IoT Based WSN – Density Based Traffic Controller

Dharshni Prabha P[1], Dr.Jagatheesan K[2]
[1]*Department of Electrical and Electronics Engineering, Paavai Engineering College, Namakkal, Tamilndau. India*
[2]*Associate Professor, Department of Electrical and Electronics Engineering, Paavai Engineering College, Namakkal, Tamilndau. India*

Abstract **-Internet of things (IoT) is the interconnection of uniquely identifiable embedded computing devices within the existing infrastructure. It also uses the wireless sensor networks which is a group of specialized transducers with a communications infrastructure for monitoring and recording conditions at diverse locations. The proposed system (Vehicle Traffic Monitoring) has a central microcontroller at every junction which receives data from tiny wireless sensor nodes place on the Road. Traffic density is calculated with the help of IOT trans-receivers and the information is passed to cloud server which will give prior information regarding traffic jam at junctions and ambulance route clearance system.**

Keywords—Electric vehicle, location, optimization, gis, IoT

INTRODUCTION

In the world of Innovative and automotive world everything is getting computerized. Each data is in effect effortlessly available. Yet, the movement signals checking is as yet done physically. The activity signals are observed physically from the control room by the executives or a predictable time is settled for signals evolving. Rather than this a computerized controller-based activity checking framework will be useful for controlling the movement. This plan of movement foundation will be useful in decreasing the activity clog issue in urban communities. This paper depicts a framework where IR sensors are incorporated with an Arduino to work the paths which measure the movement thickness. This incorporated arrangement of movement is Internet of Things (IoT) based which likewise empowers to clear the activity for emergency vehicle by giving a catch in rescue vehicle so the activity gets cleared on that side. It additionally empowers the vehicles tally that move over the sensors. Subsequently, movement controlling gets upgraded effectively, which in the end prompts huge change in rush hour gridlock framework.

LITERATURE REVIEW

A. LITERATURE REVIEW

1. Wang et al. proposed a realistic strategy for actual-time traffic maintenance offloading in fog-dependent Internet of Vehicles (IoV) systems, intending to reduce the mean reaction time for incidents provided by vehicles. By partitioning the offloading optimization problem into two sub-problems and scheduling traffic flows amid distinct fog nodes, an approximation solution is devised to solve it. They will look into using automobiles outside of the communication ranges of "road side units (RSUs)" as fog nodes to offload loads for TMS in the future. As a result of this, it is tasked with processing all of the system's communications, which might lead to excessive resource usage.
2. Yuan et al. cond Yuan et al. [8] conducted a comprehensive review of 259 publications released in the last decades, as well as relevant performances before 2010, to better identify prospective research subjects and learn how to use relevant visual analytics approaches. They created taxonomy with three first-level classifications: methods before model construction, methods during model construction, and methods after model construction. Each class is further defined by a series of relevant analysis tasks, each of which is typified by a collection of current gained works. They also addressed and highlighted research problems as well as interesting future research prospects relevant to visual analytics.
3. Sumi and Ranga suggested a smart traffic management approach for nations based on the IoT and the vehicular ad hoc network principles (VANET). Emergency vehicles are prioritized in the proposed approach for a smooth flow through traffic depending on the kind of occurrence. It directs ambulances to the smallest probable routes to their destination, and it also offers a way to identify and react to traffic signal

hacking. In terms of congestion avoidance, travel duration, and energy consumption, our solution exceeds these suggestions for emergency vehicle systems.

3. Ning et al. Proposed a realistic strategy for minimizing traffic management services response time by permitting real-time content distribution in IoV systems depending on diverse network access. For large-scale IoV systems, they initially create a crowd sensing-dependent framework. Furthermore, to provide timely replies for traffic control, a cluster-based optimization framework is explored. They assume that the messages generated by vehicles are trustworthy. This may not always be the case, however, due to the possibility of false or inaccurate information being broadcast across the network to mislead other vehicles and traffic control systems.

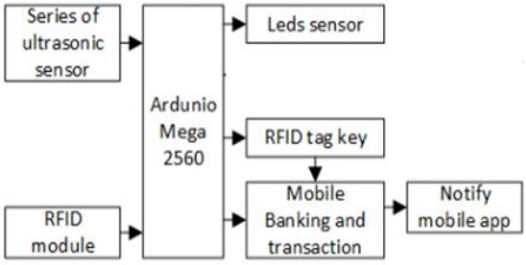
4. Lee et al. introduced an interactive visual analytics system that uses vehicle detection information to support traffic congestion investigation, monitoring, and predictions. This visual analytics technology is made to permit customers to investigate the origins, routes, and intensity of traffic congestion. The congested circumstances of a town are depicted utilizing a volume-speed rivers visualization, which shows traffic levels and velocities at the same time. However, they found no evidence of enhanced performance as a consequence of the data in the experiments.

5. Riveiro et al. proposed a visual analytics framework that supports: (1) multidimensional road traffic information analysis; (2) examination of normal behavioral models generated from information; (3) abnormal activity identification; and (4) abnormal incident explanations. The experts also identified several issues that need to be addressed, including the need to improve the assessment of observed abnormalities, as determining why the identified incidents are abnormal remains a challenge, and the limitations of the circular layout when a large number of attributes are chosen.

EXISTING SYSTEM

The main purpose of this paper is to invent an intelligent system which can make decisions for luminous control (ON/OFF/DIM) considering the light intensity. Here the day and night mode can be identified by fixing a particular intensity value on LDR sensor and street light can be controlled by IR sensor. The interesting part of this paper is the installation of solar cell for the power supply but in course of circumstances, if the solar cell is unable to do so, a secondary backup DC current will maintain the situation immediately. Another remarkable part of this project is to maintain the traffic signal automatically without any help of traffic police and monitor the entire system through internet by installing surveillance camera. All the components of this project are very simple and cost effective but efficient to make a reliable intelligence system.

B. BLOCK DIAGRAM



Existing block diagram

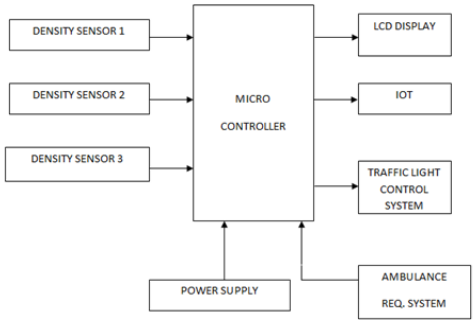
PROPOSED SYSTEM

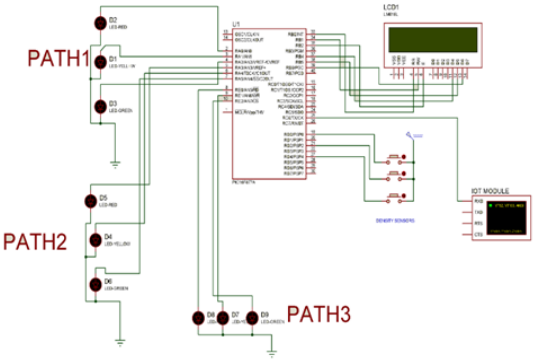
Our proposed system consists of 3 working modes,

- Density based traffic light signal controller.
- Wireless based Emergency vehicle clearance.
- IoT enabled traffic gateway.
- Peak time oriented traffic management.

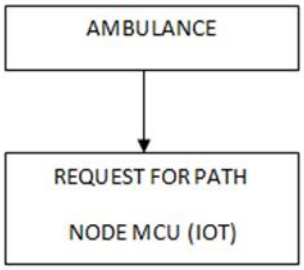
System will be intelligent and will calculate the time every time based on the density and operate in a cyclic clockwise signal lights control. Maximum and minimum time limit will be maintained. In this system, the primary aim is to gather the information of moving emergency vehicles based on WSN to provide them a clear path till their destinations and traffic signals should switch automatically to give a clear way for these emergency vehicles .IOT is placed on the emergency vehicles.IOT receiver protocol is placed at every traffic light intersection. Whenever emergency vehicle comes in the region the receiver collects the information and send this to micro controller. Micro controller holds the green signal for particular lane up to vehicle is cleared from that place. Along with this concept the peak time management will clear the traffic conjunction at the high traffic area depends on the time.

BLOCK DIAGRAM





Signal side block diagram



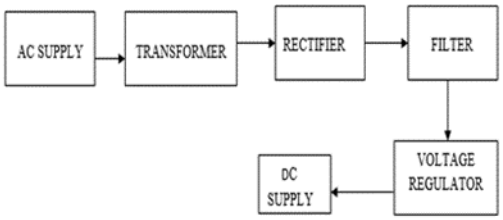
Ambulance side block diagram

First of all the power supply is given to entire components by means of the regulated power supply using the step-down transformer and the rectifier unit. Our controller will Have the native code for the time based and density based and the ambulance management signals. If any request detected from the sensor side and the IOT side the proposed PIC16F877A controller will trigger the traffic lights by toggling. At the mean time the LCD display will intimate the current status of the operation.

At the ambulance side the ambulance in-charge or the driver will give the request for the path where they need to go by the selection buttons. The selection request will passed by the IOT module to the traffic management board, then the signal arrangement will make a green path or traffic clearance system.

CIRCUIT DIAGRAM

POWER SUPPLY



A power supply (sometimes known as a power supply unit or PSU) is a device or system that supplies electrical or other types of energy to an output load or group of loads. The term is most commonly applied to electrical energy supplies, less often to mechanical ones, and rarely to others.

All digital circuits work only with low DC voltage. A power supply unit is required to provide the appropriate voltage supply. This unit consists of transformer, rectifier, filter and a regulator. AC voltage typically of 230Vrms is connected to a transformer which steps that AC voltage down to the desired AC voltage level. A diode rectifier then provides a full wave rectified voltage that is initially filtered by a simple capacitor filter to produce a DC voltage. This resulting DC voltage usually has some ripple or AC voltage variations. Regulator circuit can use this DC input to provide DC voltage that not only has much less ripple voltage but also remains in the same DC value, even when the DC voltage varies, or the load connected to the output DC voltage changes. The required DC supply is obtained from the available AC supply after rectification, filtration and regulation. Block diagram of power supply

The main components used in the power supply unit are Transformer, Rectifier, Filter and Regulator. The 230V AC supply is converted into 9V AC supply through the transformer. The output of the transformer has the same frequency as in the input AC power. This AC power is converted into DC power through diodes. Here the bridge diode is used to convert AC supply to the DC power supply. This converted DC power supply has the ripple content and for normal operation of the circuit, the ripple content of the DC power supply

should be as low as possible. Because the ripple content of the power supply will reduce the life of the circuit. So to reduce the ripple content of the DC power supply, the large value of capacitance filter is used.
This filtered output will not be the regulated voltage. For this purpose IC7805 regulator IC is used in the circuit.

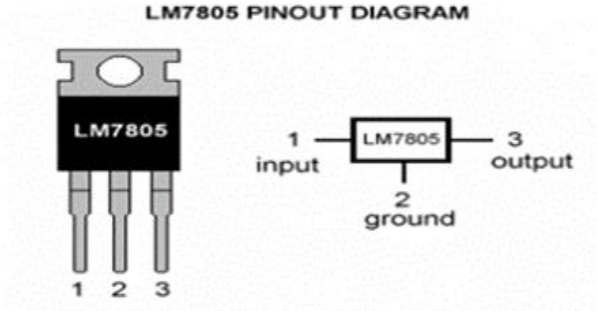
TRANSFORMER
Transformer is a device used either for stepping-up or stepping-down the AC supply voltage with a corresponding decreases or increases in the current. Here, a transformer is used for stepping-down the voltage so as to get a voltage that can be regulated to get a constant 5V.

RECTIFIER
A rectifier is a device like semiconductor, capable of converting sinusoidal input waveform units into a unidirectional waveform, with a nonzero average component.

FILTERS
Capacitors are used as filters in the power supply unit. The action of the system depends upon the fact, that the capacitors stores energy during the conduction period and delivers this energy to the load during the inverse or non-conducting period. In this way, time during which the current passes through the load is prolonged and ripple is considerably reduced.

VOLTAGE REGULATOR
The LM78XX is three terminal regulator available with several fixed output voltages making them useful in a wide range of applications. IC7805 is a fixed voltage regulators used in this circuit.

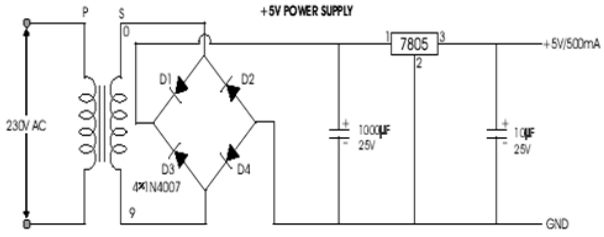
REGULATOR 7805
Voltage sources in a circuit may have fluctuations resulting in not providing fixed voltage outputs. A voltage regulator IC maintains the output voltage at a constant value. 7805 IC, a member of 78xx series of fixed linear voltage regulators used to maintain such fluctuations, is a popular voltage regulator integrated circuit (IC). The xx in 78xx indicates the output voltage it provides. 7805 IC provides +5 volts regulated power supply with provisions to add a heat sink. All voltage sources cannot able to give fixed output due to fluctuations in the circuit. For getting constant and steady output, the voltage regulators are implemented. The integrated circuits which are used for the regulation of voltage are termed as voltage regulator ICs. Here, we can discuss about IC 7805.The voltage regulator IC 7805 is actually a member of 78xx series of voltage regulator ICs. It is a fixed linear voltage regulator. The xx present in 78xx represents the value of the fixed output voltage that the particular IC provides. For 7805 IC, it is +5V DC regulated power supply.



7805 IC Rating:

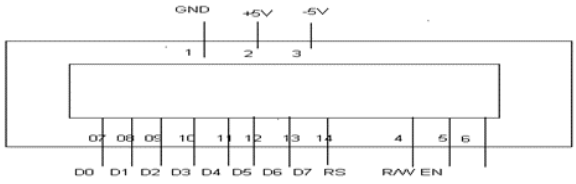
- Input voltage range 7V- 35V
- Current rating Ic = 1A
- Output voltage range VMax=5.2V, VMin=4.8V.

The purposes of coupling the components to the IC7805 are explained below. C1- It is the bypass capacitor, used to bypass very small extent spikes to the earth. C2 and C3- They are the filter capacitors. C2 is used to make the slow changes in the input voltage given to the circuit to the steady form. C3 is used to make the slow changes in the output voltage from the regulator in the circuit to the steady form. When the value of these capacitors increases, stabilization is enlarged. But these capacitors single-handedly are unable to filter the very minute changes in the input and output voltages. C4- like C1, it is also a bypass capacitor, used to bypass very small extent spikes to the ground or earth. This is done without influencing other components.



Applications of Voltage Regulator 7805 IC

PIN DIAGRAM OF LCD



PIN DESCRIPTIONS: -

Vcc, Vss and Vee: -

While Vcc and Vss provide +5V and ground respectively, Vee is used for controlling LCD contrast.

RS Register Select: -

There are two very important registers inside the LCD. The RS pin is used for their selection as follows.

If RS=0, the instruction command code register is selected, allowing the user to send a command such as clear display, cursor at home, etc.

If RS=1, the data register is selected, allowing the user to send data to be displayed on the LCD.

R/W, read/write: -

R/W input allows the user to write information to the LCD or read information from it.

R/W = 1 for reading.

R/W= 0 for writing.

EN, enable: -

The LCD to latch information presented to its data pins uses the enable pin. When data is supplied to data pins, a high–to-low pulse must be applied to this pin in order for the LCD to latch in the data present at the data pins. This pulse must be a minimum of 450 ns wide.

D0 – D7: -

The 8–bit data pins, DO – D7, are used to send information to the LCD or read the contents of the LCD's internal registers.

To display letters and numbers, we send ASCII codes for the letters A–Z, a-z numbers 0-9 to these pins while making RS=1.

There are also instruction command codes that can be sent to the LCD to clear the display or force the cursor to home position or blink the instruction command codes.

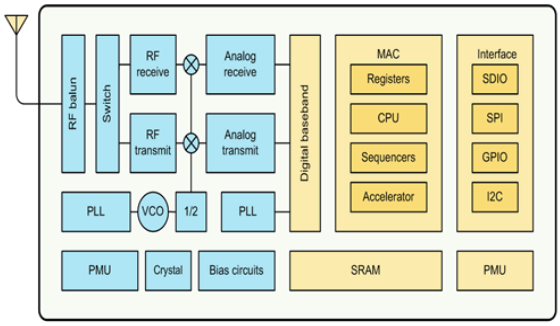
We also use RS = 0 to check the busy flag bit to see if the LCD is ready to receive information. The busy flag is D7 and can be read when R/W=1 and RS=0, as follows: if R/W = 1, RS = 0. When D7= 1 (busy flag = 1), the LCD is busy taking care of internal operations and will not accept any information.

CORE ARCHITECTURE

The PIC architecture is characterized by its multiple attributes:

- Separate code and data spaces (Harvard architecture).
- A small number of fixed length instructions
- Most instructions are single cycle execution (2 clock cycles, or 4 clock cycles in 8-bit models), with one delay cycle on branches and skips
- One accumulator (W0), the use of which (as source operand) is implied (i.e. is not encoded in the opcode)
- All RAM locations function as registers as both source and/or destination of math and other functions.
- A hardware stack for storing return addresses
- A small amount of addressable data space (32, 128, or 256 bytes, depending on the family), extended through banking
- Data space mapped CPU, port, and peripheral registers
- ALU status flags are mapped into the data space
- The program counter is also mapped into the data space and writable (this is used to implement indirect jumps).

ESP 8266:



Express if Systems' Smart Connectivity Platform (ESCP) is a set of high performance, high in tegration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed WiFi capabilities within others stems, or to function assist and alone application, with the lowest cost, and minimal space requirement.

Features:

- 802.11b/g/n
- Integrated low power 32-bitMCU
- Integrated 10-bitADC

- Integrated TCP/IP protocol stack
- Supports antenna diversity
- WiFi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes

IOT

The 'Thing' in iotcan be any device with any kind of built-in-sensors with the ability to collect and transfer data over a network without manual intervention. The embedded technology in the object helps them to interact with internal states and the external environment, which in turn helps in decisions making process. In a nutshell, iotis a concept that connects all the devices to the internet and let them communicate with each other over the internet. Iotis a giant network of connected devices – all of which gather and share data about how they are used and the environments in which they are operated.

A developer submits the application with a document containing the standards, logic, errors & exceptions handled by him to the tester. Again, if there are any issues Tester communicates it back to the Developer. It takes multiple iterations & in this manner a smart application is created.

Similarly, a room temperature sensor gathers the data and sends it across the network, which is then used by multiple device sensors to adjust their temperatures accordingly. For example, refrigerator's sensor can gather the data regarding the outside temperature and accordingly adjust the refrigerator's temperature. Similarly, your air conditioners can also adjust its temperature accordingly. This is how devices can interact, contribute & collaborate.

SOFTWARE REQUIREMENT:

EMBEDDED C:

Embedded C is one of the most popular and most commonly used Programming Languages in the development of Embedded Systems. So, in this article, we will see some of the Basics of Embedded C Program and the Programming Structure of Embedded C.

Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

Before digging in to the basics of Embedded C Program, we will first take a look at what an Embedded System is and the importance of Programming Language in Embedded Systems

EMBEDDED SYSTEM:

An Embedded System can be best described as a system which has both the hardware and software and is designed to do a specific task. A good example for an Embedded System, which many households have, is a Washing Machine.

We use washing machines almost daily but wouldn't get the idea that it is an embedded system consisting of a Processor (and other hardware as well) and software. Embedded Systems can not only be stand-alone devices like Washing Machines but also be a part of a much larger system. An example for this is a Car. A modern day Car has several individual embedded systems that perform their specific tasks with the aim of making a smooth and safe. Some of the embedded systems in a Car are Anti-lock Braking System (ABS), Temperature Monitoring System, Automatic Climate Control, Tyre Pressure Monitoring System, Engine Oil Level Monitor, etc.

PROGRAMMING IN EMBEDDED SYSTEM:

As mentioned earlier, Embedded Systems consists of both Hardware and Software. If we consider a simple Embedded System, the main Hardware Module is the Processor. The Processor is the heart of the Embedded System and it can be anything like a Microprocessor, Microcontroller, DSP, CPLD (Complex Programmable Logic Device) and FPGA (Field Programmable Gated Array).

All these devices have one thing in common: they are programmable i.e. we can write a program (which is the software part of the Embedded System) to define how the device actually works.

Embedded Software or Program allow Hardware to monitor external events (Inputs) and control external devices (Outputs) accordingly. During this process, the program for an Embedded System may have to directly manipulate the internal architecture of the Embedded Hardware (usually the processor) such as Timers, Serial Communications Interface, Interrupt Handling, and I/O Ports etc.

From the above statement, it is clear that the Software part of an Embedded System is equally important to the Hardware part. There is no point in having advanced Hardware Components with poorly written programs (Software).

There are many programming languages that are used for Embedded Systems like Assembly (low-level Programming Language), C, C++, JAVA (high-level programming languages), Visual Basic, JAVA Script (Application level Programming Languages), etc.

In the process of making a better embedded system, the programming of the system plays a vital role and hence, the selection of the Programming Language is very important.

DIFFERENCE BETWEEN C AND EMBEDDED C

There is actually not much difference between C and Embedded C apart from few extensions and the operating environment. Both C and Embedded C are ISO Standards that have almost same syntax, datatypes, functions, etc.

Embedded C is basically an extension to the Standard C Programming Language with additional features like Addressing I/O, multiple memory addressing and fixed-point arithmetic, etc.

C Programming Language is generally used for developing desktop applications whereas Embedded C is used in the development of Microcontroller based applications.

PIC C COMPILER:

CCS provides a method to attempt to make sure you can compile code written in older versions of CCS with minimal difficulty by altering the methodology to best match the desired version. Currently, there are 4 levels of compatibility provided: CCS V2.XXX, CCS V3.XXX, CCS V4.XXX and ANSI.

Notice: this only affects the compiler methodology, it does not change any drivers, libraries and include files that may have been available in previous versions.

#device CCS2

• ADC default size is set to the resolution of the device (#device ADC=10, #device
ADC=12, etc)
• boolean = int8 is compiled as: boolean = (int8 != 0)
• Overload directive is required if you want to overload functions
• Pointer size was set to only access first bank (PCM *=8, PCB *=5)
• var16 = NegConst8 is compiled as: var16 = NegConst8 & 0xFF (no sign extension)
• Compiler will NOT automatically set certain #fuses based upon certain code conditions.
• rom qualifier is called _rom

#device CCS3

• ADC default is 8 bits (#device ADC=8)
• boolean = int8 is compiled as: boolean = (int8 & 1)
• Overload directive is required if you want to overload functions
• Pointer size was set to only access first bank (PCM *=8, PCB *=5)
• var16 = NegConst8 is compiled as: var16 = NegConst8 & 0xFF (no sign extension)
• Compiler will NOT automatically set certain #fuses based upon certain code conditions.
• rom qualifier is called _rom

#device CCS4

• ADC default is 8 bits (#device ADC=8)
• boolean = int8 is compiled as: boolean = (int8 & 1)
• You can overload functions without the overload directive
• If the device has more than one bank of RAM, the default pointer size is now 16
(#device *=16)
• var16 = NegConst8 is will perform the proper sign extension
• Automatic #fuses configuration (see next section)

#device ANSI

• Same as CCS4, but if there are any discrepancies are found that differ with the ANSI standard then the change will be made to ANSI
• Data is signed by default
• cost qualifier is read-only RAM, not placed into program memory (use rom qualifier to Place into program memory)
• Compilation is case sensitive by default
• Constant strings can be passed to functions (#device PASS_STRINGS_IN_RAM)

CONCLUSION

Traditional system has many limitations to manage present increased traffic effectively. This paper proposed a smart TMS to control traffic situation more effectively and efficiently. By analyzing sensor data, it sets traffic signal time dynamically and sends the data

to a cloud server through a Wi-Fi module that is stored for further data analytics. It also deals with emergency vehicle and in case of traffic signal violation can detect and charge a fine that is paid through Traffic Wallet mobile app. The whole system is very cost effective than existing system in developing countries. But, the security should be ensured for sensors as these are equipped at roadsides. For further research, some features can be initiated to handle, this proposed system immediately to change the current terrible scenario of the traffic congestion

REFERENCE

[1] Costea, I.M. Nemtanu, F.C. Dumitrescu, C. Banu, C.V. Banu,G. S." Monitoring System with Applications in Road Transport". IEEE, (2014)

[2] Händel. P.Skog, I. "Cell phone Based Measurement Systems for Road Vehicle Traffic Monitoring and Usage-Based Insurance".vol. 8, no. 4. pp. 1238- - 1248. IEEE.(2014)

[3] An. S. Lee. B. Shin, D. "An overview of wise transportation frameworks." In: Computational Intelligenc Communication Systems and Networks (CICSyN), 2011 .Vol.2.pp.300-340.Third International Conference on. IEEE, (2011)

[4] A.P. Gokulan and D. Srinivasan, "Appropriated geometric fluffy multiagent urban traffic flag control," IEEE Trans. Intell. Transp. Syst. vol. considering a multi-objective approach Elec Power Syst Res, 199 (2021), p. 107391

[5] SagarSukode, ShilpaGite,"Vehicle Traffic Congestion Control And Monitoring System in IOT" ISSN 0973-4562 Volume10, Number8(2015) pp. 19513-19523

[6] Tejashri Gadekar1, Priyanka Chavare2, Komal Chipade3& P.S Togrikar4 'Actualizing Intelligent Traffic Control System for Congestion Control, Ambulance Clearance, and Stolen Vehicle Detection."Vol-2, Issue-4, 2016 ISSN: 2454-1362,

[7] Curiac, D.I. Wireless sensor network security enhancement using directional antennas: State of the art and research challenges. Sensors 2016, 16, 488. [Google Scholar] [CrossRef] [Green Version]

[8] George, R.; Mary, T.A.J. Review on directional antenna for wireless sensor network applications. IET Commun. 2020, 14, 715–722. [Google Scholar] [CrossRef]

[9] Arshad, B.; Ogie, R.; Barthelemy, J.; Pradhan, B.; Verstaevel, N.; Perez, P. Computer vision and IoT-based sensors in flood monitoring and mapping: A systematic review. Sensors 2019, 19, 5012. [Google Scholar] [CrossRef] [PubMed] [Green Version]

[10] Andrienko, G.; Andrienko, N.; Chen, W.; Maciejewski, R.; Zhao, Y. Visual analytics of mobility and transportation: State of the art and further research directions. IEEE Trans. Intell. Transp. Syst. 2017, 18, 2232–2249. [Google Scholar] [CrossRef]

[11] Zeng, W.; Lin, C.; Lin, J.; Jiang, J.; Xia, J.; Turkay, C.; Chen, W. Revisiting the modifiable areal unit problem in deep traffic prediction with visual analytics. IEEE Trans. Vis. Comput. Graph. 2020, 27, 839–848. [Google Scholar] [CrossRef] [PubMed]

[12] Wang, X.; Ning, Z.; Wang, L. Offloading in Internet of vehicles: A fog-enabled real-time traffic management system. IEEE Trans. Ind. Inform. 2018, 14, 4568–4578. [Google Scholar] [CrossRef]

[13] Yuan, J.; Chen, C.; Yang, W.; Liu, M.; Xia, J.; Liu, S. A survey of visual analytics techniques for machine learning. Comput. Vis. Media 2020, 7, 3–36. [Google Scholar] [CrossRef]

[14] Sumi, L.; Ranga, V. Intelligent traffic management system for prioritizing emergency vehicles in a smart city. Int. J. Eng. 2018, 31, 278–283. [Google Scholar]

[15] Wang, X.; Ning, Z.; Hu, X.; Wang, L.; Hu, B.; Cheng, J.; Leung, V.C. Optimizing content dissemination for real-time traffic management in large-scale internet of vehicle systems. IEEE Trans. Veh. Technol. 2018, 68, 1093–1105. [Google Scholar] [CrossRef]

[16] Gunda, P.K. Network-Wide Traffic Congestion Visual Analytics: A Case Study for Brisbane Bluetooth MAC Scanner Data. Ph.D. Thesis, Queensland University of Technology, Brisbane City, QLD, Australia, 2021. [Google Scholar]

[17] Lee, C.; Kim, Y.; Jin, S.; Kim, D.; Maciejewski, R.; Ebert, D.; Ko, S. A visual analytics system for exploring, monitoring, and forecasting road traffic congestion. IEEE Trans. Vis. Comput.

Graph. 2019, 26, 3133–3146. [Google Scholar] [CrossRef] [PubMed]

[18] Riveiro, M.; Lebram, M.; Elmer, M. Anomaly detection for road traffic: A visual analytics framework. IEEE Trans. Intell. Transp. Syst. 2017, 18, 2260–2270. [Google Scholar] [CrossRef]

[19] Nallaperuma, D.; Nawaratne, R.; Bandaragoda, T.; Adikari, A.; Nguyen, S.; Kempitiya, T.; De Silva, D.; Alahakoon, D.; Pothuhera, D. Online incremental machine learning platform for big data-driven smart traffic management. IEEE Trans. Intell. Transp. Syst. 2019, 20, 4679–4690. [Google Scholar] [CrossRef]

[20] Adu-Gyamfi, Y. GPU-Enabled Visual Analytics Framework for Big Transportation Datasets. J. Big Data Anal. Transp. 2019, 1, 147–159. [Google Scholar] [CrossRef] [Green Version]

[21] Lock, O.; Bednarz, T.; Pettit, C. The visual analytics of big, open public transport data–a framework and pipeline for monitoring system performance in Greater Sydney. Big Earth Data 2021, 5, 134–159. [Google Scholar] [CrossRef]

[22] De Souza, A.M.; Brennand, C.A.; Yokoyama, R.S.; Donato, E.A.; Madeira, E.R.; Villas, L.A. Traffic management systems: A classification, review, challenges, and future perspectives. Int. J. Distrib. Sens. Netw. 2017, 13, 1550147716683612. [Google Scholar] [CrossRef]