IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 1, March 2024, pp. 899~908
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i1.pp899-908

Automated signal pre-emption system for emergency vehicles using internet of things

Pushpa Sothenahalli Krishnaraju, Manjunath Thimmasandra Narayanappa, Sheela Kathavate, Rudresh Shirwaikar, Anupchandra Rao Chandrakanth Rao, Jayachnadra Rohith
Department of Information Science and Engineering, BMS Institute of Technology and Management, Bengaluru, India

Article Info

Article history:

Received Sep 15, 2021
Revised Aug 7, 2023
Accepted Nov 30, 2023

Keywords:

Internet of Things
Machine learning
Traffic management congestion

ABSTRACT

Vehicle administration systems are one of the major highlights especially in urban areas. One important critical component that requires attention are signal preemption systems. Every single work on traffic congestion identification either requires prior learning or long time to distinguish and perceive the closeness of congestion. FutureSight performs predictive analysis and control of traffic signals through the application of machine learning to aide ambulances in such a way that, a signal turns green beforehand so as to ensure an obstacle free path to the ambulance from source to destination based on various parameters such as traffic density, congestion length, previous wait times, arrival time thereby eliminating the need for human intervention. The method allows flexible interface to the driver to enter the hospital details to reach the destination with in time. The app then plans out the fastest route from the pickup spot to the selected hospital and sends this route to the system. The system then predict the amount of time that is required by the signal to remain green so as to clear all traffic at that specific junction before the ambulance arrives at that location.

Corresponding Author:

Pushpa Sothenahalli Krishnaraju
Department of Information Science and Engineering, BMS Institute of Technology and Management
Yelahanka, Bengaluru, Karnataka 560064, India
Email: pushpask@bmsit.in

1. INTRODUCTION

The growth of industrialization and urbanization has led to an immense increase in the population invariably leading to rise in the number of vehicles on road. The resulting traffic congestion and traffic jams are the major hurdles for emergency vehicles as these vehicles which are not able to reach their destination in time, resulting in added risk and danger to victims. A Patient may lose his life if there is delay in reaching of an ambulance to the hospital. According to surveys, if the ambulance can reach the hospitals in time without getting stuck in the traffic. 95% of the heart attack cases can be treated. At inception traffic was mainly controlled by policemen at every intersection, however as traffic increased gradually signals came into existence. However, these signals were primarily time based and did not consider the volume of vehicles on the road. Furthermore, if an emergency vehicle got caught in traffic, then it was the duty of the policeman to clear the road for the vehicle or it was left to the discretion of the drivers ahead of the emergency vehicles whether to make way of wait for the signal to turn green. Even if they decided to make way for the emergency vehicles when the signal was red it wasn’t recommended since it increased the probability of collision of vehicles.

We thus have to look for an automated system to manage traffic taking into account all these factors that remain unsolved today. Hence our aim is to aid ambulances rushing towards hospitals by providing best possible route with minimum delay. Here machine learning and cloud with internet of things (IoT) technology has been used for pre-empt traffic signals with minimum delay from the cloud data.

Quite a good amount of work has been done in this area, to address the challenges in managing the flow of vehicular sensing networks, reinforcement method has been used for sharing city information with different modules like monitoring module to know the behavior of the network and activates reputation management module which holds the historical information, which in turn informs path management and trust module to take appropriate decision [1]. Radio frequency identification tagged method [2] with central database processing system, which collects information of detecting emergency vehicles at junction points, violating traffic rules, detecting lost vehicles and providing current location information to the user in. Preemption of traffic signal was design with IoT and mobile application [3]. The metadata has been stored in webserver, proper security has been provided so that data is not misused. System which turn signal green when the ambulance is approaching the junction, system tries to create a green wave, however it needs predefined timings to control signals irrespective of a road conditions [4]. Object detection is done using image processing technique [5], here vehicles backlights are captured and their two lights are paired using distance formula to confirm the vehicles count and congestion. This method gave about 95.75% to 100% accuracy. Microcontroller [6] is used to estimate the time of Arrival and critical time until the emergency vehicles are preempted from the traffic. Bangalore having grown exponentially in the past two decades unlike western countries has no method of adding lanes for emergency. It consists of 2 units ambulance unit, sends global positioning system (GPS) data to the signal junction whereas the junction unit is responsible to receive GPS data and turn signal green. In case of multiple ambulance priority based first come first serve (FCFS) scheduling is used to decide which ambulance to let through. The main disadvantage is that they make use of hardware which is processor intensive. Information is gathered at each signal side using queue length of cars, and then applies optimization technique waiting time and service time. The signal sends messages to the traffic control room, if any traffic violation takes place to avoid accidents [7], [8]. System using radio frequency identification (RFID) tags for vehicles and the system performs very fast in changing the traffic signals based on the number of vehicles at the signal [9]. Ultronic sensors are used to detect the traffic and intimate the traffic congestion on the road and message is sent to the driver through mobile app so that if any congestion driver will be intimated with alternate route [10]. Emerging technologies like IoT cloud computing enhances the quality of the services for smart cities and improve the overall traffic management [11]. Predictive model for groundnut [12] management using multivariate regression technique to predict crop weather relation for groundnut pest trips under dynamic weather conditions. They have also considered two season data and a cumulative model has been developed and tested. In the paper [13] the real time images are used to identify the flow of traffic using masking algorithm. Arduino board controller ads visual basic programming has been used. The traffic light management using radio frequency (RF) signals and programmable integrated circuit microcontroller have been used to automatically set back to normal signal sequence. The model works with in the frequency of 434MHz [14]. The sustainable policies facilitated by the government for the welfare of the citizens. Challenges and initiatives have been addressed [15]. Supervised learning algorithm is used to classify the policies based on the option of the public. The virtual world with reference to IoT and the challenges with sensors, actuators and processors are addressed in [16]–[18]. Challenges in building smart dynamic and flexible network and network and service management in intelligent distributed system. Advancement is technology has helped the researches to come out with new methods to save the lives of the needy people in the time of emergency. Technologies like IoT, Data analytics, cloud computing [19] and play a major role in emergency vehicular system and smart cities. Paper address the way internet of things, information and communications technolog (ICT) and cloud can be leveraged in various applications towards the smart cities. Addresses innovations, challenges and application based scenario in. Two controllers are used in parallel to capture the lane for green signal and calculate appropriate signal time as per the real time traffic. This paper uses fuzzy models [20] to change traffic signals, which in-turn provides clearance to the emergency vehicle. Traffic light are managed using peripheral interface controller. Emergency vehicles itself can trigger the traffic lights to change their signal and provide clearance path for the vehicle. Later traffic signals will be back to normal using radio frequency. The limitation here is that it works within the range of 55 meters [21]. This avoids traffic and also includes smart control system for the clearance of emergency vehicles. This paper aims at addressing accidents where emergency vehicles and care takers to be notified in time and also location to be shared with ambulance and care takers with the help of GPS [22]–[25] and the global system for mobile communication (GSM).

2. RESEARCH METHOD

FutureSight involves the culmination of many emerging technologies such as internet of things, machine learning, and data analytics. hence it makes its presence felt in multiple fields especially in Healthcare: Increasing traffic in metropolitan cities and towns has caused havoc in the movement of emergency vehicles. These congestions severely limit the probability of the patient being admitted to the hospital on time. In such conditions these systems can ease transportation time to a greater extent and increase chances of patient's betterment. Our aim in this paper is to help ambulance with emergemcy patients to the hospital in the fastest possible way with less delay using machine learning to analyse and trigger the signal automatically and cloud with IoT setup to get realtime data. The pre-emption time can take on real values and these values are given to the Regressor algorithm. The system predicts ahead of time when to turn the traffic signal green whereas existing systems wait for the ambulance to arrive at that junction and then proceeds to change the signal. This way any factors delaying the vehicle previously will be accordingly optimized and minimized in the future iterations. It also makes use of the benefits of IoT to send data, communicate and manipulate traffic signals in real time. FutureSight would initially consist of 4 prominent modules.

a) Mobile application: Used by ambulance drivers to chalk out a route from incident location to destination. This application would consist of an option to select the preferred hospital and a field to enter the criticality of the patient's condition. Once provided with this information it finds the most optimal path from source to destination.
b) Data collector: Module that acquires traffic data such as density, congestion level, congestion length, and GPS coordinates. Initial plan to acquire this data is through the use of application programming interfaces (Google, Waze, 511.org and OpenStreetMap). The reason we are using software to gather data is because utilization of hardware makes the overall application very intensive.
c) Prediction module: Algorithm that analyses data and builds a training model to predict future values. The algorithm can either be based on random forest regressor learning model. The primary reason we are applying machine learning is because the system can take in data as it progresses and can use this data to predict better results thereby increasing overall accuracy.
d) Traffic signal controller: IoT sub-system that controls signals in accordance to data received from the prediction algorithm. These systems are planned to be installed in each traffic signal and receive data.

Figure 1 indicates a very abstract representation of the system. First driver chooses the hospital through the app. The app calculates the best path using routing algorithm considering congestion data into consideration. Once the most optimal route is confirmed, the allied routing information will be sent to the cloud server through the mobile application. Then the cloud compute engine initiates an iterative and recursive prediction process which occurs till the ambulance reaches the destination. Every signal position at intersections or otherwise are recognized on the route map using a concept of 'Plus Codes' developed by Google. The hardware sensors to obtain required data will be kept as a fall-back option as it is desired to keep the system hassle free from processor intensive tasks. Then the prediction model starts to compute the pre-emption time required to keep the intersection traffic free while the EV arrives at desired spot. At the same time, this prediction model can be designed manually applying Python or by using 'Google's machine learning framework tensor flow'. Now, random forest regressor model is applied in prediction module as referred in Figure 2. The same data will be sent to the signal monitoring point that is implemented using NodeMCU, an affordable cost but reliable on IoT microcontroller chip. Then the signal resumes on a normal operation as the ambulance passes the junction. In case of manifold ambulances at contiguous intersections the ambulance with a perilous patient will be given upper urgency.

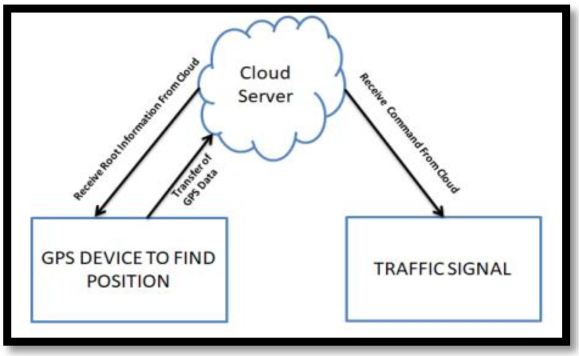


Figure 1. Abstract system implementation

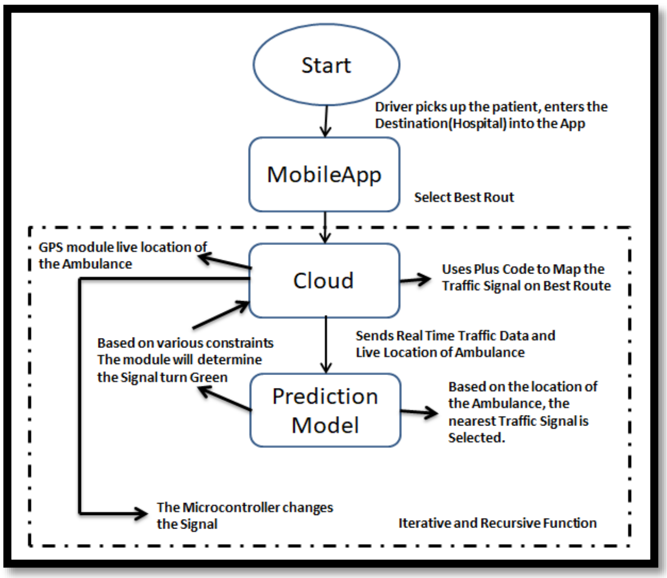


Figure 2. System flow diagram

2.1. The android app module

The Android application shown in Figure 3 consists of a search field in which the hospital to go to is entered. The search field supports autocomplete using Google autocomplete application programming interface (API). The app also consists of a slider that takes discrete values from 1 to 10 that is set by the driver to indicate the criticality of the patient. The criticality parameter is used when calculating the pre-emption time and when multiple ambulances are approaching the same junction. The ambulance carrying the patient with the higher criticality is given priority. As shown in Figure 3, the app uses a custom tracking service that runs in the background. The application broadcast's location information every second in real-time to Google's firebase real time database.

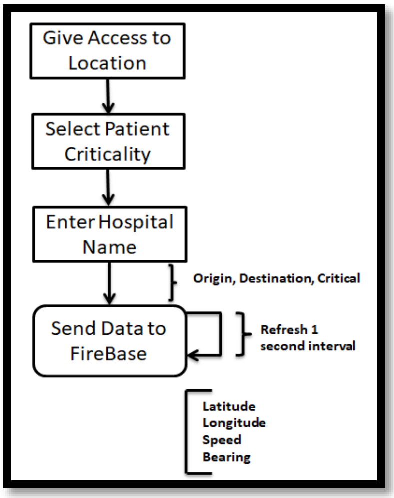


Figure 3. Android module architecture

The information that we use from the data includes bearing, speed, latitude, longitude, time, slider and search. Our Firebase listener script connects to the real time database and continuously listens for new data as shown in Table 1. As soon as the driver starts the trip the listener daemon retrieves this information from firebase and stores this in Mongo for further use.

Table 1. The app broadcasts the following information

Parameter	Value
Accuracy	23.749000549316406
Altitude	847.9000244140625
Bearing	0
Bearing Accuracy Degrees	0
Complete	True
ElapseRealtimeNanos	999425672000000
FromMockProvider	False
Latitude	13.1364087
Longitude	77.570853
Provider	fused
Speed	0
SpeedAccuracyMeterPerSecond	0
time	1557295757289
VerticalAccuracyMaters	2
Location	"Vikram hospital"

2.2. The data collector module

The data collector module in Figure 4 comprises of the following sub-process or programs that execute in sequence, route finder, decoder, comparator and gatherer. When the app sends the destination, origin and criticality information to firebase the program that continuously listens pulls this data from the database, parses it and stores in our own database for further processing. A program that is listening to this database collection for new data retrieves this incident information and calls the directions module to calculate the most optimal route.

The directions module is a Python wrapper built over the Google directions API. Once we have supplied our API key and the requested parameters to the module namely origin, destination and transit mode (driving by default). The gmaps direction function is called on these parameters to return data in javascript object notation (JSON) format. This JSON data contains route information such as route legs, turn by turn instructions, detailed origin and source information. Now each of these route legs are represented as polylines. Polylines are encoded data that can be represented on a map. These polylines are accumulated and formatted which are then sent to the decoding sub-module.

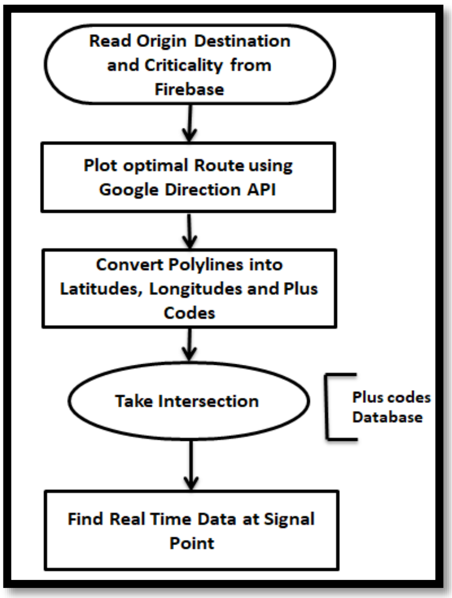


Figure 4. Data collector module architecture

2.2.1. Calculating the optimal route

When the app sends the destination, origin and criticality information to firebase the program that continuously listens pulls this data from the database, parses it and stores in our own database for further processing. Once we have supplied our API key and the requested parameters to the module namely origin, destination and transit mode. The gmaps.direction() function is called on these parameters to return data in JSON format. This JSON data contains route information such as route legs, turn by turn instructions, detailed origin and source of information. Now each of these route legs is represented as polylines. Polylines are encoded data that can be represented on a map. These polylines are accumulated and formatted which are then sent to the decoding sub-module.

2.2.2. Decoding and parsing polylines

Polyline encoding is a lossy compression algorithm that allows us to store a series of coordinates as a single string. Point coordinates are encoded using signed values. The encoding process converts a binary value into a series of character codes for american standard code for information interchange (ASCII) characters using the familiar base 64 encoding scheme: to ensure proper display of these characters, encoded values are summed with 63 (the ASCII character '?') before converting them into ASCII. The algorithm that is used in this application is the decoder algorithm which executes the steps in reverse order. Each of the polylines is decoded to a list of latitudes and longitudes.

2.3. The time predictor module

2.3.1. Retrieving the data from the database

The signal to be processed is retrieved from the signals database using plus codes as primary key. Now in practicality there would be different signals controlling different directions so within each document a separate dataset for each direction is maintained. Hence, we require the direction towards which the ambulance is travelling so as to retrieve the appropriate dataset as mentioned in data collector module Figure 4.

2.3.2. Building the model

Once the data for the requested signal and direction are retrieved, the data is formatted and sent to the scikit-learn module so that a reliable prediction model can be constructed. The underlying supervised machine learning algorithm that is used is random forests. Random forests perform a majority vote across all the models to find one with the highest accuracy. Since the pre-emption time can take on real values the algorithm is a regressor.

2.4. The hardware module

Each of the signal points act as clients or subscribers who are subscribed to a particular topic. The broker is an intermediary that receives messages from the server and broadcasts that message to all the subscribers of a topic. The broker used in this case is the hive message queuing telemetry transport (MQTT) broker. Communication between the server and clients is secured using transport layer security (TLS) encryption to the man-in-the-middle MITM attacks and protect the integrity of the data. The server in this case acts as a publisher which sends commands to the broker as shown in data collector module Figure 4.

2.4.1. The Signal to be controlled

Hardware controller shown in data collector module Figure 4. Communication between the server and signals takes place wirelessly over the internet and makes use of the message queue telemetry transport (MQTT) protocol for communication. The reason that MQTT was chosen is because there could be situations where multiple signals control the same direction, this requires a broadcast architecture which can be achieved using MQTT. The Broker is an intermediary that receives messages from the server and broadcasts that message to all the subscribers of a topic. The Broker used in this case is the Hive MQTT Broker. Communication between the server and clients is secured using TLS encryption to the MITM attacks and protect the integrity of the data.

2.4.2. Sending commands to the signal

Once the signal to be changed is identified, at the junction, and the signal pre-emption time is sent to a custom python script. This python script connect to the hive broker using the Paho MQTT client Library. Objects are created on the fly depending on the number of signals or messages are published by calling publish() on these objects. The scripts sends binary data and is interpreted as follows; data bit 1 turns a signal green. Data bit 0 turns a signal red. Once the command is sent to the signal the script sleeps for a particular period of time (specified by the pre-emption time). Once it starts up again it sends the data bit 0 to change back the state of the Signal. The broker listens on port 1883 for messages sent by the server. The Signal's LED's are controlled by the digital pins on the IoT chip. Since the signal operates at a high current value, Electromagnetic relays are used for switch control. The signal resumes its normal operation once the Ambulance crosses it.

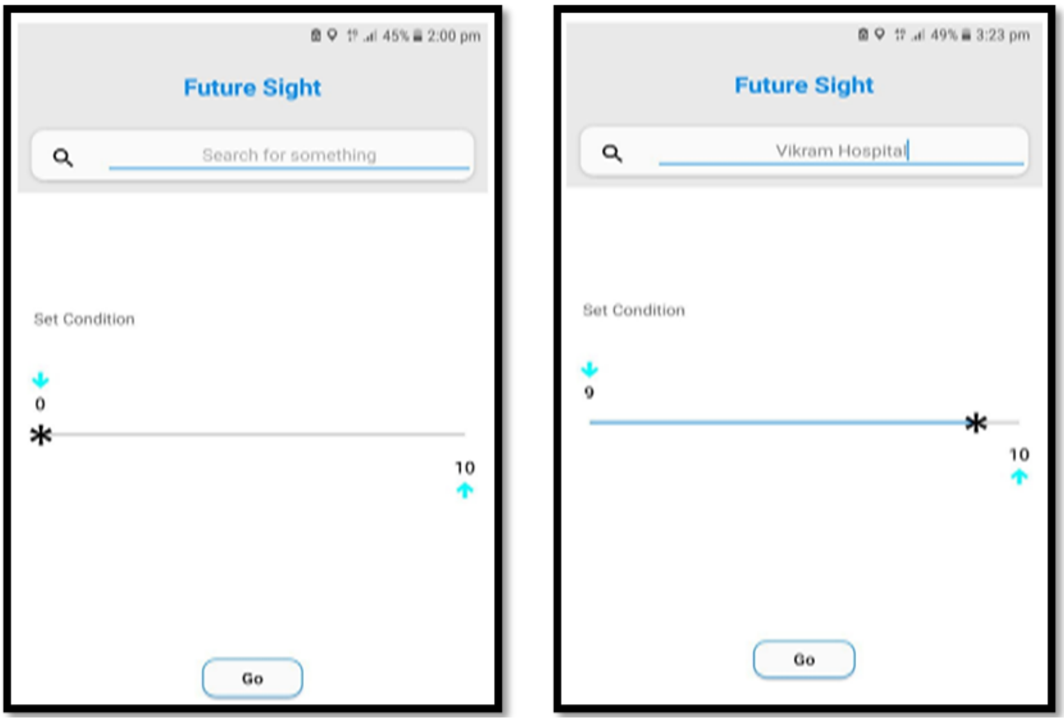
2.4.3. The NodeMCU controller

As mentioned before each signal is monitored and controlled by NodeMCU. The NodeMCU connects to the Hive Broker through a Wireless Access Point which is secured with a password. Each NodeMCU contains a unique topic though which the Server sends commands to change state. The chip continuously listens for messages on the subscribed topic once its successfully connected to the Broker. In essence, one topic receives 1 (turns signal green) as data, all other topics receive 0 (turns signal red) and vice versa.

3. RESULTS AND DISCUSSION

The technology used to pre-empt signals require hardware like RFID, sirens, and cameras, that are either mounted onto the vehicle or the traffic signals. They have a more manual approach to control the traffic lights to pre-empt the emergency vehicles and most of these approaches are not that reliable and practical, also there is a use of a predetermined route to the hospital without considering the external factors like traffic congestion, which drastically influences the transit time of the emergency vehicle.

Communication using message queue telemetry transport (MQTT) protocol for communication as shown in Figure 5(a) and (b). Signal pre-emption time. This python script connects to the hive broker using the PAHO MQTT client library. Objects are created on the fly depending on the number of signals are messages are published by calling publish () on these objects. The Figure 6 as shown about hardware module initialization and server-side signals. The hardware module initialization as shown in Figure 6(a). Data bit 1 turns a Signal Green as shown in Figure 6(b), and Data bit 0 turns a signal Red as in Figure 6(c). Once the command is sent to the signal the script sleeps for a particular period of time (specified by the pre-emption time). Once it starts up again it sends the data bit 0 to change back the state of the Signal. App provides flexible user interface to the ambulance driver. Accurate and reliable traffic data at real time and perform accurate prediction overtime. After testing the system for a few routes, the travel time was reduced by approximately, 8-10 minutes. The comparison graph shows that the model prediction time for regular transit and with the optimizer. Using bagging as the optimizer further time has been reduced as shown in Figure 7.



(a) (b)

Figure 5. Communication using message queue telemetry transport; (a) Application landing screen and (b) Completed fields (sent to server)

Figure 6. Hardware module initialization and server side signals, (a) Hardware module initialization, (b) Server sends data bit 1 to signal (commend sent to the controller is "1", the signal turns green), and (c) Server sends data bit 0 to signal (commend sent to the controller is "0", the signal turns red)

Figure 7. Comparison of graph run1 and run 2

4. CONCLUSION

As Human life is precious, one should take safety measures in all walks of life. For example, ambulances services in health section are need of the day. The human intervention is minimized due to improvement in the prediction model through Machine Learning. Which ensures cent point accuracy. With the help of GPS accuracy, locations are detected. In the present article, FutureSight uses machine learning to perform predictive analysis and control of traffic signals, assisting ambulances in such a way that, based on various parameters such as Traffic Density, Congestion Length, Previous Wait Times, and Arrival Time, a signal turns green in advance to ensure an obstacle-free path to the ambulance from source to destination, thereby obviating the need for human intervention. The approach gives the driver a flexible interface so they can still arrive at their destination on time. The app then determines the quickest route between the pickup location and the chosen hospital and submits it to the system. Hence, this system may be considered more consistent and instantaneous compared to hardware based and pictorial based detection systems. Though this system is cost effective, but it is deployed using Machine Learning, cloud, and IoT which establishes in a combination which is enormously commanding and precise. Future research may extend this work by distributed Hadoop System can replace Mango DB to accommodate huge amount of data that gets accumulated. Hadoop cluster helps in increasing processing speed and reduceing computational overhead. Further more than one emergency can be facilitated for multiple trips using concurrency control and care will be taken to authenticate the trips of the driver so that they don't misuse for their personal use. To alert pedestrians and drivers led monitors can be installed at signals. To decrease the travel time from travel to pickup point, reverse process of using the app can be facilitaed. Estimated arrival time of ambulance and patients' condition can be notified and sent to the hospitals in prior, so that necessary arrangements can be setup in hospitals. Real time data which is more sensitive can be analysed by using improved prediction methods.

REFERENCES

[1] J. Wang, C. Jiang, K. Zhang, T. Q. S. Quek, Y. Ren, and L. Hanzo, "Vehicular Sensing Networks in a Smart City: Principles, Technologies and Applications," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 122–132, 2018, doi: 10.1109/MWC.2017.1600275.

[2] S. S. K. Valhavankar Shripad N, Vibhute Aniket S, Bhagat Akshay G, "Intelligent traffic control system (emergency vehicle clearance and lost vehicle detection)," *International Journal of Innovative Research in Computer and Communication Engineerin*,

vol. 4, no. 4, 2016.
[3] N. R. Prabhanshu Attri, Fatima Rafiqui, "Traffic signal preemption (TSP) system for ordinary vehicles in case of emergency based on Internet of Things ecosystem," *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016.
[4] R. Sundar, S. Hebbar, and V. Golla, "Implementing intelligent traffic control system for congestion control, ambulance clearance, and stolen vehicle detection," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 1109–1113, 2015, doi: 10.1109/JSEN.2014.2360288.
[5] A. Nidhal, U. K. Ngah, and W. Ismail, "Real time traffic congestion detection system," in *2014 5th International Conference on Intelligent and Advanced Systems: Technological Convergence for Sustainable Future, ICIAS 2014 - Proceedings*, IEEE, 2014. doi: 10.1109/ICIAS.2014.6869538.
[6] A. S. Eltayeb, H. O. Almubarak, and T. A. Attia, "A GPS based traffic light pre-emption control system for emergency vehicles," in *Proceedings - 2013 International Conference on Computer, Electrical and Electronics Engineering: "Research Makes a Difference", ICCEEE 2013*, IEEE, 2013, pp. 724–729. doi: 10.1109/ICCEEE.2013.6634030.
[7] I. M. Hegde R, Sali R, "RFID and GPS based automatic lane clearance system for ambulance," *Int. J. Adv. Elect. Electron. Eng 2.3*, pp. 102–107, 2013.
[8] F. A. Al-Nasser and H. Rowaihy, "Simulation of dynamic traffic control system based on wireless sensor network," in *ISCI 2011 - 2011 IEEE Symposium on Computers and Informatics*, IEEE, 2011, pp. 40–45. doi: 10.1109/ISCI.2011.5958880.
[9] A. Chattaraj, S. Bansal, and A. Chandra, "An intelligent traffic control system using RFID," *IEEE Potentials*, vol. 28, no. 3, pp. 40–43, 2009, doi: 10.1109/MPOT.2009.932094.
[10] S. M. Ahir D, Bharade S, Botre P, Nagane S, "Intelligent traffic control system for smart ambulance," *IRJET*, vol. 5, pp. 355–358, 2018.
[11] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112–121, Apr. 2014, doi: 10.1109/JIOT.2013.2296516.
[12] M. Divya, T. N. Manjunath, and R. S. Hegadi, "A study on developing analytical model for groundnut pest management using data mining techniques," in *Proceedings - 2014 6th International Conference on Computational Intelligence and Communication Networks, CICN 2014*, IEEE, Nov. 2014, pp. 691–696. doi: 10.1109/CICN.2014.152.
[13] R. Vani, N. Thendral, J. C. Kavitha, and N. P. G. Bhavani, "Intelligent Traffic Control System with Priority to Emergency Vehicles," *IOP Conference Series: Materials Science and Engineering*, vol. 455, no. 1, p. 012023, Dec. 2018, doi: 10.1088/1757-899X/455/1/012023.
[14] M. A. I. N. M. Z. Hashim, A. S. Jaafar, N. A. Ali, L. Salahuddin, N. R. Mohamad, "Traffic Light Control System for Emergency Vehicles Using Radio Frequency," *IOSR Journal of Engineering*, vol. 3, no. 7, pp. 43–52, 2013, doi: 10.9790/3021-03754352.
[15] H. Iqbal, S. Paul, and K. Khan, "Sustainable governance in smart cities and use of supervised learning based opinion mining," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 1, p. 489, 2021, doi: 10.11591/ijece.v11i1.pp489-497.
[16] P. Khare and A. Khare, "Internet of Things for Smart Cities," in *Smart Cities and Smart Spaces*, IGI Global, 2019, pp. 799–815. doi: 10.4018/978-1-5225-7030-1.ch036.
[17] M. J. Kaur and P. Maheshwari, "Building smart cities applications using IoT and cloud-based architectures," in *2016 International Conference on Industrial Informatics and Computer Systems, CIICS 2016*, IEEE, 2016. doi: 10.1109/ICCSII.2016.7462433.
[18] R. Petrolo, V. Loscri, and N. Mitton, "Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 1, 2017, doi: 10.1002/ett.2931.
[19] M. Naphade, G. Banavar, C. Harrison, J. Paraszczak, and R. Morris, "Smarter cities and their innovation challenges," *Computer*, vol. 44, no. 6, pp. 32–39, 2011, doi: 10.1109/MC.2011.187.
[20] A. Agrawal and R. Paulus, "Improving traffic and emergency vehicle clearance at congested intersections using fuzzy inference engine," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 4, pp. 3176–3185, 2021, doi: 10.11591/ijece.v11i4.pp3176-3185.
[21] N. Hashim, F. Idris, A. F. Kadmin, and S. S. J. Sidek, "Automatic traffic light controller for emergency vehicle using peripheral interface controller," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 3, pp. 1788–1794, 2019, doi: 10.11591/ijece.v9i3.pp1788-1794.
[22] Y. R U, K. K.Modak, P. S.Shivshette, and S. S.Vhaval, "Smart Traffic Control System," *International Journal of Electronics and Communication Engineering*, vol. 3, no. 3, pp. 20–23, 2016, doi: 10.14445/23488549/ijece-v3i3p106.
[23] S. Sharma and S. Sebastian, "IoT based car accident detection and notification algorithm for general road accidents," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 5, pp. 4020–4026, 2019, doi: 10.11591/ijece.v9i5.pp4020-4026.
[24] P. Guo, M. Wang, H. Rong, and W. Wang, "Intelligent Traffic Control Method for Emergency Vehicles Prioritization Based on DSRC Transportation System," in *SAE Technical Papers*, in ICVS, vol. 2018- Augus. SAE International, 2018. doi: 10.4271/2018-01-1644.
[25] H. Hsiao, J. Chang, and P. Simeonov, "Preventing Emergency Vehicle Crashes: Status and Challenges of Human Factors Issues," *Human Factors*, vol. 60, no. 7, pp. 1048–1072, 2018, doi: 10.1177/0018720818786132.

BIOGRAPHIES OF AUTHORS

Dr Pushpa Sothenahalli Krishnaraju has received B.E. degree in Computer Science and Engineering in 1995. Masters in Computer Science and Engineering in 2004. Ph.D. in Computer Science and Engineering in 2017. She has overall 20 years of experience in teaching, research and administration. Her research areas include Wireless Sensor Networks, Internet of Things and Data Science. She has 11 journal articles and 17 conference proceedings published in reputed Scopus and web of science listed journals. She can be contacted at email: pushpask@bmsit.in.

Dr Manjunath Thimmasandra Narayanappa has received B E. degree in Computer Science and Engineering in 2001. His Masters from the VTU in 2005. He received his Ph.D. in Computer Science and Engineering in 2015 from the Bharathiar University, India. He has received Best Employer of the Year 2013 from Acharya Institute of Technology, Best Consultant Award from Wipro Technologies during 2011. Dear Boss Award from FINRA Team-Wipro Technologies during 2010, Feather in My Cap Award from NYSE Team-Wipro Technologies during 2009. His research focuses on Data Warehouse and Data Mining, Network Security and Data Science. He can be contacted at email: manju.tn@bmsit.in.

Dr. Sheela Kathavate has received her B.E. degree in Computer Science and Engineering from Malnad College of Engineering, Hassan and Masters in 1995 from Louisiana state University, USA. She received her Ph.D. from VTU, Karnataka in 2019. She has worked for ITI Ltd., Bangalore and Bank One Corporation, USA, spanning over a period of 8 years and is in teaching for 15 years. Her areas of interest are Parallel Computing and Data Science. She can be contacted at email: sheela_cs@sirmvit.edu.

Dr. Rudresh Shirwaikar has a Ph.D. in Computer Science and Engineering from Manipal Institute of Technology, Manipal Academy of Higher Education (MAHE), Karnataka. Bangalore. His research areas include AI in Medicine, Data mining, Machine learning and Data warehousing. He has 15 journal articles and 10 conference proceedings published in reputed Scopus and web of science listed journals. He can be contacted at email: rudreshshirwaikar@gmail.com.

Anupchandra Rao Chandrakanth Rao has received B.E. degree in Computer Science and Engineering in 2019 at BMS Institute of Technology and Management, Bangalore. His research focuses on Bigdata, Internet of Things and Data Science.

Jayachnadra Rohith has received B.E. degree in Computer Science and Engineering in 2019 at BMS Institute of Technology and Management, Bangalore. His research focuses on Bigdata, Internet of Things and Data Science.