Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

IoT Based Smart Traffic Control System with Emergency Vehicle Identification and Real-Time Web Monitoring

Dr. A. V. Santhosh Babu[1], D. Vimal Kumar[2], K. Surya Prakash[3], A. Anton Maria Jones[4], S. Logeswar[5], S. Mokith[6]

[1,2] Professors, Department of Information Technology, Hindusthan Institute of Technology, Coimbatore, Tamil Nadu, India.

[3,4,5,6] Students, Department of Information Technology, Hindusthan Institute of Technology, Coimbatore, Tamil Nadu, India.

[1] dr.santhoshbabuav@hit.edu.in, [2] vimalkumar.d@hit.edu.in, [3]720822106056@hit.edu.in, [4]720822106006@hit.edu.in, [5]720822106022@hit.edu.in, [6]720822106029@hit.edu.in

---***---

Abstract- This project is about a traffic control system that uses the Internet of Things or IoT to make traffic flow better in cities. It helps emergency vehicles get through quickly. The system uses three IR sensors at each intersection to check how busy the traffic is. These sensors can tell if the traffic is low, medium or high. Depending on how busy it's the green light stays on for 5, 10 or 15 seconds. There's also an RF module that works at 433 MHz It can detect when an emergency vehicle is coming. When it does it makes a corridor so the emergency vehicle can pass through without stopping. The system uses an Arduino Mega 2560 microcontroller. It also sends traffic data to the Thing Speak cloud. This way people can monitor the traffic in time. We also made a web dashboard using HTML, CSS, JavaScript and Node.js. This dashboard has maps shows data in a visual way and lets users export data. Our system is pretty accurate. It can detect things 97.3% of the time. It also reduces the waiting time by 58%. The system helps make traffic flow better and gets emergency vehicles where they need to go. The IoT-based traffic control system is a tool, for cities. It uses technology to make a big difference.

Keywords - IoT, Smart Traffic Control System, Emergency Vehicle Detection, IR Sensors, RF Module, Arduino Mega 2560, Dynamic Signal Timing, ThingSpeak Cloud, ESP8266, Real- Time Web Monitoring, Traffic Density, Green Corridor, Intelligent Transportation System

1. INTRODUCTION

Urban traffic congestion is a problem in cities all over the world. It causes longer commute times, higher fuel consumption and more pollution. Traffic lights usually follow a fixed schedule that doesn't adjust to changing road conditions. This leads to delays during both busy and quiet hours. Ambulances often get stuck in these delays, which can affect survival rates.

We introduce a traffic control system that uses the Internet of Things (IoT) to address these issues. The system monitors traffic in time and adjusts the traffic signals accordingly. It uses sensors to measure traffic density at each junction categorizing it as low, medium or high. Based on the density the green light duration is set to 5 seconds, 10 seconds or 15 seconds. For ambulances a radio frequency module detects approaching vehicles and instantly provides a corridor with signal changes happening in under one second.

The system uses an Arduino Mega 2560 microcontroller and an ESP8266 module to send data to the ThingSpeak cloud platform. This allows for access. A custom web dashboard built with HTML, CSS, JavaScript and Node.js offers features such as user login with CAPTCHA and OTP verification. The dashboard has interactive mapping, graphical data representation and report export options in PDF and Excel formats. Our testing results show a detection accuracy of 97.3% and a 58% improvement, in waiting time compared to conventional systems.

2. LITERATURE REVIEW

Traffic congestion has turned into a major concern in urban areas around the world. It has increased the time of traveling, fuel consumption, and pollution. Conventional traffic control systems with a predetermined timing pattern are not effective to deal with changing traffic patterns, particularly during peak hours and emergency conditions. In recent times, many researchers have introduced intelligent traffic management systems by applying a variety of technologies such as VANET, IoT, computer vision, and AI.

Author (Year)	Technology / Method	Key Contribution

Bairi et al. (2025)	VANET	Enables emergency vehicles to communicate with traffic signals for priority passage
Hanif et al. (2025)	AI with Transfer Learning	Detects ambulances using cameras and changes signal timings automatically
Islam (2025)	AI and Machine Learning	Dynamically adjusts signal timings based on real-time traffic patterns
Author (Year)	Technology / Method	Key Contribution
Arefin et al. (2025)	Sensors and Intelligent Algorithms	Detects accidents in real-time and notifies authorities
Kheder and Mohammed (2024)	IoT and Deep Learning	Analyzes traffic patterns using sensor-collected data

3. PROPOSED SYSTEM

The proposed system also includes the integration of smart traffic management through the use of various IoT-based technologies to achieve the following: the central processing of the system through the use of an Arduino Mega 2560 microcontroller; cloud computing through the integration of an ESP8266 Wi-Fi module; the use of radio frequency modules to detect emergency vehicles; and the use of infrared sensors to measure the density of the traffic. The system also adjusts the time of the traffic signal and provides priority for emergency vehicles through the green corridor.

BLOCK DIAGRAM:

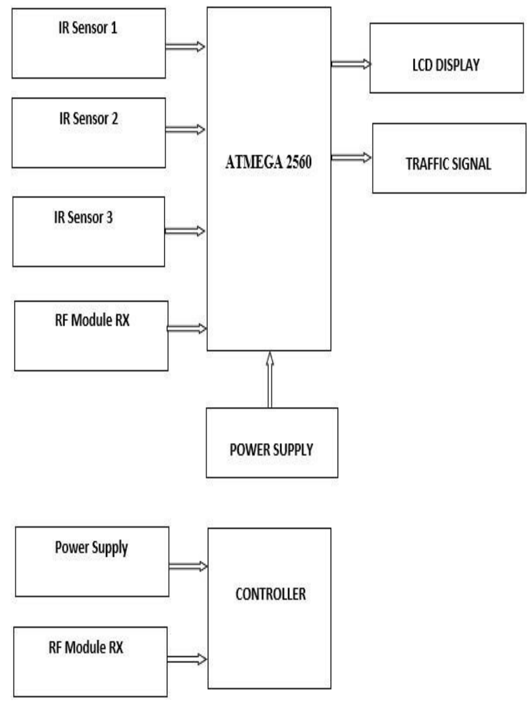

Fig. 1: Proposed System Block Diagram

- **System Architecture**

The traffic management system is constructed of an Arduino Mega 2560 for processing, three infrared sensors to estimate traffic density, a 433 MHz RF receiver module and an emergency vehicle RF transmitter to control the flow of traffic, and green, yellow, and red LED traffic signal controllers for the four directions (North, East, South, and West). The system provides motorists with real-time updates and countdown timers via a 16x2 Liquid Crystal Display (LCD). All components receive power from 12V and 5V regulated power supplies.

Fig. 2: IR Sensor

- **Traffic Density Detection Mechanism**

To detect the build-up of the traffic, there are three IR (infrared) sensors placed at specific intervals before the stop line. In the absence of vehicles on the line, all the sensors will be active and send the signal to the system (HIGH). In the presence of build-up of the traffic line, the sensors will be activated one by one. In the case where only the first sensor is activated, the system identifies the traffic line as low and gives a green signal for 5 seconds. In the case where the first and second sensors are activated; the system identifies the traffic line as medium and gives a green signal for 10 seconds. In the case where all the sensors are activated, the system identifies the traffic line as high and adjusts the green signal to 15 seconds to allow sufficient time to clear the traffic line.

Fig. 4: Example of placement of sensor for detection of vehicle density

- **Emergency Vehicle Priority System**

RF Transmitters at 433 MHz are mounted in emergency vehicles like ambulances. When the vehicle is close to an intersection, the transmitter emits an identification signal at a specified frequency. Traffic signal RF receiver modules can capture the identification signal within a range of 30 meters. When a receiver is activated, the corresponding microcontroller bypasses the preset traffic light sequence and turns the applicable traffic light to green while all the other traffic lights are turned to red. For this purpose, the traffic light control system has an LCD screen that displays the message "Ambulance Detected GO".

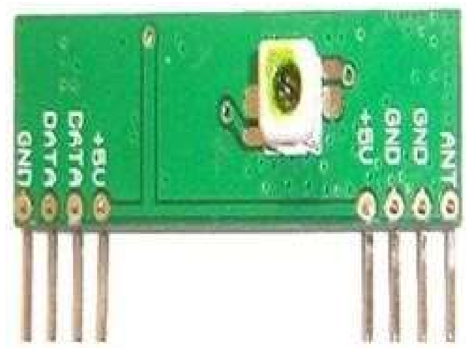

Fig. 5: Emergency vehicle detection receiver.

The response time of the system from signal identification to traffic light signal change is less than 1 second.

- **Cloud Integration and Web Dashboard**

The traffic density data and emergency vehicle counts and signal status are sent to the ThingSpeak cloud platform using the WiFi module. The data is uploaded every thirty seconds. Stored in six fields that can be configured which makes it possible to look at trends over time and see what is happening in real time. A web dashboard was made using Node.js and MongoDB and Chart.js and Leaflet.js which allows people to monitor things from away.

The web dashboard has important features, including Cloud Integration and Web Dashboard security measures like user authentication with CAPTCHA and OTP verification Traffic density, emergency vehicle count, and traffic signals are sent to the ThingSpeak platform using the ESP8266 WiFi module.

Data is sent at a regular interval of 30 seconds. Six fields are used to store the sent data, allowing for historical trend analysis. A web interface for comprehensive remote monitoring is created using Node.js, MongoDB, Chart.js, and Leaflet.js. Some of the prominent features of the web interface include user authentication using CAPTCHA and OTP, display of the positions of emergency vehicles using a map, display of traffic trends using a real-time graph, and generation of reports using export. Role-based access control for users includes admin, police, and operator.

Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

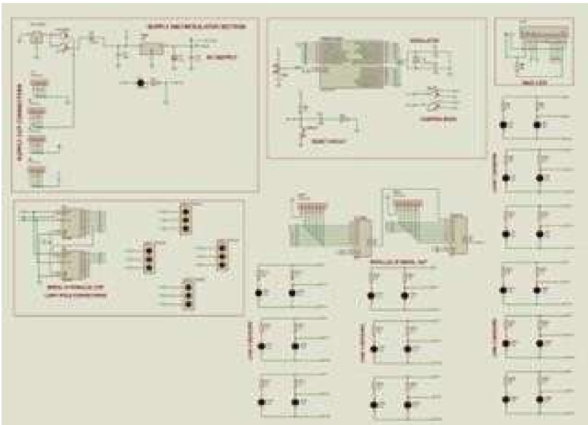

Fig. 6: Circuit Diagram.

The Cloud Integration and Web Dashboard also have maps that show where the emergency vehicles are and it has charts that show what is happening with traffic in time and people can export data to make reports, in PDF and Excel formats. The Cloud Integration and Web Dashboard have levels of access so admin and police and operator users can only do what they are supposed to do.

Fig. 7: ThingSpeak Cloud Data Visualization

4. IMPLEMENTATION

The smart traffic control system is composed of various parts such as the hardware, programming, cloud integration, and development of the web dashboard. Each of these parts is well connected to ensure the smooth working of the smart traffic control system in time.

- **Setting Up the Hardware**

Firstly, we assemble the microcontroller, which is Arduino Mega 2560 and is the brain of the smart traffic control system. Then we place the IR sensors at a distance of 5 meters, 10 meters, and 15 meters from the stop line and connect them to the microcontroller to sense the traffic density. Next, we connect twelve LEDs to the microcontroller to show the green signal for the four directions of the road: North, East, South, and West. Then we connect the receiver module to the microcontroller to sense the emergency vehicles.

Next, we connect the ESP8266 WiFi module to the microcontroller to connect to the cloud platform. Then we connect a 16x2 LCD to the microcontroller to show the status of the smart traffic control system in time. Each of these parts is connected to a power supply to function properly.

The RF receiver module for 433 MHz is connected to digital pin 48 for the detection of emergency vehicles. The ESP8266 WiFi module is connected to the Arduino for serial communication. Pins 18 and 19 are connected for the transmission of cloud data. A 16x2 LCD display with I2C communication is connected to pins 20 and 21 for the display of real-time information.

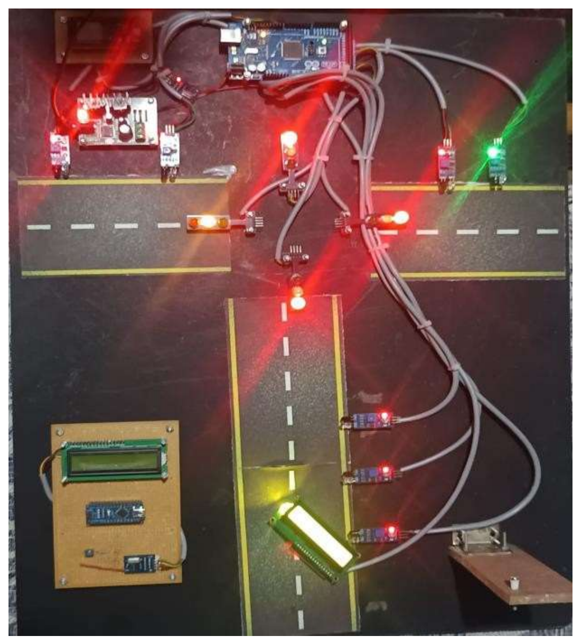

Fig. 1: Complete Hardware Setup

- **Programming the Arduino**

The Arduino IDE is used for programming the Arduino. The programming language for the Arduino is C++. The programming for the smart traffic control system is as follows:

The smart traffic control system checks the pins for the density of the traffic. The smart traffic control system checks for the presence of emergency vehicles. The smart traffic control system checks the LCD display every second. The smart traffic control system checks for the transmission of data to the cloud every 30 seconds. The programming language for the Arduino Mega 2560 is the Arduino Integrated Development Environment using the C++ programming language to control all system functions. The program initializes the pin modes for the IR sensors, LED traffic lights, RF receiver, and LCD display in the setup function. The program reads the IR sensor inputs to detect the traffic density and calls the functions to set.

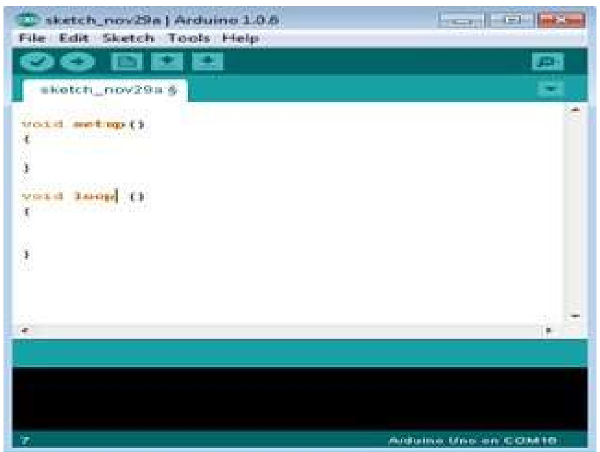

Fig. 2: Arduino IDE Interface

The Arduino code is written using the Arduino IDE with C++. The code sets all the pin modes, sets the LED pins to output mode, and sets the sensor pins to input mode. The code also continuously checks the three IR sensor inputs to determine the level of traffic density and adjusts the green light period to 5 seconds, 10 seconds, or 15 seconds accordingly.

- **Connecting ESP8266 To ThingSpeak**

The ESP8266 module connects to a wireless network and transmits the data to the API server of the ThingSpeak application. The smart traffic control system sends HTTP GET requests to the API server of the application to transmit the data such as signal counts, emergency vehicle counts, and traffic density levels. The data is transmitted every 30 seconds to enable near real-time monitoring of the traffic control system.

The ESP8266 WiFi module communicates with the local wireless network by using AT commands and makes a TCP connection to the ThingSpeak API server. Data transmission occurs by making HTTP GET requests with the write API key and data values at 30-second intervals.

Fig. 3: ESP8266 AT Commands Testing

The ESP8266 WiFi module connects to the local wireless network through the AT commands and makes TCP connections to the ThingSpeak API server. The data is sent through the HTTP GET request with the write API key and the field values. There are six fields defined in the ThingSpeak channel. The fields include active signal counts, emergency vehicle detection, congested areas, and low, medium, and high-density area counts.

- **Developing the Web Dashboard**

The web dashboard is developed using web technologies to develop a comprehensive web-based solution for the smart traffic control system.

➢ **Frontend**

HTML5, CSS3, and JavaScript along with Chart.js for graphing and Leaflet.js for mapping are used to develop the frontend of the web-based solution for the smart traffic control system. The frontend is designed in such a way that the layouts of the web pages adjust smoothly to the screen size. This way, the web-based solution can be accessed through various devices. Real-time updates of the web pages are achieved through AJAX calls to the backend to fetch new data every 30 seconds.

➢ **Backend**

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

We have developed the backend of the web dashboard for the smart traffic control system by using the Node.js platform along with the Express framework, MongoDB Atlas for the database, and RESTful API. In the backend, the RESTful API is developed with appropriate error handling and validation.

➢ **Security**

We have implemented the CAPTCHA verification OTP for password hashing in the signup process by using bcrypt and HTTP cookies to secure the web dashboard of the smart traffic control system. In the CAPTCHA system, random alphanumeric strings of six characters with mixed case are generated to prevent the bots from accessing the web application.

➢ **Login Page**

The login page ensures secure authentication through email and password fields, along with a six-character alphanumeric CAPTCHA verification to restrict bot sessions. Role-based access control is implemented by categorizing users as admin, police, and operator. The forgot password feature is also implemented to allow users to change their passwords through email-based OTP verification. Secure session management is ensured through HTTP cookies.

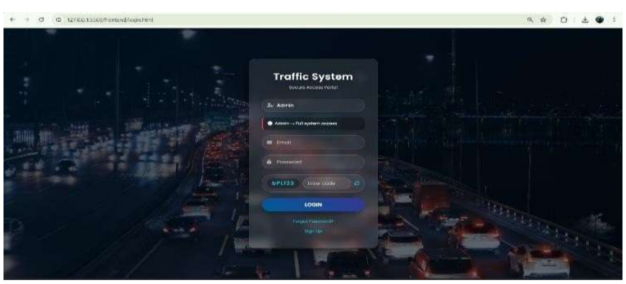

Fig. 5: Login Page

➢ **Signup Page**

The signup page asks users to provide their details such as first name, last name, mobile number, email, and password with password strength checking. The project uses double OTP verification by asking users to verify their mobile numbers and email addresses. The project also includes a confirm password field to avoid typing errors. After successful verification of user details, it securely stores user data in MongoDB Atlas with bcrypt password hashing.

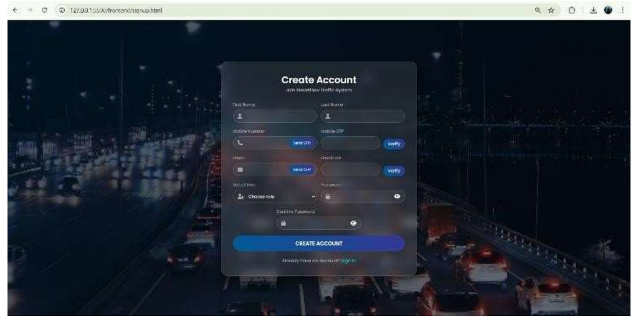

Fig. 6: Signup Page

- **Dashboard Pages**

➢ **Dashboard Page**

The dashboard page displays information on vehicles, active signals, emergency vehicles, and congested areas. It also shows information on traffic status, gauge meter, and interactive charts for the traffic control system.

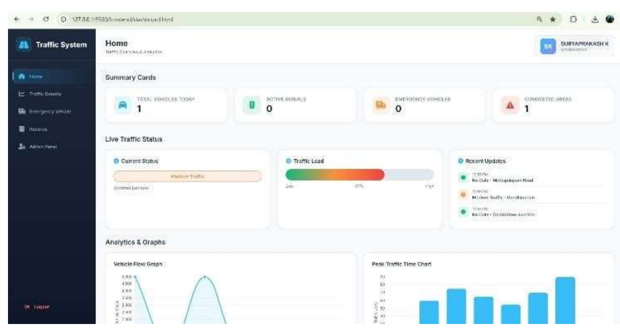

Fig. 7: Main Dashboard

➢ **Traffic Density Page**

The page displays area selection dropdowns for ten locations in Coimbatore with density status cards. The page also displays a real-time vehicle count chart and traffic density history charts for the traffic control system.

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

Fig. 8: Traffic Density Page

➢ Emergency Vehicle Page

The page is designed to display the detection cards of the ambulance, fire truck, and police vehicle with a map. In addition, the page also includes the records table for the passage of the emergency vehicle for the traffic control system.

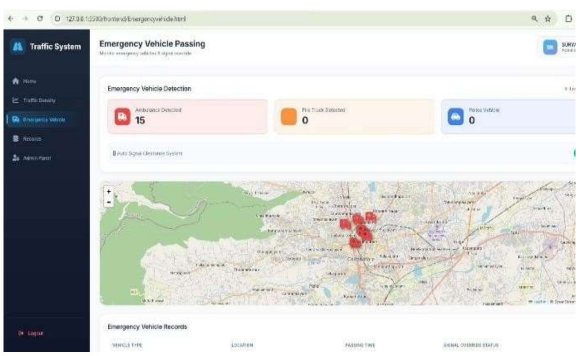

Fig. 9: Emergency Vehicle Page

➢ Records Page

The page is designed to include the option to filter the records by date range, area, and vehicle type. In addition, the page also includes the option to export the records. Furthermore, the page includes the option to navigate through the records of the traffic control system.

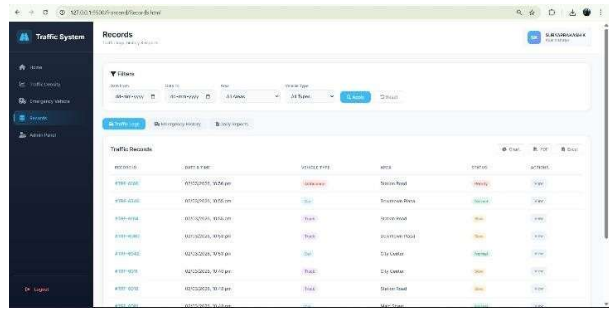

Fig. 10: Records Page

➢ Admin Panel

The admin panel is restricted to admin users and requires OTP verification. In addition, the admin panel is designed to display the user statistics, login graph, and the table of registered users. Furthermore, the admin panel includes the option to view the activity logs and block or unblock the accounts for the traffic control system.

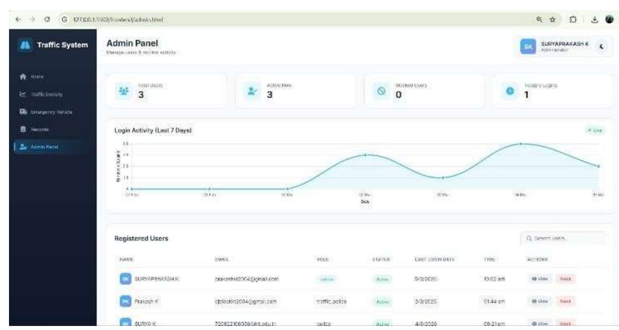

Fig. 11: Admin Panel

5. RESULTS AND DISCUSSION

The proposed smart traffic control system is implemented and tested under various conditions to assess the efficiency of the system in terms of accuracy of traffic density detection, efficiency of signal timing, and data transmission through the cloud platform. The tests were performed at a four-way intersection setup as described in the previous section of the hardware implementation of the proposed system.

- **Results of Traffic Density Detection**

It is observed that the proposed system is highly accurate in detecting the traffic density levels through the use of three IR sensors. For the experiment, a total of 150 tests were conducted for detecting three levels of traffic density, and each test was performed 50 times for each level of traffic density. The accuracy of the proposed system is presented in Table 1. The response time of the system is within 500 milliseconds to react to the change in traffic conditions.

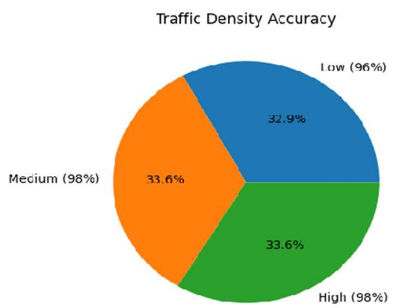

Fig. 1: Traffic Density Distribution Pie Chart

Three IR sensors were placed at a distance of 5, 10, and 15 meters from the stop line, and they were able to detect the three levels of traffic densities with high precision. The IR sensors were activated one by one, depending on the distance, when the vehicles were stationary at the intersection. The microcontroller was then able to accurately identify the level of congestion. In total, 150 tests were carried out, with 50 tests for each level of traffic density. The overall accuracy of the system was 97.3%, with the low-density level having an accuracy of 96%, medium-density level having an accuracy of 98%, and the high-density level having an accuracy of 98%. The time taken by the system to respond to the change in the signal after detecting the vehicles was less than 500 milliseconds. Such a prompt response is very important in the case of sudden surges in the volume of vehicles, which the conventional system cannot handle.

- **Signal Timing Results**

The length of the green signal was adjusted depending on the levels of traffic densities. The system was able to reduce the average waiting time.

Density Level	Green Signal Duration	Waiting Time Reduction
Low Density	5 seconds	83%
Medium Density	10 seconds	67%
High Density	15 seconds	33%

Density Level	Green Signal Duration	Waiting Time Reduction
Overall	10 seconds	58%

Table 1: Dynamic Green Signal Duration

The conventional fixed time signals had a cycle time of 30 seconds irrespective of traffic conditions. The proposed system reduced overall average waiting time from 60 seconds to 25 seconds, resulting in a 58% improvement.

Fig. 2: Waiting Time Reduction Pie Chart

In medium density, traffic was allocated 10 seconds of green time, sufficient for smooth traffic flow. High density traffic was allocated 15 seconds of green time, sufficient for smooth traffic flow even on congested lanes.

The overall average waiting time was reduced from 60 seconds, as experienced with conventional fixed time signals, to 25 seconds with the proposed system, resulting in a reduction of 58%. The waiting time was reduced from 30 seconds to 5 seconds, resulting in an 83% reduction during low traffic periods.

- **Emergency Vehicle Detection Results**

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

The RF receiver module, operating on 433 MHz, was able to detect the emergency vehicles up to 30 meters. In all, 30 tests were carried out, all of them successful.

Distance	Detection Status	Response Time
5 meters	Detected	0.6 seconds
10 meters	Detected	0.8 seconds
15 meters	Detected	0.9 seconds
20 meters	Detected	1.0 seconds
25 meters	Detected	1.2 seconds
30 meters	Detected	1.5 seconds

Table 2: RF Module Detection Performance

The average response time was found to be less than 1 second, with 100% accuracy for the detection of objects at any distance. Once the object was detected, the light immediately turned green for the approaching direction and red for all other directions.

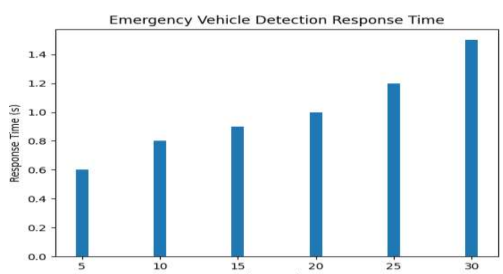

Fig. 3: Emergency Vehicle Detection Response Time vs Distance

The RF receiver module with a frequency of 433 MHz was able to detect emergency vehicles with RF transmitters at a distance of up to 30 meters. A total of 30 tests were carried out at various distances. The results showed that successful detection was achieved in all cases, which equates to 100% accuracy. The range of time taken to respond was between 0.6 seconds at a distance of 5 meters and 1.5 seconds at a distance of 30 meters. The average time taken was less than one second. Once the emergency vehicle was detected, the system immediately overrode the normal sequence of signals. The green light was turned on for the approaching traffic, and all other directions showed red.

- **LCD Display Output**

The 16x2 LCD display showed the output of the traffic information in real-time under all conditions. The LCD displayed various messages depending on the situation.

The 16x2 LCD display was used for displaying real-time information related to traffic conditions for all operating conditions with a response time of less than 100 milliseconds. Under normal conditions, the LCD display showed the active direction along with a countdown timer for the drivers to anticipate the signal change.

When low or medium density was detected, the LCD displayed "Normal Traffic" or "Medium Traffic" along with the remaining time. When high density was detected, the LCD displayed "Heavy Traffic TAKE DIVERSION" for drivers to take alternative routes for easy traffic flow. When an emergency vehicle was detected, the LCD immediately displayed "Ambulance Detected GO" for the drivers waiting to give way for the ambulance.

Condition	LCD Display Message
Normal Operation	"NORTH-SOUTH Time: 5"
Low Density	"Normal Traffic" with timer
Medium Density	"Medium Traffic" with timer
High Density	"Heavy Traffic TAKE DIVERSION"
Emergency Mode	**"Ambulance Detected GO"**

Table 3: LCD Display Messages

The LCD display response time was below 100 milliseconds for all updates. Therefore, it was able to display clear and timely information to the driver.

- **ThingSpeak Cloud Results**

The upload of data to the ThingSpeak channel with six data fields was successful. Data upload success rate was 100%. The data from the sensors was able to upload to the ThingSpeak cloud platform via the ESP8266 WiFi module. It was able to achieve a 100% success rate during the testing period. There were six data fields created on the ThingSpeak channel. They included the number of active signals, emergency vehicles detected, congested areas count, and low, medium, and high-density areas. Data was sent at intervals of 30 seconds. It was able to achieve near real-time data monitoring.

Field	Data Stored	Update Interval
Field 1	Active signals count	30 seconds
Field 2	Emergency vehicles detected	Real-time
Field 3	Congested areas count	30 seconds
Field 4	Low density areas	30 seconds
Field 5	Medium density areas	30 seconds
Field 6	High density areas	30 seconds

Table 4: ThingSpeak Data Fields

Time-series graphs are automatically created by ThingSpeak. The sensor data was successfully uploaded to the ThingSpeak cloud using the ESP8266 WiFi module with a success rate of 100% during the testing period. Six data fields were configured to store active signals count, emergency vehicles detected, congested areas, low, medium, and high-density area counts. These values are uploaded at an interval of 30 seconds.

The ThingSpeak platform automatically generates time-series graphs for each data field, thereby allowing real-time visualization and historical trend analysis of traffic conditions from anywhere.

The sensor data was successfully uploaded to the cloud using ThingSpeak cloud and ESP8266, achieving a 100% success rate. Six fields were used for storing active signals, emergency vehicles, and density counts, and the data was updated every 30 seconds. The platform generated time-series graphs for visualization. Sensor data has been successfully uploaded to the ThingSpeak cloud using ESP8266 technology with a 100% success rate.

Six fields were created to store active signals, emergency vehicles, and density counts at a 30-second interval. Time-series graphs for live visualizations were automatically created by the system. Mechanisms for auto-reconnecting and retrying data upload were also implemented to handle any possible disruptions in the network. Live data can be accessed by authorized personnel at any location through a web interface or a mobile app.

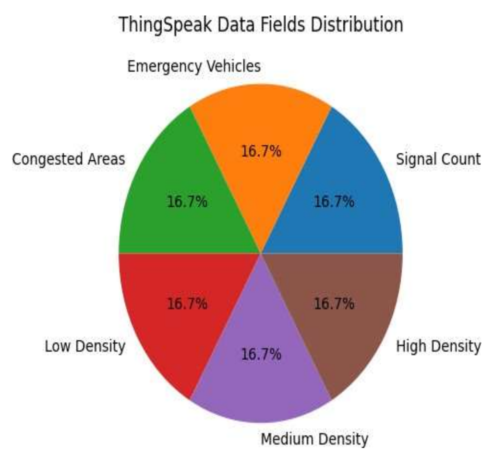

Fig. 4: ThingSpeak Cloud Output

- **Web Dashboard Results**

The web dashboard showed real-time traffic data across multiple pages. All pages were loaded within 1-2 seconds, and the data were correctly displayed.

The web dashboard, built using Node.js, MongoDB, Chart.js, and Leaflet.js, successfully showed the real- time traffic data across multiple pages. The pages were loaded within 1-2 seconds. The login page implemented email, password, and CAPTCHA (six alphanumeric characters) verification, successfully preventing bot attacks.

The signup page implemented two-factor verification using one-time passwords for mobile and email. The main dashboard showed four cards with total vehicles today,

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

active signals, emergency vehicles, and congested areas, along with the real-time traffic status and charts.

The records page showed data filtering by date, area, and vehicle type, along with export options for PDF, Excel, and chart formats.

- **Overall System Performance**

The proposed system has demonstrated outstanding performance in all the essential parameters. The accuracy of the proposed system for the detection of the density of the traffic flow is 97.3%, and the detection of the emergency vehicle is 100% up to 30 meters with an average response time of less than 1 second.

The proposed system for the dynamic signal timing reduced the waiting time of the vehicle by 58% compared to the existing system. The proposed system for the upload of the data to the cloud maintained a 100% success rate with an interval of 30 seconds, and the web dashboard loaded in 1-2 seconds.

Parameter	Result
DensityDetection Accuracy	97.3%
EmergencyVehicle Detection Rate	100% (up to 30m)
Average Response Time	< 1 second
Waiting Time Reduction	58%
Data Upload Success Rate	100%
Dashboard Load Time	1-2 seconds
LCD Update Time	< 100 ms

Table 5: Overall System Performance Summary

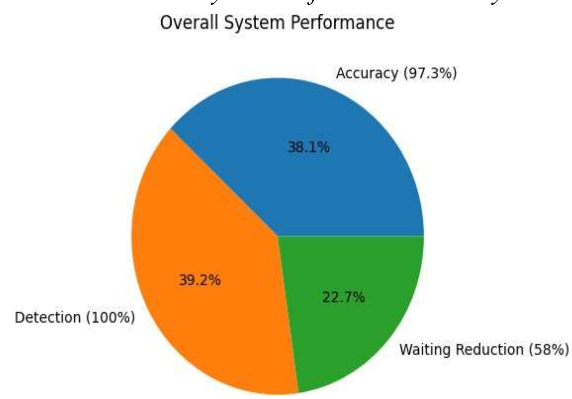

Fig. 5: Overall Sytem Performance

The proposed system has shown outstanding results in all parameters. The accuracy of traffic density detection was found to be 97.3%, thereby classifying the level of congestion accurately. The detection of emergency vehicles was found to be 100% within a distance of 30 meters, and the average response time was found to be less than a second. The dynamic signal control resulted in a reduction of average vehicle wait times by 58%. Low traffic conditions resulted in a reduction of 83%. Uploading of data to the cloud was found to have a 100% success rate with a 30-second interval.

6. CONCLUSION

The smart traffic light system with ambulance detection is a new concept for handling traffic congestion in urban cities while giving priority to ambulances. The IR sensors are used for traffic density detection and are continuously monitoring the traffic flow for dynamic signal control. The IR sensors are used for dynamic signal control; the signal is controlled for 5 seconds for low density, 10 seconds for medium density, and 15 seconds for high density traffic. The RF-based ambulance detection instantly identifies approaching emergency vehicles and provides a green corridor for faster and safer passage.

The system was successful in meeting all its objectives of monitoring traffic density in real-time, adjusting signal timings dynamically, giving priority to emergency vehicles, storing data in the cloud, and providing comprehensive visualization through the web. This fully automated and cost-effective solution can be implemented in modern cities. The smart traffic control system successfully proves the implementation of an IoT-based solution to address the issue of urban traffic congestion and prioritize emergency vehicles. The system utilizes three IR sensors to detect the density of the traffic in real time and dynamic signal timing of 5, 10, and 15 seconds based on low, medium, and high-density traffic, respectively. The RF module, operating at 433 MHz, detects emergency vehicles and provides an immediate green corridor with a response time of less than 1 second. The system achieved 97.3% detection accuracy and reduced the average waiting time of the vehicles by 58%.

7. FUTURE SCOPE

The proposed system can also be improved by incorporating artificial intelligence and machine learning to analyze the patterns of the previous day's traffic and make predictions for the future. This way, the system can be more proactive in signal control rather than just reacting to the conditions. The computer vision feature of the system can include cameras to detect the type of vehicle, such as car, bus, or motorcycle. This feature can also be used for the priority of public transport, accident detection, and safety of pedestrians. The GPS feature can be added to track the location of emergency vehicles and also to predict the time of arrival, thus enabling the clearing of the path through the hospital and fire station systems. A mobile app can also be developed to send alerts to the users of the road network and also to request priority passage for emergency services. Expanding to multi-intersection coordination will ensure the formation of green wave corridors for smooth traffic flow across the city. Vehicle-to-infrastructure communication will ensure direct interaction of vehicles and signals for optimal vehicle timings. Integration with smart city infrastructure such as public transport, parking, and environmental sensors will ensure a holistic solution for urban mobility. This system may also be enhanced with the incorporation of artificial intelligence in the prediction of the flow of traffic, GPS technology in emergency vehicle tracking, and computer vision in the identification of vehicles and accident detection.

8. REFERENCES

[1] P. Bairi, S. Swain, A. Bandyopadhyay, K. Aurangzeb, M. Alhussein, and S. Mallik, "Intelligent VANET-based traffic signal control system for emergency vehicle prioritization and improved traffic management," Egyptian Informatics Journal, vol. 30, pp. 100700, Jun. 2025.

[2] M. Hanif et al., "Ambulance detection and priority passage at urban intersections using transfer learning and explainable AI," Int. J. Adv. Comput. Sci. Appl., vol. 16, no. 10, pp. 285- 292, Oct. 2025.

[3] M. Islam, "Analysis of AI-enabled adaptive traffic control systems for urban mobility optimization through intelligent road network management," Review of Applied Science and Technology, vol. 4, no. 2, pp. 207-232, 2025.

[4] Md S. Arefin, Md I. S. Mahin, and F. A. Mily, "Real-time rapid accident detection for optimizing road safety in Bangladesh," Heliyon, vol. 11, Art. no. e42432, Jan. 2025.

[5] M. Q. Kheder and A. A. Mohammed, "Real-time traffic monitoring system using IoT-aided robotics and deep learning techniques," Kuwait Journal of Science, vol. 51, 2024.

[6] A. A. Alhaj, N. I. Zanoon, A. Alrabea, H. I. Alnatsheh, O. Jawabreh, M. Abu-Faraj, and B. J. A. Ali, "Improving the smart cities traffic management systems using VANETs and IoT features," Journal of Statistics Applications & Probability, vol. 12, no. 2, pp. 405-414, 2023.

[7] M. Gupta, H. Miglani, P. Deo, and A. Barhate, "Real-time traffic control and monitoring using computer vision and IoT- based intelligent traffic systems," e-Prime - Advances in Electrical Engineering, Electronics and Energy, vol. 5, pp. 100211, Jul. 2023.

[8] D. Goel, S. Chaudhury, and H. Ghosh, "An IoT approach for context-aware smart traffic management using ontology," Procedia Computer Science, vol. 198, Aug. 2022.

[9] A. A. Ouallane, A. Bahnasse, A. Bakali, and M. Talea, "Overview of road traffic management solutions based on IoT and AI," Procedia Computer Science, vol. 198, pp. 518-523, 2022.

[10] L. Zhong and Y. Chen, "A novel real-time traffic signal control strategy for emergency vehicles in intelligent transportation systems," IEEE Access, vol. 10, pp. 19481- 19495, Feb. 2022.

[11] V. K. Dileep, P. S. Athira, P. Jayakumar, and S. Praveena, "A Comprehensive Review of Smart Emergency Vehicle Detection and Response Systems," in *2025 3rd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, Coimbatore, India, Apr. 2025, pp. 1–6. doi: 10.1109/ICAECA63854.2025.11012205.

[12] A. Jangir, D. Jain, K. Nayak, and V. R. S. Mahala, "Density Based Traffic Control System and Emergency Vehicle Detection Using Arduino," *International Journal of AdvancedEngineering, Management and Science*, vol. 11, no. 2, pp. 134–

141, 2025. doi: 10.22161/ijaems.112.12.

[13] K. Priya, K. Priyadharshini, R. S. Krishnan, J. R. F. Raj, I. J. Settu, and A. Srinivasan, "Advancing Urban Traffic Control with IoT and Deep Learning: A YOLOv8 and LSTM-Based Adaptive Signal System," in *2025 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, May 2025, pp. 1–7. doi: 10.1109/ICICCS65191.2025.10984735.

[14] S. T, S. V. S, and A. S. Savanth, "Integrated Traffic Control System for Emergency Vehicles," in *2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, Bengaluru, India, Jan. 2024, pp. 1–5. doi: 10.1109/IITCEE59897.2024.10467245.

[15] P. S. S. N. Rao, K. S. Reddy, and M. S. Kumar, "Dynamic Traffic Signal Optimization Congestion Management and Emergency Vehicle Prioritization," in *2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, Bengaluru, India, Feb. 2025, pp. 960–963. doi: 10.1109/IDCIOT64235.2025.10915151.

[16] V. Mishra, S. Shukla, S. K. Singh, S. Srivastava, and A. Garg, "Real-Time Adaptive Traffic Control System with Emergency Vehicle Detection Based on Computer Vision and Sound Frequency Monitoring," in *2025 2nd International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE)*, Chennai, India, May 2025, pp. 1–6. doi: 10.1109/RMKMATE62263.2025.11042468.

[17] A. R. Zulkifli, K. Ali, and Z. Abd Rahman, "IoT-Enabled Traffic Management System for Emergency Vehicle Prioritization," in *Proceedings of the International Conference on Smart Computing and Communication*, Singapore, 2022, pp. 245–253. doi: 10.1007/978-981-19-0825-5_23.

[18] M. T. Nafis and M. H. Khan, "IoT Enabled Traffic Control Model Using Raspberry Pi," *International Journal of Advanced Research in Computer Science*, vol. 9, no. 3, pp. 157–160, Jun. 2018. doi: 10.26483/ijarcs. v9i3.5972.

[19] E. Basil and S. D. Sawant, "Intelligent Transportation System Using RFID to Reduce Congestion, Ambulance Priority and Stolen Vehicle Tracking," in *2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon)*, Bengaluru, India, Aug. 2017, pp. 1–5. doi: 10.1109/SmartTechCon.2017.9036164.

[20] J. Hosur, R. Rashmi, and M. Dakshayini, "IoT Based Smart Traffic Control System for Emergency Vehicle Clearance," in *2019 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, Mangalore, India, Aug. 2019, pp. 1–6. doi: 10.1109/DISCOVER.2019.89876