. The transmitter emits light, while the receiver continuously detects it. When the light transmission is interrupted, the counting process begins. This interruption, when the receiver no longer detects

the transmitted light, indicates the presence of an object between the transmitter and receiver, utilizing the line-of-sight concept.

Traffic signals serve as a mechanism to regulate vehicular flow. In recent years, transportation has gained immense significance for both logistics and everyday human activities, leading to an increase in the number of vehicles on the road. Consequently, traffic congestion and road accidents have become commonplace in bustling cities. Traffic signals offer an efficient, cost-effective, automatic, and equitable solution at critical junctions where vehicles must change directions, such as roundabouts, culverts, and busy pedestrian crossings.

Basic Concept

The project we have undertaken is the design of an 8-lane traffic controller. The fundamental objective of this design is to avert vehicular collisions by providing timely signals to various directions for predetermined intervals, after which the subsequent waiting vehicles receive the same attention. This cyclical process ensures systematic traffic control, thereby enhancing overall traffic management efficiency.

3. Implementation

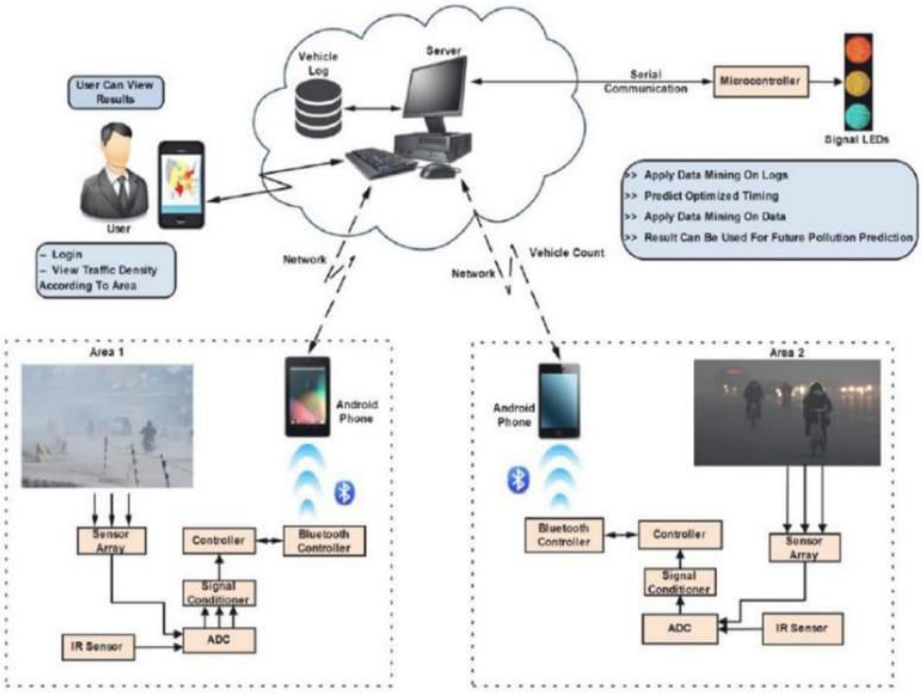


Fig. 1: TMS architecture diagram

Traffic Monitoring System–TMS using multiple IRs in IOT model to learn vehicles count in real time then updates signal timing of every side traffic lights according's to the predicting factors using KNN algorithms. It uses IRs to collect signals from numerous inputs on basis IOT model setup to learn vehicles count available in a traffic signal. So it will get cleared depends on the dependability of vehicles count to either increase or decreases timing of particular signals using KNN algorithms by data analysis.

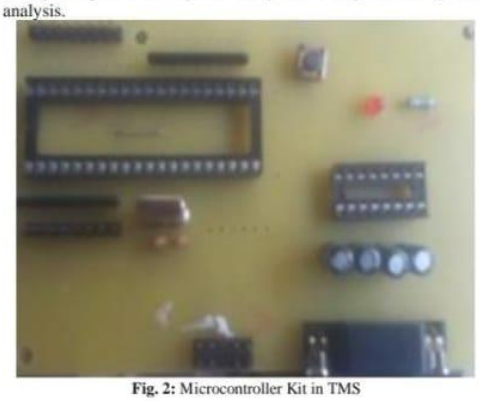


Fig. 2: Microcontroller Kit in TMS

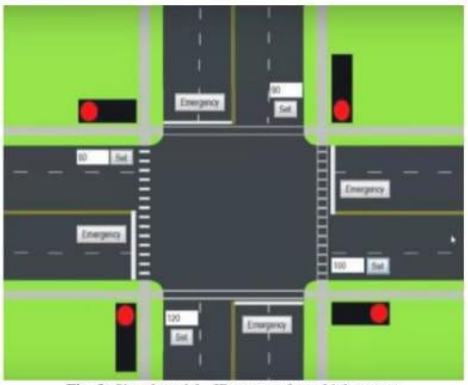


Fig. 3: Signal model - IR sensors for vehicles count

4. Modules

In this implemented application considered three major works as modules processes of the project.

These modules describes in detail about work done in the project which is listed below:

1. Android Application
2. Server Side Communication
3. User Communication

1. Android Application: In this Android 1st module about android kit utilized to transfer data from onside the micro-

controller to the other side server communication. Then it's coded for the android application. The server performing communication operation towards data transaction.

2. Server Communication: This module is used to perform algorithms computation using clustering technique from data given inputs using an android device, which is used to store all the previously stored data i.e., like the history of all the previous vehicles already used the same signal crossing. KNN model algorithm is utilized for clustering model extraction from the given inputs, which is working on non-parametric methods for classification & regression model of data analyses. This input which contains k closest outputs of training examples for the feature space. K-NN working on the instance-based learning those function is approximated as local values.This project always collects data for four side road crossing as the vehicle counts. According to the count as inputs, this algorithm going to decide the signal timing intervals to get higher time limits for that particular signal to avoid the traffic queuing in a dense number of vehicles.
3. User Communication: From the user's communication, they can able to view required information about traffic condition in that particular area. If the user decided to travel in the particular area, then they can use current scenario of traffic conditions by this android application.

5. TMS results

Fig. 4: TMS mobile application

Fig. 5: TMS mobile application

Android Mobile application for Traffic Monitoring System –TMS which is using multiple IRs for IOT model. To check IRs sensors status particular pages used from the admin side, Client Users can able to see traffic flow and timing for that particular road lane.

Fig. 6: TMS using IR in IOT

TMS results are displayed for user and admin to monitor traffic flow in particular route using multiple IRs in IOT model. Client Users checking their traffic flow as per their wishes to know about which lane and path-usage in that area nearby to them.

Fig. 7: TMS using IR in IOT

Four IR case study results from TMS apps system are displayed to understand monitoring of traffic flow in any particular route which based on IRs. Timing of any signal is varying as per KNN logical methodology algorithms purely depends on IR data captured in real time scenario to updates the timing of that traffic signals

6. Future scope

In the future advancements of this TMS, a model ambulance can able to communicate with all base station to get an easy free lane to rush up reaching the hospital on time for needy people. So such scenarios signal automatically is cleared with its arrival schedule.

7. Conclusion

TMS -Traffic Monitoring Signal timing has been developed by using multiple features of hardware components in IOT. Traffic optimization is achieved using IOT platform for efficient utilizing allocating varying time to all traffic signal according to available

Proposed System

The quotidian challenge of traffic congestion has become an increasingly formidable issue. Currently, automation systems designed to ameliorate such problems are conspicuously absent in India. There exists a pressing need to leverage the Internet of Things (IoT) within traffic signal monitoring and control systems, thereby ushering in a paradigm of advanced traffic management. The envisaged system aspires to

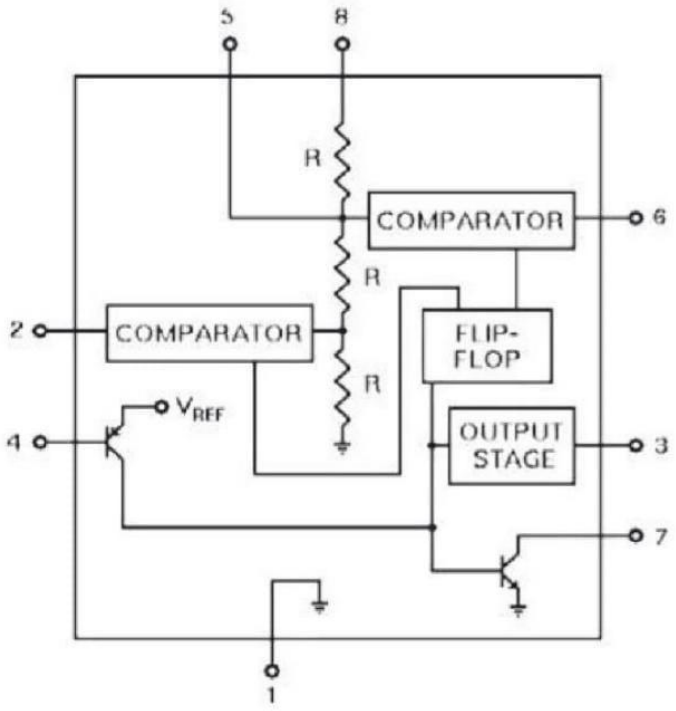
function with heightened intelligence, incorporating sophisticated control features applicable to all four directions of traffic flow.

In urban areas where vehicular traffic is dense, it is imperative to delineate a priority hierarchy within the Traffic Management System (TMS). This hierarchy would be contingent on traffic density, assigning either lower or higher priority based on vehicle count. The TMS must be equipped to adeptly manage traffic flow according to the vehicular density specific to each area. To this end, each road lane must be outfitted with Infrared (IR) sensors capable of monitoring and capturing vehicular data.

The proposed system allocates signal durations predicated on the vehicle count data collected from these IR sensors. The system's architecture necessitates the deployment of multiple IR sensors, a microcontroller for automation control, a Bluetooth controller, an Android mobile device, and a PC server. Each IR sensor comprises an IR transmitter and receiver, strategically positioned to monitor traffic in both directions of a road lane. This configuration ensures the system can dynamically adjust signal timings to mitigate traffic congestion

effectively.

BLOCK DIAGRAM



Advantages

Economical Microcontrollers: The deployment of microcontrollers, which are notably inexpensive, renders the system cost-effective. The usage of batteries in both the transmitter and receiver minimizes power consumption, facilitating effortless installation.

High-Speed and Precision System: The system boasts superior speed and precision. The adoption of Field-Programmable Gate Arrays (FPGAs) ensures economic feasibility and system flexibility.

Reduced Waiting Time: By decreasing the waiting time at traffic signals, the system effectively obviates the need for human intervention. It seamlessly transitions to a conventional system when the intelligent system is unnecessary, guaranteeing uninterrupted traffic flow during emergency situations.

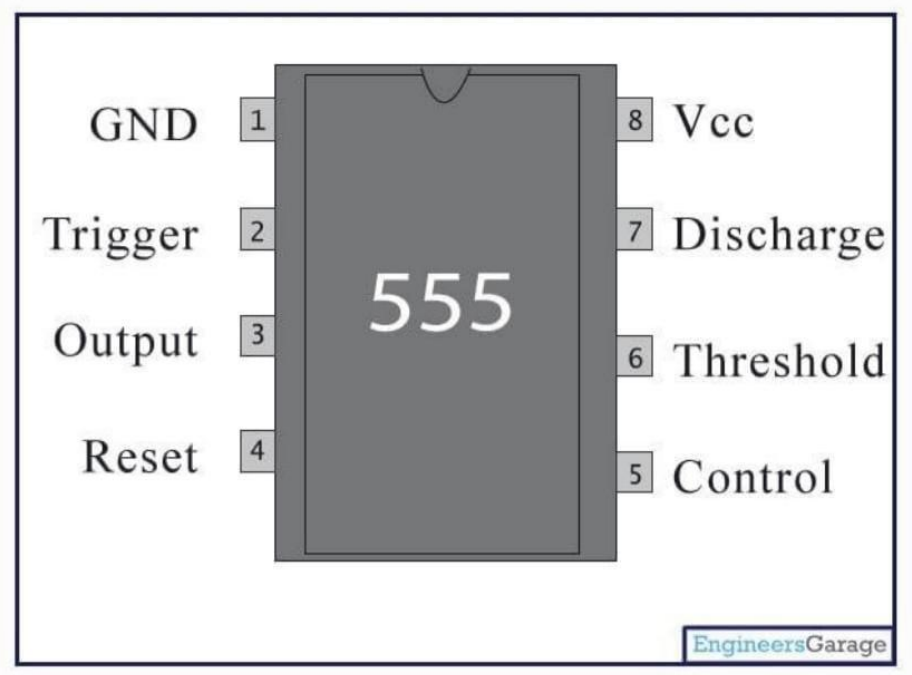
Disadvantages

Line-of-Sight Requirement: The necessity for the transmitter and receiver to maintain a direct line of sight significantly reduces system flexibility. Additionally, high-intensity lighting can interfere with the infrared signals.

Complexity and Cost: The system's complexity is heightened due to the incorporation of sensors, resulting in increased costs. A constant connection to a database is imperative for optimal functionality.

Maintenance and Priority Setting: The deployment of photoelectric sensors incurs high maintenance costs due to exposure to rugged external conditions. Furthermore, setting traffic priorities in advance is necessary, but it is often impractical to ascertain such priorities ahead of time.

PIN DIAGRAM



Component Description: TIMER

555 Timer IC

In our project, the 555 Timer IC has been employed as the timing component.

555 IC

The 555 Timer IC, renowned for its simplicity yet remarkable versatility, has endured the test of time and evolved into various technological incarnations. Presently, the primary iterations are the original bipolar version and the more contemporary CMOS counterpart. The principal distinctions between these versions lie in their power consumption requirements and their operational frequency limits.

The 555 Timer functions as a TTL digital logic circuit within the controller assembly, generating a periodic square wave signal. The temporal characteristics and duty cycle of this signal are contingent upon the resistors and capacitors interfaced with the timer.

Applications

Precision Timing: Utilized for exact timing applications.

Pulse Generation: Employed to generate precise pulse signals.

Sequential Timing: Applied in scenarios requiring sequential timing.

Time Delay Generation: Facilitates the generation of time delays.

Pulse Width Modulation: Used in pulse width modulation tasks.

Pulse Position Modulation: Applied for pulse position modulation.

Linear Ramp Generator: Functions as a linear ramp generator.

pseudo code

1. Initialize the system

   - Set up IR sensors on all road lanes

   - Initialize microcontroller and communication interfaces (Bluetooth, Android device, PC server)

2. Start monitoring vehicle counts
   - For each road lane:
     - Read vehicle count from IR sensors
     - Store the count in a variable

3. Determine signal timing based on vehicle counts
   - For each lane:
     - If vehicle count is high:
       - Assign a longer green light duration
     - If vehicle count is low:
       - Assign a shorter green light duration

4. Update traffic signal timings
   - Send signal timing data to traffic lights
   - Adjust signal intervals according to the priority assigned based on vehicle count

5. Communicate with Android device and PC server
   - Send real-time data to the Android device for user interface
   - Send processed data to PC server for further analysis and reporting

6. Repeat the monitoring and adjustment process
   - Continuously update vehicle counts and adjust signal timings as needed

7. End of system operation
   - Shut down sensors and controllers
   - Save data logs for review

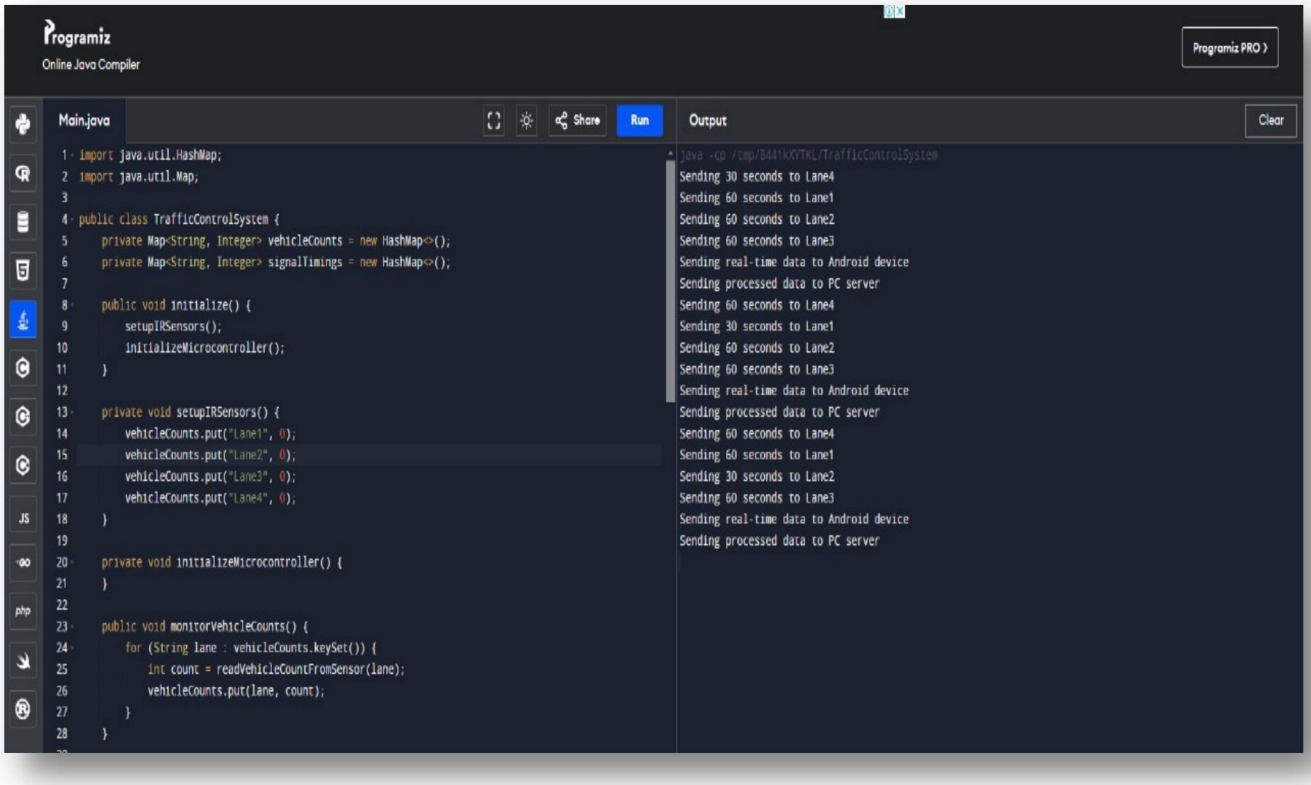
PROGRAM

```
import java.util.HashMap;
import java.util.Map;

public class TrafficControlSystem {
    private Map<String, Integer> vehicleCounts = new HashMap<>();
    private Map<String, Integer> signalTimings = new HashMap<>();

    public void initialize() {
        setupIRSensors();
        initializeMicrocontroller();
    }

    private void setupIRSensors() {
        vehicleCounts.put("Lane1", 0);
        vehicleCounts.put("Lane2", 0);
        vehicleCounts.put("Lane3", 0);
        vehicleCounts.put("Lane4", 0);
    }

    private void initializeMicrocontroller() {
    }

    public void monitorVehicleCounts() {
        for (String lane : vehicleCounts.keySet()) {
            int count = readVehicleCountFromSensor(lane);
            vehicleCounts.put(lane, count);
        }
```

```
}

private int readVehicleCountFromSensor(String lane) {
    return (int) (Math.random() * 100);
}

public void determineSignalTiming() {
    for (String lane : vehicleCounts.keySet()) {
        int count = vehicleCounts.get(lane);
        int duration = (count > 10) ? 60 : 30;
        signalTimings.put(lane, duration);
    }
}

public void updateSignalTimings() {
    for (Map.Entry<String, Integer> entry : signalTimings.entrySet()) {
        String lane = entry.getKey();
        int duration = entry.getValue();
        sendSignalTimingToTrafficLight(lane, duration);
    }
}

private void sendSignalTimingToTrafficLight(String lane, int duration) {
    System.out.println("Sending " + duration + " seconds to " + lane);
}

public void communicateWithDevices() {
    sendRealTimeDataToAndroidDevice();
```

```
    sendProcessedDataToPCServer();
}

private void sendRealTimeDataToAndroidDevice() {
    System.out.println("Sending real-time data to Android device");
}

private void sendProcessedDataToPCServer() {
    System.out.println("Sending processed data to PC server");
}

public void run() {
    initialize();
    while (true) {
        monitorVehicleCounts();
        determineSignalTiming();
        updateSignalTimings();
        communicateWithDevices();
        try {
            Thread.sleep(10000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

public void shutdown() {
    System.out.println("Shutting down system");
```

```
    }

    public static void main(String[] args) {
        TrafficControlSystem system = new TrafficControlSystem();
        Runtime.getRuntime().addShutdownHook(new Thread(() -> system.shutdown()));
        system.run();
    }
}
```



Future Scopes

This paper delineates a multi-modal Smart Traffic Control System (STSC) tailored for smart city infrastructure, with extensive applicability in intelligent transportation systems within urban environments. The principal constituents of the proposed STSC encompass the Roadside Unit (RSU) controller, Onboard Unit (OBU), signal controller, and cloud center. This system underpins diverse smart city ITS applications, including Electric Vehicle Signal Priority (EVSP), Transit Signal Priority (TSP), eco-driving,

Adaptive Traffic Signal Control (ATSC), pre-timed signal control, and Road-to-Vehicle (R2V) message dissemination. The RSU controller stands as the focal point of this study, with a comprehensive discourse on system architecture, middleware integration, peripheral hardware modules, and control algorithms. The STSC framework is designed in adherence to the Urban Traffic Control Protocol V3.0, ensuring compatibility with conventional traffic signal controllers, facilitating rapid and cost-efficient deployment. An innovative traffic signal scheme is particularly employed.

Conclusion

This project epitomizes a highly efficacious methodology for traffic optimization, achieved through the recalibration of threshold parameters for real-time applications. The system is designed to regulate traffic on four-way intersections in accordance with predefined traffic control barriers. The proposed mechanism is poised to contribute significantly to the infrastructure of a developed nation by mitigating traffic congestion and facilitating timely access for emergency vehicles. Consequently, this advanced system promises to enhance traffic management with a greater degree of automation.

References

Li, Z., Shahidchpour, M., Babrumirad, S., & Khodaci, A. (2017). "Optimizing traffic signal settings in smart cities." IEEE Transactions on Smart Grid, 8(5), 2382-2393.

Singh, I., Tripathi, S., & Arora, H. (2009). "Time optimization for traffic signal content using genetic algorithm." International Journal of Recent Trends in Engineering, 2(2), 4-6.

Pable, S.N., Welekar, A., & Gaikwad-Patil, T. (2014). "Implementation of Priority Based Signal Management in Traffic System." International Journal of Engineering Research and Technology (IJERT), 3(5), 1679-1582.

Milanes, V., Villagra, J., Godoy, J., Simo, J., Pérez, I., & Onieva, E. (2012). "An intelligent V2I-based traffic management system." IEEE Transactions on Intelligent Transportation Systems, 13(1), 49-58.

Krishnan, S. (2016). "Traffic Flow Optimization and Vehicle Safety in Cities." International Journal of Innovative Research in Science, Engineering and Technology, 5(5), 7814-7820.