EAI Endorsed Transactions
on Internet of Things

Research Article EAI.EU

Semantic Interoperable Traffic Management Framework for IoT Smart City Applications

Sivadi Balakrishna[1],* and M. Thirumaran[2]

[1,2]Department of Computer Science and Engineering, Pondicherry Engineering College, Pondicherry, India.

Abstract

Real-time traffic monitoring and controlling are one of the biggest problems in this present living world. So many researchers have dealt with and put their effort into this problem, as a result, several types of approaches have developed. Currently using traffic monitoring and alert systems are not up to the needs of smart city applications and more expensive. This paper proposes an Internet of Things (IoT) based Smart Real-Time Traffic Monitoring System to provide better service with low cost for Smart city applications using semantic annotations. The proposed framework contains two phases namely-traffic monitoring unit and traffic reduction unit. The first phase analyses semantic traffic to detect an emergency, the latter phase removes redundant semantic information for traffic reduction. Simulation results suggest that the framework is capable of accurate and early detection of an emergency as well as traffic reduction while keeping sufficient information to report the emergency.

Keywords: Semantic Interoperability**,** Traffic Monitoring, Traffic detection, Smart City, IoT, Raspberry Pi, ThingSpeak, Cloud.

Received on 31 October 2017, accepted on 03 December 2017, published on 23 January 2018

doi: 10.4108/eai.11-9-2018.155482

1. Introduction

In recent years, the urban population growth is increasing tremendously. The global urban population is expected to grow 1.86% per every year from 2015 to 2020. In between 2020 to 2025 years, this increase will be 1.63% and 2025 to 2030 it goes to the 1.44% approximately by World Health Organization (WHO). In an urban population, there is huge popularity for cars and minimum two cars owned by one family. The popularity of private cars getting urban traffic is more. Because of this, traffic is becoming one of the big problems in urban areas across all over the world [1]. The heavy traffic in urban cities is leading to congestion, accidents that have affected by the property loss, waste of time, environmental pollution, and sometimes it goes to next level of man's death. Therefore, there is a big need for traffic monitoring and reduction system in urban areas to become Smart cities [2]. The best approach to solve this problem using the Internet of Things and it provides a new trend to intelligent traffic management**.**

To improve traffic conditions and traffic pressure, this paper intends to use the IoT technology, Cloud computing, Raspberry Pi and semantic annotation technologies. The actual scenario of this research is as follows: The information generated by IoT devices data collected from all roads through gateways can presented to all travelers and users. After collecting the real-time traffic data, the system can recognize the current traffic, traffic flow conditions and can predict the future traffic in urban

[*]Corresponding author. Email:balakrishna.sivadi@gmail.com

areas. After that semantic traffic monitoring and semantic traffic detection has been found for semantically interoperable the traffic data. Based on the system-generated data may useful to drivers for choosing optimal routes. Therefore, the system is dynamically administrative, control and monitor on moving cars. If constructions on intelligent traffic system using IoT, then there is a lot of benefits for users such as improving traffic conditions, less traffic jam and high reliability, traffic safety, management costs and independence of weather conditions. Bridges, roads, tunnels, vehicles, traffic signals and drivers these are some of the traffic elements in IoT. All these items will connected to the web for monitoring and identification through different types of IoT devices like RFID, sensors, actuators, Global Position Systems (GPS) and laser scanners. As of late fame of private engine, vehicles are getting urban activity more swarmed. As result activity, observing is getting to be noticeably one of the imperative issues in enormous smart city framework everywhere throughout the world. Some of these concerns are movement clog and mischances that normally cause a critical exercise in futility, property harm, and environmental contamination. Any kind of clog on streets, at last, prompts budgetary misfortunes. In this way, there is a pressing need to enhance traffic management. The presence of the Internet of Things (IoT) gives another pattern to canny traffic improvement.

This exploration proposes to utilize the IoT, specialist and other advances to enhance activity conditions and mitigate the activity weight. Data created by movement IoT and gathered on all streets can introduced to explorers and other clients. Through gathered ongoing movement information, the traffic monitoring system can perceive current activity operation, movement stream conditions and can foresee the future movement stream. The framework may issue a few most recent continuous moving data that helps drivers picking ideal paths. Thusly, the framework can absolutely administrate, screen and control on moving vehicles.

Building a smart movement framework in view of IoT has a number of advantages such as change of activity conditions, lessening the car influx and administration costs, high unwavering quality, traffic security and freedom of climate conditions. Such movement of an IoT must incorporate each component of traffic such as streets, spans, burrows, traffic signs, vehicles, and also drivers [3]. Every one of these things will be associated with the web for helpful ID and administration through sensor devices, for example, RFID devices, infrared sensors, worldwide situating frameworks, laser scanners, and so forth. Undertaking the IoT gives motion data accumulation and combination, supports preparing an investigation of all classes of motion data on streets in an expansive territory and naturally. Hence, present-day traffic administration is developing into a wise transport framework in light of IoT [4].

The term ‘Traffic’ will play the key role for accessing the logistics and services available on the road, so the developed system will be more reliable and accurate towards the traffic management. The IoT of data will be produced along with various sensor based technologies. This movement is observing that applications should ensure to keep any type of security attack visit in urban cities. So there is such type of model executions can be found in [3, 4] and the Smart Santander EU venture.

The remainder of this paper is organised as follows: section 2 shows the related work of this study. The semantic interoperable real-time traffic management framework has been described in Section 3. The section 4 includes the experimental results and analysis. Finally, section 5 consists of Conclusion and future works.

2. Related Work

Real-time traffic monitoring and controlling are one of the biggest problems in this present living world. So many researchers have dealt and put their effort on this problem, as a result, several types of approaches have developed. Bhadra et al [5] has applied agent-based fuzzy logic technology involving multiple approaches and vehicle movements for traffic control situations. Based on a fuzzy neural network, has been proposed a traffic flow prediction mechanism in chaotic traffic flow time series. The authors Anupama Mallik et al. [6] developed various strategies for integrating dynamic data into Intelligent Transport Systems (ITS). P. Pyykonen et al. [7] applied in enterprise services for an effective integration of Service Oriented Architecture (SOA) and Internet of Things (IoT). Due to the revolution in Internet of Things (IoT), many of the researchers shifted their attention. So that more convenient environment [8] established by composing different types of intelligent systems in various domains. Ubicomp [9] and FeDNet [10] are different Internet of Things (IoT) systems for communication using message-passing techniques. The authors D. Bandyopadhyay and J. Sen et al. [11] proposed various applications applied in different domains like smart city, smart home, smart metering, smart transport and healthcare.

D. Singh et al. [12] the existing status of the Smart city transport management system, the researchers have been move on real-time transport systems using the Internet of Things (IoT). V.Katiyar et al. [13] proposed a technique for choosing movement blockage on roads using picture getting ready techniques and a display for controlling activity motions in light of information get from streets taken by a camcorder. We isolate activity thickness, which thinks about to total range controlled by vehicles all over the place similar to the total of pixels in a video layout instead of figuring a number of vehicles. We set two parameters for yield, variable activity cycle and weighted time for each road in perspective of activity thickness and control movement lights, successively it is extremely time unpredictable and in addition sweeping.

Therefore, P. Pyykonen et al. [14] the opportunity has already come and gone for limited to manage the road turned parking slot issue. There are diverse systems open for traffic organization, for example, video data examination, infrared sensors, inductive circle acknowledgment, remote sensor framework, and so on. Each one of these procedures are effective procedures for sharp activity organization. However, the issue with these systems is that the foundation time, the cost caused for the foundation and support of the system is high. From now on, another development called Radio Frequency Identification (RFID) is introduced C.Yulian et al. [15] which can joined with the present hailing structure that can go about as a key to splendid traffic organization constantly. Use of this new development will incite diminished traffic monitoring. Bottlenecks recognized early and thus early preventive 2016 International Conference on Emerging Technological Trends [ICETT] measures can take in this way saving time and cost of the driver. X. Yu et al. [16] presented display called dynamic movement observing framework. It implies that different influencing factors are ought to check. Durga Devi Sanju et al. [17] developed that GPS based vehicle ensuing framework. It gives to diminishing the short separation traveling's and in addition to same substance is concentrated on different variables are to be thinking about convenient information submitting by utilizing VSNs vector remove directing calculation and gives the high dependable correspondence. It covering the range extreme separation fire gathering information. Vijender Kumar Solanki et al. [18] made an RFID based conceptual model on insight into motion control framework. It controlling movement stream, decreasing mischances in rush hour gridlock spots and remote area transmission.

So, after careful observation of all these literature survey, there is a need for semantic annotations in traffic management systems.

These are the following contributions made on this proposed approach:

1. To detect the moving vehicles on busy highway road and correspondingly estimate the traffic congestion or load.
2. To analyze the gathered traffic data on providing some relevant traffic features to semantically interconnect and interpret the moving vehicle networks.
3. To estimate the causes for traffic jam by observing traffic data, and provide the solutions for choosing optimal routes.
4. To compare the results obtained both for traffic monitoring and traffic reduction in an efficient way.

3. IoT-based Semantic Interoperable Smart Real-Time Traffic Monitoring Framework

In this paper, to develop an Internet of Things (IoT) based smart real-time traffic monitoring system, used a ThingSpeak, Raspberry Pi 3 Model B, and a webcam to analyze a traffic on a busy highway. Then connecting the Raspberry Pi device to the machine for deploys a traffic-monitoring algorithm. ThingSpeak- a cloud aggregator used to store the data, analyze and visualize the data online.

In the Internet of Things (IoT), the analytics are used anywhere:

1. The edge node
2. Offline on the desktop
3. In the cloud

Here we have shown how to monitor the real-time traffic data on the busy highway and how the real-time data collected on the cloud. After storing, the aggregated data on the cloud can perform analytics for the edge device and finally perform analysis on the cloud data. The Fig.1 illustrates a framework of workflow analysis and perform online analysis on the stored cloud data. In this framework, uses ThingSpeak for both storage and analysis.

The proposed semantic traffic management framework based on Internet of Things may consists of four layers as shown in fig. Those are the Physical layer, Data processing layer, Semantic analysis layer, and Application layer.

1. **Physical layer:** This layer is used to collect the data coming from different IoT devices. This is the lowest layer observed in the proposed framework. It captures the various heterogeneous types of data acquiring from heterogeneous IoT devices like sensors, actuators, WSN, and cameras etc. The high amount of big data is transferred to the next level layer i.e., Data processing layer for performing the data analysis and processing raw data.
2. **Data processing layer:** This layer is used for acquiring the needed data from the physical layer collected raw data. The data processing layer mainly consists of two sub modules namely- Feature extraction and

annotated data. The feature extraction is used to extract raw data by applying the camera vision techniques like SPP-net and GMM to manage the real-time smart traffic data and detect the high traffic congested data on busy roads. The acquired data is sometimes may be simple or sometimes may be complex. So based on representation of data, both feature extraction and annotated data is transfer to the next layer i.e. semantic analysis layer for performing meaningful data as well as interpretation by providing ontology based semantic techniques.

3. **Semantic analysis layer:** This is the crucial layer for this proposed framework. The Simulink model has been applied on this layer. It will takes the RDF or triple store data to semantically annotate the traffic information. The ambiguity of the annotated data is minimized by applying the Simulink model. The ontology based domain knowledge systems is used the context aware data to predict the real-time traffic data for reducing the traffic jams and unforeseen delay.
4. **Application layer:** This layer is used to perform analysis and visualization of the semantically annotated data to the users. This layer is generate the traffic alerts, count of vehicles in real-time situations.

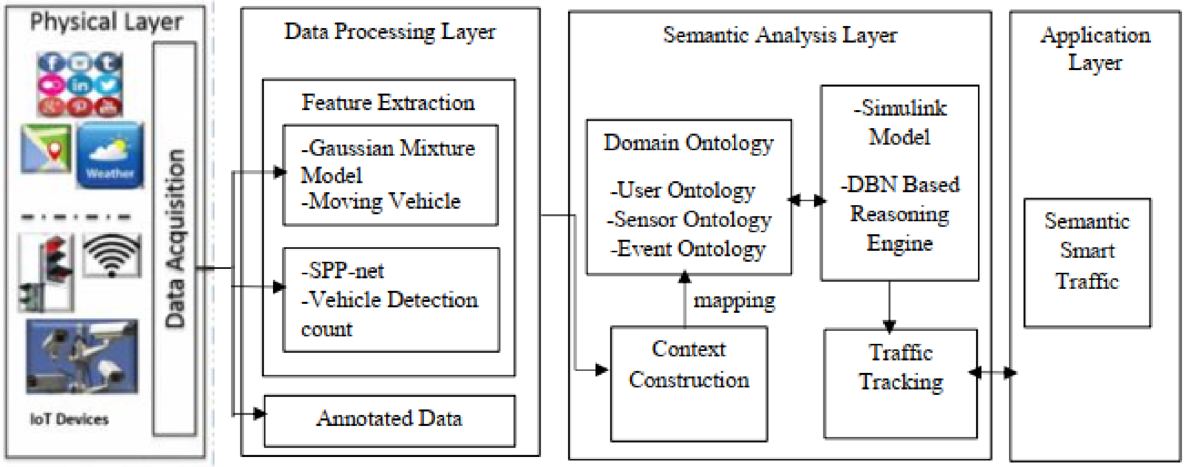


Fig.1. Semantic Smart Traffic Framework

3.1 Hardware Setup

In order to develop traffic monitoring system, we used Raspberry Pi 3 Model B and USB webcam. We placed a USB webcam near a window on the second floor of my house building that overlooks national highway road 2, Tirupati, India. To get the clear view of vehicles places the webcam may place both sides of the highway. The Fig.2 depicts the Raspberry Pi 3 Model B and it mainly having two USB ports, switching power supply cable with 5V 2A and Wi-Fi/WLAN for internet connectivity. The USB webcam connected to one USB port and Wi-Fi/WLAN dongle on another USB port of the Raspberry Pi. Finally, connect the Raspberry Pi to a WLAN in the house building.

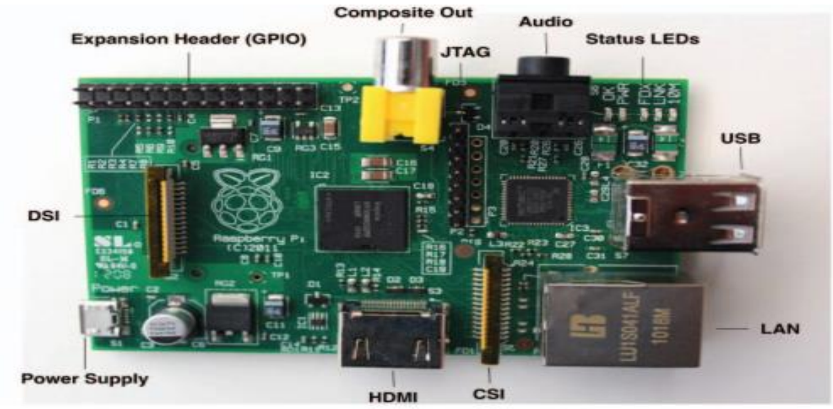


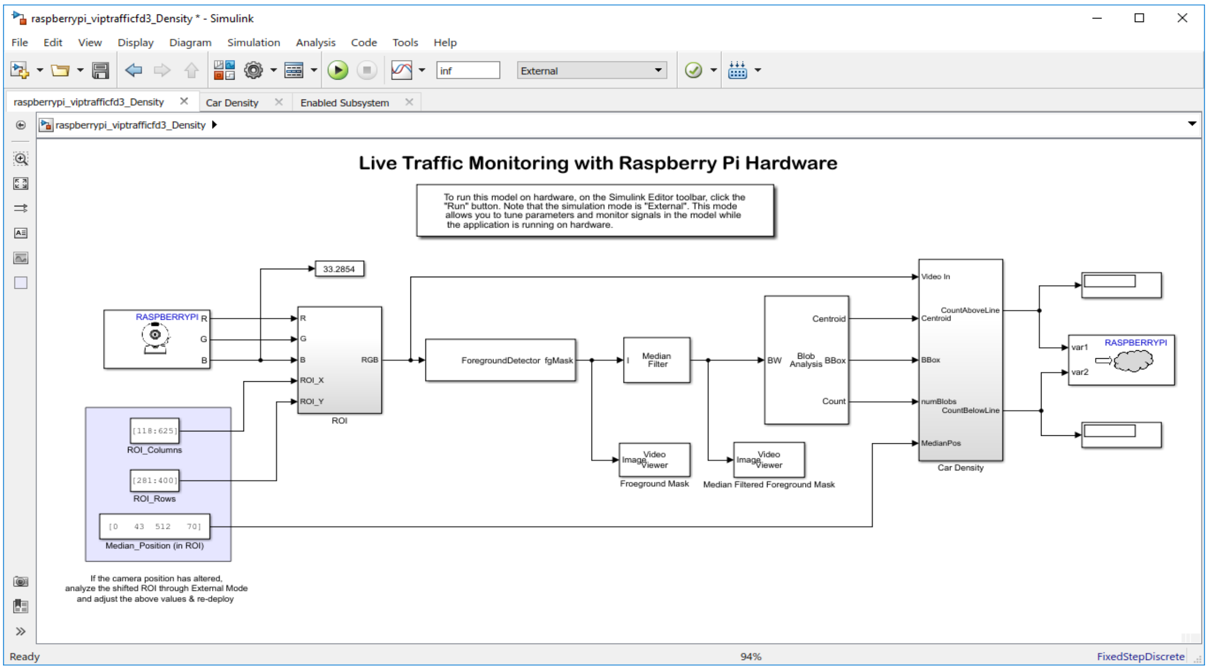
Fig.2. Raspberry Pi 3 Model B

3.2 Deploying the Algorithm to the Hardware

In order to develop the traffic monitoring algorithm, we used following a list of parts available in MATLAB 2016a.

1. Simulink
2. Image processing Toolbox
3. Computer Vision System Toolbox
4. Simulink Support Package for Raspberry Pi 3 Hardware.

Simulink model have developed by MathWorks and integrated with MATLAB – is a graphical modelling language tool for simulating, modelling and analyzing multi-domain systems. The MATLAB algorithm is incorporate into Simulink model and then exports the results into MATLAB for analysis and visualizations. It supports simulation, system level design, automatic code generation, testing and visualization of IoT data. In this example, we develop a model using Simulink and generated code that runs on the Raspberry Pi 3 model B. the following Fig. 3 shows the Simulink model for traffic monitoring.



To run this model on hardware, on the Simulink Editor toolbar, click the "Run" button. Note that the simulation mode is "External". This mode allows you to tune parameters and monitor signals in the model while the application is running on hardware.

If the camera position has altered, analyze the shifted ROI through External Mode and adjust the above values & re-deploy

Fig.3. Simulink model for Traffic Monitoring

It supports simulation, system-level design, automatic code generation, testing and visualization of IoT data. In this experiment, developed a model using Simulink and that runs on the Raspberry Pi 3 model B. the above Fig. 3 shows the Simulink model for traffic monitoring. In this Simulink model, an external capability of Simulink is used to develop the algorithm. In external mode, Simulink gathers the real-time traffic streaming data from Raspberry Pi and the users can see the video on desktop/mobile using the SDL video display while the Simulink model is running.

The USB cam placed for a particular region of highway and vehicles are moving from left to right. The USB webcam connected to one USB port of the Raspberry Pi captures video with a selected region. Next, authors are used the ForegroundDetector for estimating low vision and pixels of a video sequence captured by the USB webcam as shown in Fig.4

It estimates mainly using Gaussian mixture models and produces the appropriate results on moving cars. To avoid and remove the unwanted noise in the ForegroundDetector mask post-processed by using MedianFilter for analyzing the Post-processed data used the Blob Analysis block- that computes centroids of the blob containing the cars. The Car Density block counts the number of vehicles travelling both eastbound traffic and westbound. Traffic direction in that video frame. This Car Density Block divides the video capturing region into two sections along with the highway median. After that, the real-time traffic streaming data sent to ThingSpeak-the data aggregator for analyzing & visualizations. Here individually we sent vehicle count value to each eastbound and westbound region. We send both eastbound traffic and westbound traffic values to ThingSpeak (Channel

Id) and store the westbound value in field1 and store the eastbound value in field2.

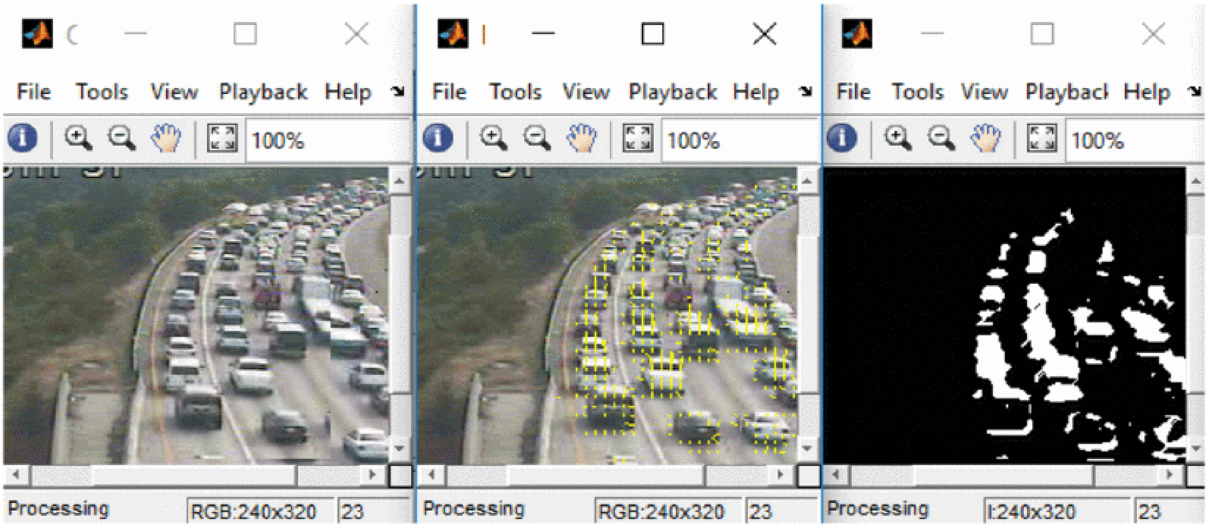

File Tools View Playback Help
100%
Processing
RGB:240x320
23
File Tools View Playback Help
100%
Processing
RGB:240x320
23
File Tools View Playbacl Help
100%
Processing
I:240x320
23

Fig. 4. Estimating total vehicles count (east bound and west bound traffic)

The Algorithm 1 depicts to the usage of this Simulink model based on this semantic annotated algorithm as shown below.

Algorithm 1: For semantically measured traffic data.

Input: A group of S snapshots along with input videos IV (Input Video).

Output: Generating high traffic load or low traffic load.

```
for S snapshots in IV do
    Calculate the moving vehicles $C^t_{mv}$ and correspondingly total vehicles $C^t_{tv}$ in a taken frame
    Find out the stopped vehicles $C^t_{sv} = C^t_{tv} - C^t_{mv}$
    Count the threshold value , if $C^t_{sv}$ > threshold value
        then
            High traffic load
        else
            Low traffic load
        end if
end for
repeat
```

3.4 Analyzing Data on ThingSpeak

The traffic-monitoring model done by using Simulink deployed into Raspberry Pi hardware device. Then we can start analysis on real-time disseminate data stored in the ThingSpeak cloud aggregator sent by the Raspberry Pi 3 Model B fetching from ThingSpeak.

3.4.1 Reading one week of data into MATLAB

Already we discussed that ThingSpeak allows receiving data from Raspberry Pi once every 15 seconds. In addition, it retrieves approximately 40,000 data points only. In a single read operation, ThingSpeak allows only 8000 points. Therefore, we need to create for loop to gather the real-time traffic data in batches. Here each iteration divides into two vectors namely all traffic data and timestamp and then append the traffic and time as shown in Appendix A.1.

Plotting traffic data and next, we need to plot the graph as well as label the graph. From the raw data, the daily fluctuations in traffic are clearly visible. Daily fluctuations are hardly changing from weekdays to normal days. We can observe a different pattern on the weekdays (18/12-22/12) compared to weekend days (16/12 and 17/12). Due to communication traffic, the weekdays show stronger in morning and peaks in afternoon.

3.5 Viewing at Day-to-day Mean Bulk for Eastbound and Westbound Traffic and Plotting

In order to visualize (or) measure the traffic density each day, first, we can average the traffic density observations. A bar chart used to visualize data. We observed that Monday traffic is higher than both eastbound traffic and westbound traffic that follows the weekend. This flow of execution code shown in Appendix A.2.

The Appendix A.3 shows the code for drawing a one-week density data using both eastbound and westbound traffic information. If you observe the Fig.5, the total number of moving vehicles is to be counted on both westbound and eastbound traffic density information on starting date as 16th December to ending date as 23rd December. The corresponding traffic counting using eastbound and westbound traffic density shown in Fig.6 (a) and (b) respectively.

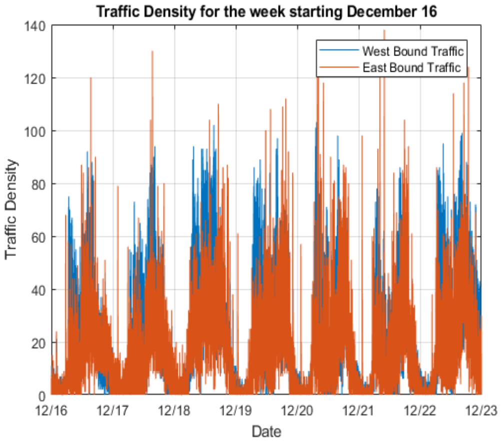
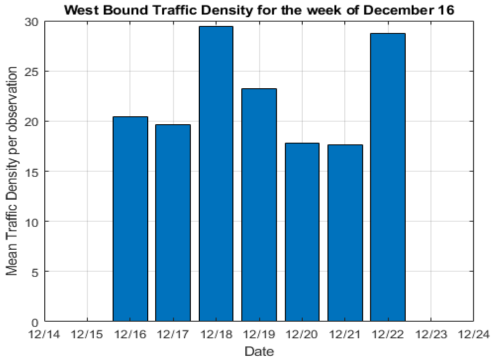
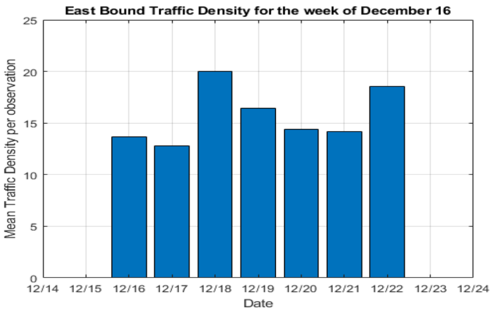


Fig.5. Traffic Density for one week



(a)



(b)

Fig.6. Traffic Density (a) Westbound (b) East Bound

3.5.1 Taking a closer look at individual days during the week and on the weekend

Look at traffic on the individual days, to gain more insight into our traffic data. For that, we can take each day, specify a start time, and stop time code shown in Appendix A.4.

3.5.2 Fetch each individual day in a loop

We retrieve all the 7 days data from ThingSpeak and down sample the real-time traffic data to remove short-term fluctuations. To find times and plot where the traffic density is the highest use the findpeaks function and observe for each day. We look at the eastbound traffic data for simplicity. The code for this process is shown in Appendix A.5.

3.5.3 Estimating Traffic density

The estimation traffic density is based on DBN (Dynamic Bayesian Networks) for finding situation tracking in real-time. The Algorithm 1 is shows the total steps for measuring traffic.

1. **Detecting moving and non-moving vehicles**: for finding the total count of vehicles in an appropriate video, have to use the GMM and SPP-net models.
2. **Calculate the stopped vehicles and moving vehicles from total count of vehicles**: In order to calculate the both stopped vehicles and moving vehicles, first find out the velocity of the vehicles. If the velocity is zero in position, then the status of that vehicle is stopped vehicles $C^{t}{}_{sv,}$ otherwise it is counted as moving vehicle $C^{t}{}_{mv}$. SPP-net model is used for finding the count of moving vehicle is:
   $C^{t}{}_{sv}= C^{t}{}_{tv} - C^{t}{}_{mv}$, here t ranges between 1 less than are equal to n.

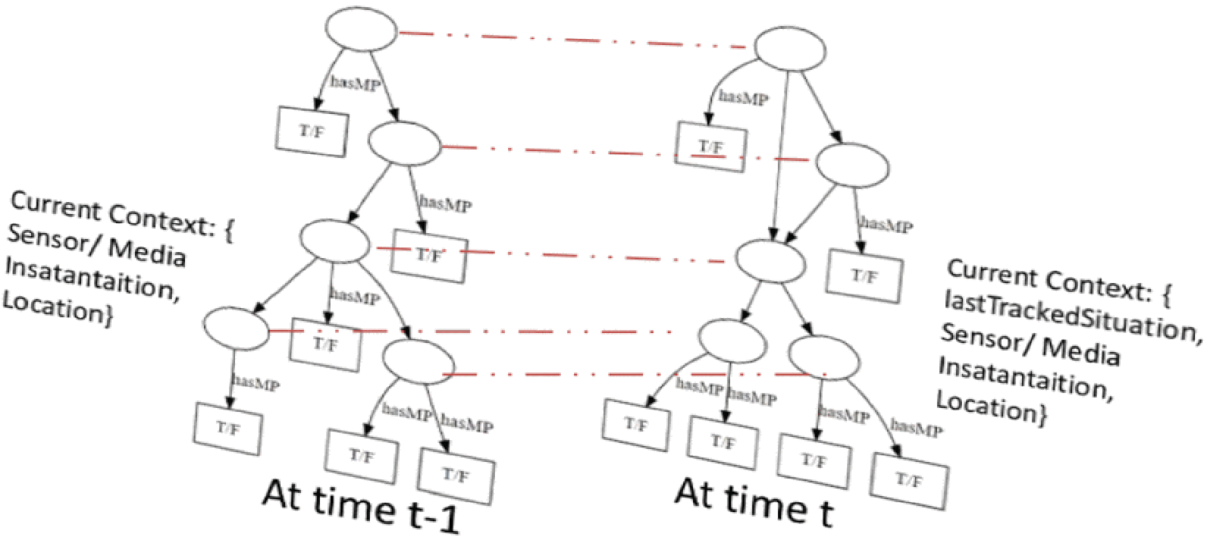

hasMP
T/F
Current Context: { Sensor/ Media Insatantaition, Location}
Current Context: { lastTrackedSituation, Sensor/ Media Insatantaition, Location}
At time t-1
At time t

Fig.7. DBN model for semantic traffic

3.6 Downsample into 48 silos of approximately 30-minute masses of data and find heights

The generated real-time raw traffic data is hard to visualize and very spiky also. Therefore, do you want to see the highest volume in a particular time in a day then we need to look at the data on a time scale greater than 15 seconds do. For calculation of this divide, the 24-hour day into 30 minutes segments. Here each segment begins near the top of the hour and means while end time at 30 minutes later. Fig. 9 shows the individual date traffic information.

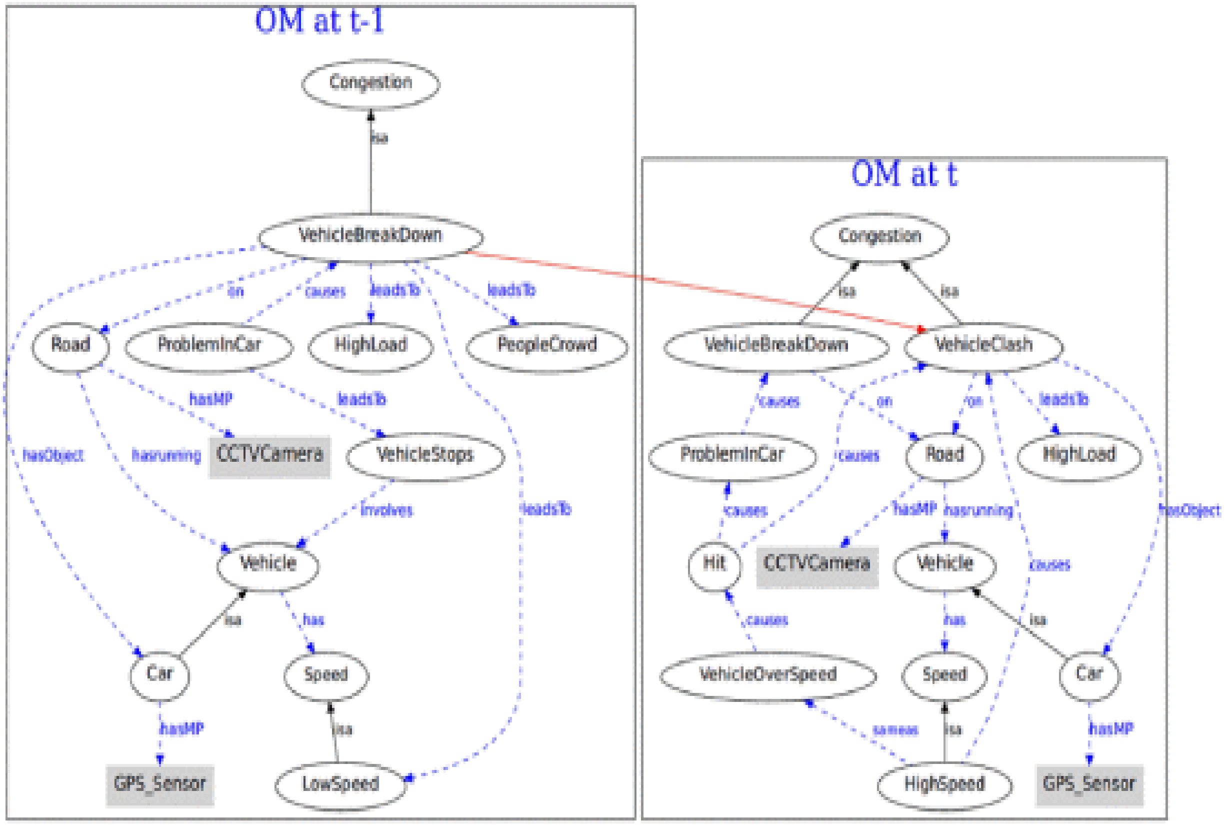
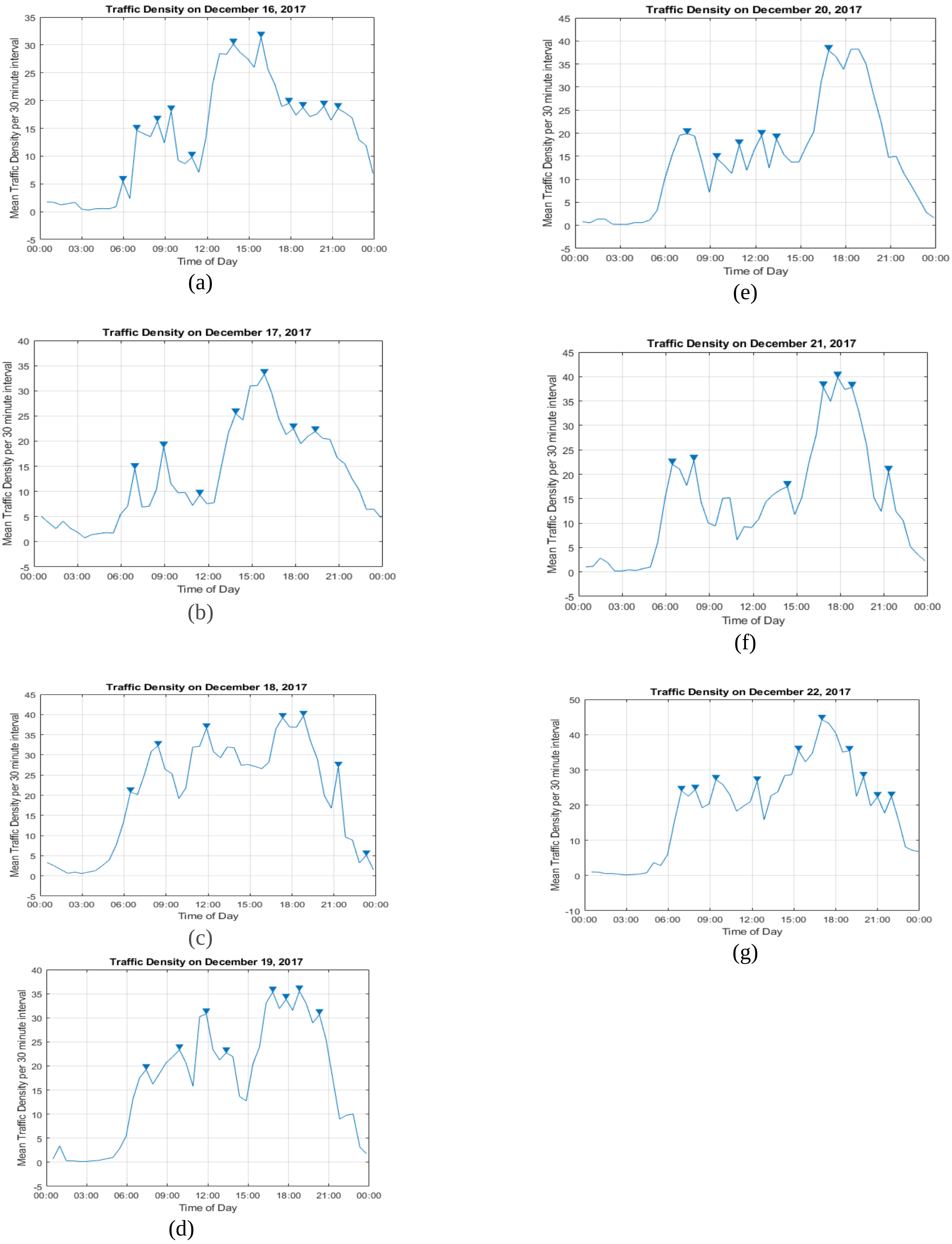


Fig.8. DBN based result for tracking situation in smart traffic



Traffic Density on December 16, 2017

(a)

Traffic Density on December 17, 2017

(b)

Traffic Density on December 18, 2017

(c)

Traffic Density on December 19, 2017

(d)

Traffic Density on December 20, 2017

(e)

Traffic Density on December 21, 2017

(f)

Traffic Density on December 22, 2017

(g)

Fig. 9. Traffic Density on day wise- (a). On December 16 (b). On December 17 (c). On December 18 (d). On December 19 (e). On December 20 (f). On December 21 (g). On December 22

3.6.1 Weekend observations for Saturday, December 16

This is first-weekend day observation of traffic pattern and many peaks are scattered throughout the day but the maximum peak times occur at 1:52 and 3:51 time. This is not surprising as the highway connects many big industries and data centers that frequented during weekend days only.

3.6.2 Weekend observations for Sunday, December 17

On Sunday, the traffic patterns are similar to the Saturday. This day also having so many large peak times occurred, but maximum peak times occurred afternoon again at 1:52 and 3:51 pm.

3.6.3 Weekend observations for Monday, December 18

On Monday, we have been seeing a working day for all employees and for normal people communicates to their destination places to and from work. We observed the morning rush from 7:55 to 8:25 am and scattered as large peaks. Again evening also the employees are reaching their destination, so evening rush at 5:19 pm and 6:48 pm. We even see a strong peak time fluctuations at 12 noon as people go out for lunch or perhaps.

3.6.4 Weekend observations for Tuesday, December 19

On Tuesday, observed pattern is similar to Monday. In this whole day, so many peak times occurred out of those two peaks in the morning and three peaks in the evening figured out.

3.6.5 Weekend observations for Wednesday, December 20

On Wednesday, morning from 6:57 am to 7:27 am, the traffic density is heavy. In the evening, we see two strong peak times out of many peak times at Cloud evening.

3.6.6 Weekend observations for Thursday, December 21

On Thursday, the peak timings are- two at the morning and three at evening occurred, and the heavy rush is same as like on Wednesday.

3.6.7 Weekend observations for Friday, December 22

On Friday, we observed that continuously the traffic is going like anything and we observed maximum peak times thought a day.

3.7 Online Analysis and Visualization: Creating Visualizations inside ThingSpeak

As of now, we had developed traffic monitoring algorithm using Simulink model and that has deployed onto the Raspberry Pi 3 Model B. Then we programmed for traffic monitoring algorithm and send the results to the ThingSpeak IoT Cloud platform for online analysis and visualizations. After that, we took one-week traffic patterns and performed some analysis on that and get results like when is the maximum peak times occurs and when is the low peak time occurs at daily wise. We used eastbound traffic and westbound traffic for measuring a number of vehicles are moving on both ends. It is easy to calculate the mean value for daily and weekly traffic data observations. To do this we set a threshold value for finding moderate or heavy traffic patterns based on our deep observations. This code will automatically update whenever the ThingSpeak IoT Cloud platform has viewed. So now, we can easily find the heavy traffic timings on this highway road as shown in Fig.10.

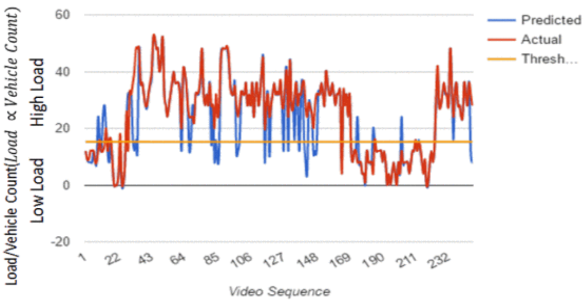


Fig.10. Online Analysis and Visualizations on ThingSpeak

3.8 Create a dynamic visualization of traffic density: Viewing the data online

Up to now, we have created a channel on ThingSpeak that delineates the crude movement thickness in both eastward rush-hour gridlock and westward activity bearings and assessments peak-times of activity on day-by-day or week after week premise. The created channel has situated on ThingSpeak. We can play out some examination and perceptions at 24*7, conduct utilizing any web program and keep running on a machine or portable. With a specific end goal to see the movement conditions on the web, go to ThingSpeak IoT Cloud stage.

4. Performance Evaluation

4.1 Simulation Setup

Presently, we exhibit the simulation results to assess the execution of the IoT-based semantic framework by actualizing semantic traffic monitoring as well as traffic reduction. For simulation purposes, we demonstrated an IoT, where smoothly moving nodes speak to the portable clients in a level, square, and encased system region. The nodes take after the Simulink versatility show. The speed and the delay time of Simulink portability demonstrate have been predefined inside a range to such an extent that it impersonates gradually strolling versatile clients taking a short break between developments. Every one of these nodes has a similar transmission range and cache measure. The Transmission Range (TR) is chosen inside the correspondence scope of Bluetooth and Wi-Fi. Even though moving inside the system, the nodes may observe events and create reports. A node can just observe an event if the event is inside its witness Range, WR. The WR is kept shorter to permit seeing of the events inside the nearness of the client. The events happen in the system following the Poisson circulation. The detailing likelihood of witnesses is Pr the procedure of report generations takes after three stages: (1) an arbitrary reports identified with that the occasion is chosen from the database. (2) area of the event is acquired by utilizing the worldwide situating framework show in the mobile phone and getting to a guide of the system territory (either on the web or disconnected, e.g., put away in the gadget), and (3) time of the witness is included based the inner clock of the witness. Note that every node has an inside time that begins in the meantime toward the start of the recreation. A report implanted with area and time data recognizes one of a kind event and reports. For example, in a simulation scenario, two unique nodes may see a similar event at two distinct events and produce reports. Presently, if these reports are gotten by a similar middle of the road node, it looks at the area of the crisis. Since the quantities of crisis in every recreation run stay settled as got from Poisson entry, the node distinguishes the as of late revealed occasion to be a copy one. Along these lines, in spite of the fact that the reports are distributed in various events, they are viewed as repetitive. Then again, if a middle node gets reports created by various nodes at a comparative time or a similar time, it will, in any case, think about the area of the events happened. Since the areas of the occasions are unique, they are thought to appear as something else. The hubs take after the DBN based steering to send these reports to the anycast goal. Presently, the time span for report accumulation, t, can be intended to gather the reports created in the system much of the time or discontinuously. Since we need to permit some asset reserve funds in our hidden system and also identify an event as fast as would be prudent, we have kept the window extensively little. In Table 1, we have featured the values utilized for a few essential parameters utilized in the simulation environment.

Table 1. Simulation parameters and its values

Parameter	**Value**
Number of Nodes (N)	200
Area of Network (A)	550 * 550 m
Speed (RWP)	0 to 2 ms
Pause Time (RWP)	0 to 1 s
Transmission Range (TR)	10 to 100 m
Size of Cache (SC)	60
Range of Witness (WR)	5 to 50 m
Event Rate (μ_E)	0.1 to 0.5
Probability for generation of reports (P_r)	0.2 to 1.0
Number of anycast destination	3
Destination Location (D2) 1	(50,150)
Destination Location (D2) 2	(250,350)
Destination Location (D2) 3	(500,500)
Frame Collection Time (t)	30 s
Time of Simulation (T)	1800 s

5.2 Performance Evaluation Metrics

To evaluate the performance of the proposed semantic interoperable traffic management framework, we have used the following metrics.

5.2.1 Latency of Detection (LoD)

This is the time passed b/w event occurrence time and event detection time. The formula for occurrence and detection of events is as follows:

$$LoD_{in} = T_{oc} - T_{in} \qquad (1)$$

$$LoD_{dt} = T_{oc} - T_{dt} \qquad (2)$$

Here LoD_{in} and LoD_{dt} are the differences of time event occurrence at intermediate nodes and event detection respectively. The notations T_{oc} ,T_{in}, and T_{dt} are the occurrence event time, intermediate node detection time, and the detection time of destination respectively. The results are generated at LoD_{in} is always lower than the results generated at LoD_{dt}. The reason behind this is to early detection of an event applying through semantics.

5.2.2 Accuracy of Detection (AoD)

To find the accuracy of detection by applying this formula and is define as

$$AoD = \frac{N_{dt}}{N_{te} - N_{te'}} * 100\% \qquad (3)$$

Here N_{dt} is the detected total events in network. N_{te} is the total events occurred in network, and $N_{te}{}'$is total events not occurred or not predicted in the network and is opposite to N_{te}. The AoD is ideally 100%, because predicted events in network is more detected compare to the intermediate nodes.

5.2.3 Cost of Reduction (CoR)

To find the Cost of Reduction by applying this formula and is define as

$$CoR = 1 - (\frac{N_{dt}}{N_{te} - N_{te'}}) * 100\% \qquad (4)$$

Here N_{dt} is the detected total events in network. N_{te} is the total events occurred in network, and $N_{te}{}'$is total events not occurred or not predicted in the network and is opposite to N_{te}. The CoR is ideally 0%, because destination has received reports about each witnessed event although the semantic traffic has been reduced.

5.2.4 Efficiency of Reduction (EoR)

To find the Efficiency of Reduction by applying this formula and is define as

$$EoR = \left(1 - \frac{\alpha}{\beta}\right) * 100\% \qquad (5)$$

Here α is the total number of events that has been received in the final destination and β is the total number of events produced in the network field. The EoR is as much as higher, indicates that removal of semantic information is high. So that, the higher traffic reduction as resulted.

5.3 Semantic Interoperable Traffic Monitoring Performance

To evaluate the semantic monitoring performance in terms of LoD, we present Figures 11–13. In Figure 11, the effect of WR on LoD has been presented for N is 80; 160, Sc is 60, TR is 100 m and WR is 50 m. An increased witness range can be observed to increase both LoD_{in} and LoD_{dt}. Besides, more nodes in the network also increases the latency in both cases. It can be understood that higher WR and increased N will increase the opportunity of more events being witnessed by more nodes. As a result, many more reports will be introduced in the network. Hence, while performing the semantic monitoring, the intermediate nodes will require more time to process the collected reports than earlier. Therefore, more time will be required to detect an event. For that reason, a rise in the LoD_{in} has been observed. To compare the performance of LoD_{in}, we have used LoD_{dt} as the benchmark performance throughout this section. As defined earlier, the latter is the event detection time when no monitoring employed in the network, i.e., destination detects the events. The increased WR, and increased N will result in congestion within the network. Therefore, it will result in route failure, and increased delay to reach the destination.

In Figure 12, we observe the similar result for different values of TR in which WRD 50 m and other parameter values remain same as previous. Both LoD_{in} and LoD_{dt} are found to increase with increase TR and N. The combined effect of higher TR and N is that more reports have been generated and collected in the network. As explained earlier, more reports require more time to be processed and thus a rise in the LoD_{in} is observed. On the other hand, numerous reports will cause congestion and result in a higher LoD. However, as expected, LoD_{in} still remains lower than the LoD_{dt} exhibiting the effectiveness of the proposed semantic monitoring. Besides, in Figure 13, we have investigated the effect of increased event in the network on LoD_{in} and LoD_{dt}. It is observed that if many events are occurring in the network, the delay to detect the events becomes higher. However, intermediate nodes have been able to detect the events before the destination.

Table 2. Accuracy of detection, AoD, at different witness ranges with case 1: TR is 50 m E is 0 to 4, Pr is 0 to 6, Sc is 40 and case-2: TR is 100 m, E is 0:5, Pr is 0 to 8, Sc is 30.

Witness Range (m)	**AoD_1 (%)**	**N_B %**
5	98.74	99.82
20	98.8	100
35	100	100
50	100	100
Witness Range (m)	**AoD_2 (%)**	**N_B %**
5	98.82	99.87
20	99.98	100
35	100	100
50	100	100

Observing the results presented in Table 2, we can comment that the number of detected events are similar as the number of witnessed events in the network in most cases. Hence, the event monitoring framework can be regarded as precise enough. In case of a lower witness range, performance of the AoD lags behind because of the fact that with a smaller witness range, the intermediate nodes do not receive enough reports. The lesser number of reports degrades the performance of the classifier by handicapping its ability to classify them accurately. As a result, the AoD degrades

By observing Figures 11, we can justify the use of semantic monitoring in an IoT-based Simulink model. As the number of nodes, their witness range, transmission range, event rate and reporting probability increase, the intermediate nodes have detected events with much lower delay than by the destination. Consequently, this will allow anyone to be informed about an emergency much earlier. In addition, since the proposed framework allows the event detection possible by any node in the network, it decreases reliance on the traditional public safety networks. However, not only a lower LoD_{in} is required to completely justify the implementation of a semantic monitoring framework, but it is also important that we investigate the AoD. Therefore, in Table 2, we present two different cases of computed AoD for different witness ranges. The benchmark, NB, is a ratio of total events witnessed in the network to the total occurred events in the network.

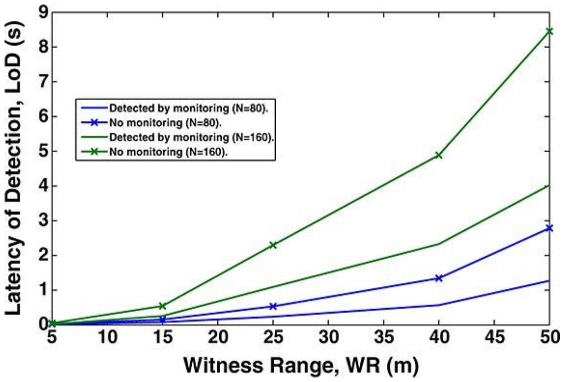


Fig.11. Impact of witness range on latency of detection

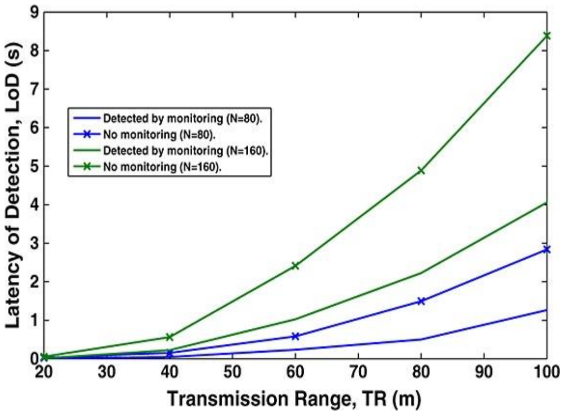


Fig.12. Impact of transmission range on latency of detection

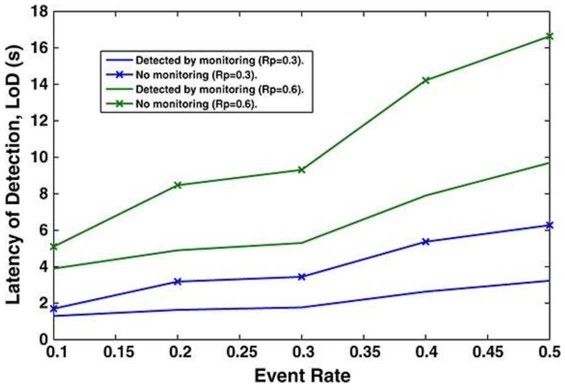


Fig.13. Impact of event rate on latency of detection

5.4 Semantic Interoperable Traffic Reduction Performance

In Table 3, we present two different cases of computed CoR for different transmission ranges and compared them with the benchmark. The benchmark values of CoR, CoR_B, are obtained with no semantic traffic reduction implemented in the network. From the results presented in Table 3, we can conclude that in most cases, the destination is aware of all the events that occurred in the network after the proposed semantic traffic reduction. Hence, the CoR is relatively low or 0 and the proposed framework can be referred as sufficiently reliable. In case of a lower transmission range, performance of the proposed scheme downgrades, because some reports may not be forwarded to the intermediate nodes because of not having a route. The other parameters for the simulation are Sc is 60, TR is 100 m, Pr is 0 to 6, and E is 0 to 5.

Table 3. Cost of Reduction, CoR, at different transmission ranges with case 1: WR is 20 m E is 0 to 4, Pr is 0 to 6, Sc is 40 and case 2: WRD 50 m, E D 0:5, Pr is 0 to 8, Sc is 30

Transmission Range (m)	**CoR_1 (%)**	**CoR_B %**
10	5	0.37
40	3	0
70	1.6	0
100	0.4	0
Transmission Range (m)	**CoR_2 (%)**	**CoR_B %**
10	3.5	0.3
40	2.4	0
70	0	0
100	0	0

Next, we present the impact of witness range on the EoR in Figure 14. The EoR is presented for N is equal to 40, 100, 160, Sc is equal to 60, TR is 100 m, Pr is 0 to 8, and μ_E is 0 to 5. As the witness range increases, the EoR becomes higher. An increased witness range will allow increased witnesses to observe the event. Because the number of reports increases, the redundancy will also be increased. With more number of nodes in the network, for any given witness range, the semantic reports will be increased. Therefore, the reduction ratio will go up. The results have been compared with the EoR in the network when no semantic reduction has been utilized. In this scenario, destination has received all reports generated in the network. Hence, EoR will be zero.

Similar impact of transmission range is found on EoR in Figure 15. For WR is 50 m and all other parameters same as previous, an increase in EoR is found with increasing transmission range. An increase in the transmission range accelerates the EoR because higher success in establishing routes can be obtained. Besides, many nodes may employ with the same intermediate node with increased transmission range and thus increase the EoR. Furthermore, an increase in the number of nodes will increase the EoR. It is because of the increased semantically similar reports in the network. In Figure 16, the EoR is presented to show the impact of the event rate. An increased event rate will result into higher number of events occurring within the network area. Therefore, it will increase the possibilities of more witnesses coming across them. Hence, many semantic reports will be generated and subjected to more reduction. Moreover, if the reporting probability is also increased, a further reduction of semantic reports is observed. The other simulation parameters in this results are the following: Sc is 60, TR is 100 m, and WR is 5 m.

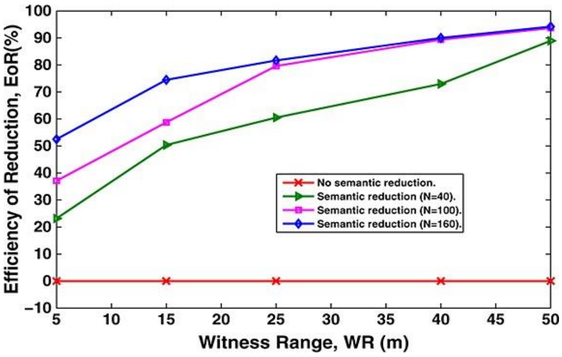


Fig.14. Impact of witness range on efficiency of reduction

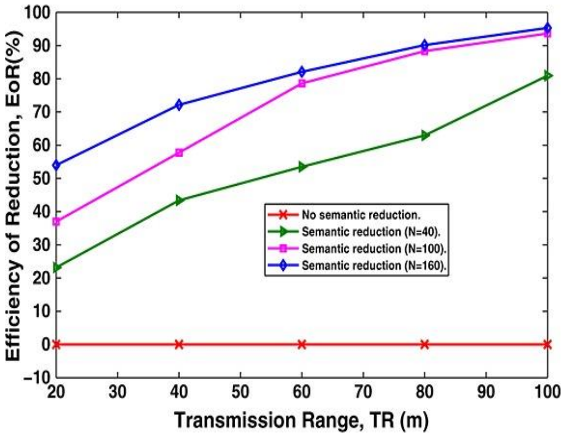


Fig.15. Impact of transmission range on efficiency of reduction

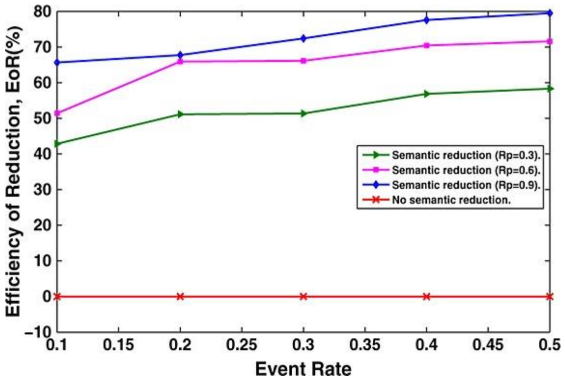


Fig.16. Impact of event rate on efficiency of reduction

The benchmark is kept same as before (no semantic reduction is occurred in the network) for comparing performances presented in Figures 15 and 16.

Finally, to summarize the results obtained in 5.3 and 5.4, it can be mentioned that the semantic traffic monitoring results sufficiently accurate and early detection of the reported emergencies. The detection time through monitoring is found lower than that of the destination under different conditions. On the other hand, it can also be mentioned that the proposed framework efficiently reduces semantic traffic in the network. However, the cost of the reduction remains low, that is, none or only a minor fraction of the total occurred events within the network is missed to be reported. Therefore, according to the performances, the proposed IoT-based semantic traffic management framework enables early detection of an event while being accurate enough and also reduces the semantic traffic such that sufficient information is kept about an event.

6. Conclusion and Future work

Analytics are everywhere exists and occurred in an Internet of Things (IoT). In this paper, authors developed a semantic smart real-time traffic management system for measuring live traffic patterns on the daily or weekly basis. The proposed framework enabled semantic traffic management through two phases. Although the first phase, Traffic Monitoring Unit, monitored the traffic semantically to allow the early detection of an emergency, the latter phase, Traffic Reduction Unit, reduced the semantic traffic related to an emergency to avoid information explosion. Simulation results supported that the accurate and early detection of an emergency are possible within the framework. Moreover, the framework enables the related traffic reduction such that the emergency can yet be reported within the framework.

This work addressed semantic traffic properties based on several simple assumptions. As a result, the proposed framework can be further improved by addressing the following issues in the future. Firstly, we assumed that the reports were publicly shared in the network. Therefore, we did not consider any security or privacy issues. In future, can develop the traffic-monitoring algorithm on providing night-vision camera for capturing high-quality images, as detecting the vehicles at night. However, in the future, we will be considering these issues to evaluate the proposed framework's performance.

References

[1] Kapileswar Nellore, Gerhard P. Hancke, "A Survey on Urban Traffic Management System Using Wireless Sensor Networks", *Sensors*, vol. 16, no. 157, 2016.

[2] Md. Rokebul Islam, Nafis Ibn Shahi, Dewan Tanzim ul Karim, Abdullah Al Mamun, Md. Khalilur Rhaman, "An Efficient Algorithm for Detecting Traffic Congestion and a Framework for Smart Traffic Control System", *IEEE conference ICACT*, pp. 802-807, 2016.

[3] Mallik Anupama "Ontology based context aware situation tracking", *Internet of Things (WF-IoT) 2015 IEEE 2nd World Forum*, 2015.

[4] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, Michele Zorzi, "Internet of things for smart cities", *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 2232, 2014.

[5] S.Bhadra, A. Kundu and S. K. Guha, "An Agent based Efficient Traffic Framework using Fuzzy", Fourth International Conference on Advanced Computing & Communication Technologies, 2014.

[6] Anupama Mallik, Hiranmay Ghosh, Santanu Chaudhury, Gaurav Harit, "MOWL: An Ontology Representation Language for Web-based Multimedia Applications", *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 10, no. 1, pp. 21, Dec. 2013.

[7] P. Pyykonen, J. Laitinen, J. Viitanen, P. Eloranta and Korhonen, "IoT for Intelligent Traffic System, IoT for intelligent traffic system", International Conference on Intelligent Computer Communication and Processing (ICCP), 2013 IEEE.

[8] European Lighthouse Integrated Project - 7th Frameworks, Internet of Things - Architecture. http://www.iot-a.eu/ 2012.

[9] K. Kotis, & A. Katasonov, "Semantic Interoperability on the Web of Things: The Smart Gateway Framework", In Proceedings of the Sixth International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2012), Palermo, 2012.

[10] http://newsroom.cisco.com/release/1308288, 2014.

[11] D. Bandyopadhyay and J. Sen, "The internet of things - applications and challenges in technology and Standardization", Springer International Journal of Wireless Personal Communications, 2011, vol. 58, no. 1, pp. 49-69.

[12] D. Singh, G. Tripathi and A. J. Jara, "A survey of Internet-of-Things: Future Vision, Architecture, Challenges and Services", IEEE World of Forum on Internet of Things, 2017.

[13] V.Katiyar, P. Kumar and N. Chand, ―An Intelligent Transportation System Architecture using Wireless Sensor Network‖, International Journal Computer Applications, 2011,vol. 14, pp. 22-26.

[14] P. Pyykonen, J. Laitinen, J. Viitanen, P. Eloranta and Korhonen, "IoT for Intelligent Traffic System, IoT for intelligent traffic system", International Conference on Intelligent Computer Communication and Processing (ICCP), 2013 IEEE.

[15] C.Yulian, L. Wenfeng and J. Zhang, "Real-Time Traffic Information Collecting and, Monitoring System Based on the Internet of Things", 6th International Conference on Pervasive Computing and Applications (ICPCA), 2011 IEEE.

[16] X. Yu, F. Sun and X. Cheng, "Intelligent Urban Traffic Management System Based on Cloud Computing and Internet of Things", International Conference on Computer Science & Service System, 2012 IEEE, pp. 216

[17] Durga Devi Sanju, A. Subramani and Vijendar Kumar Solanki "Smart City: IoT Based Prototype For Parking Monitoring & Parking Management System Commanded By Mobile App" in Second International Conference on Research in Intelligent and Computing in Engineering. 2018.

[18] Vijender Kumar Solanki, Venkatesan, M & Katiyar, S " Conceptual Model for Smart Cities for Irrigation and Highway Lamps using IoT", International Journal of Interactive Multimedia and Artificial Intelligence, (ISSN 1989-1660), 2018.

Appendix A

```
        SDate = datetime (16/12/2017’, ‘InputFormat’, ‘dd/MM/yyyy’);
        EDate = datetime (23/12/2017’, ‘InputFormat’, ‘dd/MM/yyyy’);
            DtVector = sDate: eDate;
            if (dtVector (end)! = eDate)
                DtVector = [dtVector, eDate];
            end if
AltrafficData = [];
Timestamp = [];
for dayCount = 1: length (dtVector) -1
            DateRange = [dtVector (dayCount), dtVector (daycount+1)];
            [chnl Data, t] = thingSpeakRead (chnl id, ‘DRange’ DRange);
            [AltrafficData] = [altrafficData; chnl Data]
            [Tstamp] = [tstamp: t];
end for
```

A.1. Reading one-week data

```
ETraffic = altrafficData (: 4)
WTraffic = altrafficData (: 2);
            for I = 1:6
              ETraffic (i) = [];
              WTraffic (i) = [];
            end
  Dailymeaneast = mean (reshape (eTraffic, floor (length (alltrafficData)/7), []) ;
  Dailymeanwest = mean (reshape (wTraffic, floor (length (alltrafficData)/7), []) ;
  Dates = dtVector;
   Dates (7) = [];
  Dates = datenum (dates);
    figure
            bar (dates, dailymeaneast);
        grid on
            Xlabel (‘Date’)
            Ylabel (‘Mean traffic Density per observation’)
    Title (‘East Bound Traffic Density for the week of the December 16’)
     datetick
     figure
            bar(dates, dailymeanwest);
       grid on
           Xlabel (‘Date’)
           Ylabel(‘Mean traffic Density per observation’)
    Title (‘West Bound Traffic Density for the week of the December 16’)
      datetick
```

A.2. Eastbound and Westbound Traffic

```
figureplot( timestamp, alltrafficData);
   datetick;
   |    xlabel (‘Date’);
   |    ylabel(‘Traffic Density’);
   |    grid on
   |        |  title (‘traffic Density for the week starting December 16’);
   |        |  legend(West bound Traffic’.’ East Bound traffic’);
```

A.3. West bound Traffic

```
      Clear all
StrtTime {1} = ’16-12-2017 00:00:00’; weekend day
StpTime {1} = ’16-12-2017 23:59:59’;
           |  StrtTime {2} = ‘’17-12-2017 00:00:00’;
           |  StpTime {2} = ‘’17-12-2017 23:59:59;
StrtTime {3} = ‘’18 12-2017 00:00:00’;
StpTime {3} = ‘’18 12-2017 23:59:59;
           |  StrtTime {4} = ‘’19-12-2017 00:00:00’;
           |  StpTime {4} = ‘’19-12-2017 23:59:59;
StrtTime {5} = ‘’20-12-2017 00:00:00’;
StpTime {5} = ‘’20-12-2017 23:59:59;
           |  StrtTime {6} = ‘’21-12-2017 00:00:00’;
           |  StpTime {6} = ‘’21-12-2017 23:59:59;
 StrtTime {7} = ‘’22-12-2017 00:00:00’;
 StpTime {7} = ‘’22-12-2017 23:59:59’;
```

A.4. Traffic density for individual days

```
for kk = 1:7
|
|       SDate = dttime (strtTime {kk}, ‘InputFormat’, dd/MM/yyyy HH:mm:ss’);
|       EDate = dttime (stpTime {kk}, ‘InputFormat’, dd/MM/yyyy HH:mm:ss’);
|           DtVector = [sDate], eDate];
|          [Daily, t] = thingSpeakRead (Chnl Id, ‘DtRange’, dtVector);
|
|           |       DailyEast = Daily (:,2);
|           |            Tstamp = dttime (t, ‘ConvertFrom’, ‘datenum’);
|           |             DtAnalyzed = strtTime {kk}
|           DtAnalyzed = {dtAnalyzed (1: end-8))}
end
```

A.5. Fetch individual day in a loop