Machine Learning based Improved Gaussian Mixture Model for IoT Real-Time Data Analysis

Aprendizaje automático basado en mezcla Gaussiana mejorada Modelo para IoT en tiempo real. Análisis de los datos

Aprendizado baseado em máquina na mistura gaussiana melhorada Modelo de IoT em tempo real. Análise de dados

Sivadi Balakrishna[1]
Moorthy Thirumaran[2]
Vijender Kumar Solanki[3]

Received: August 20th, 2019
Accepted: November 24th, 2019
Available: January 31th, 2020

How to cite this article:
S. Balakrishna, M. Thirumaran, V. Kumar Solanki, "Machine Learning based Improved Gaussian Mixture Model for IoT Real-Time Data Analysis," *Revista Ingeniería Solidaria*, vol. 16, no. 1, 2020.
doi: https://doi.org/10.16925/2357-6014.2020.01.02

Artículo de investigación. https://doi.org/10.16925/2357-6014.2020.01.02

[1] Department of Computer Science and Engineering, Pondicherry Engineering College, Pondicherry, India.
ORCID: https://orcid.org/0000-0002-8939-9307

[2] Department of Computer Science and Engineering, Pondicherry Engineering College, Pondicherry, India.
ORCID: https://orcid.org/0000-0002-2210-1553

[3] Department of Computer Science and Engineering, CMR Institute of Technology, Hyderabad, India.
ORCID: https://orcid.org/0000-0001-5784-1052
Email: vijendersolanki@cmritonline.ac.in; spesinfo@yahoo.com

Abstract

Introduction: The article is the product of the research "Due to the increase in popularity of Internet of Things (IoT), a huge amount of sensor data is being generated from various smart city applications", developed at Pondicherry University in the year 2019.

Problem: To acquire and analyze the huge amount of sensor-generated data effectively is a significant problem when processing the data.

Objective: To propose a novel framework for IoT sensor data analysis using machine learning based improved Gaussian Mixture Model (GMM) by acquired real-time data.

*Methodology***:** In this paper, the clustering based GMM models are used to find the density patterns on a daily or weekly basis for user requirements. The ThingSpeak cloud platform used for performing analysis and visualizations.

Results: An analysis has been performed on the proposed mechanism implemented on real-time traffic data with Accuracy, Precision, Recall, and F-Score as measures.

Conclusions: The results indicate that the proposed mechanism is efficient when compared with the state-of-the-art schemes.

Originality: Applying GMM and ThingSpeak Cloud platform to perform analysis on IoT real-time data is the first approach to find traffic density patterns on busy roads.

Restrictions: There is a need to develop the application for mobile users to find the optimal traffic routes based on density patterns. The authors could not concentrate on the security aspect for finding density patterns.

Keywords: IoT, data analysis, machine learning, GMM, ThingSpeak.

Resumen

Introducción: el artículo es producto de la investigación "Debido al aumento en la popularidad de Internet de las cosas (IoT), se está generando una gran cantidad de datos de sensores a partir de varias aplicaciones de ciudades inteligentes ", desarrollado en la Universidad de Pondicherry en el año 2019.

Problema: adquirir y analizar datos generados por sensores de manera efectiva pues es un problema importante al procesar los datos.

Objetivo: proponer un marco novedoso para el análisis de datos del sensor IoT utilizando el aprendizaje automático basado en mejoras desde el Modelo de mezcla gaussiana (GMM) por datos adquiridos en tiempo real.

Metodología: en este documento, los modelos GMM basados en agrupamiento se utilizan para encontrar los patrones de densidad en un día o semanalmente para los requisitos del usuario. La plataforma en la nube ThingSpeak utilizada para realizar análisis y visualizaciones.

Resultados: se realizó un análisis sobre el mecanismo propuesto implementado en datos de tráfico en tiempo real con precisión, recuperación y F-Score como medidas.

Conclusiones: los resultados indican que el mecanismo propuesto es eficiente en comparación con el estado de esquemas de arte.

Originalidad: la aplicación de la plataforma GMM y ThingSpeak Cloud para realizar análisis de datos en tiempo real de IoT es el primer enfoque para encontrar patrones de densidad de tráfico en carreteras transitadas.

Restricciones: Existe la necesidad de desarrollar la aplicación para que los usuarios móviles encuentren las rutas de tráfico óptimas basadas en patrones de densidad. Los autores no pudieron desarrollar el aspecto de seguridad para encontrar patrones de densidad.

Palabras clave: IoT, análisis de datos, aprendizaje automático, GMM, ThingSpeak.

Resumo

Introdução: o artigo é um produto de pesquisa "Devido ao aumento da popularidade da Internet das coisas (IoT), uma grande quantidade de dados do sensor está sendo gerada a partir de várias aplicações de cidades inteligentes ", desenvolvidas no Universidade de Pondicherry no ano de 2019.

Problema: Adquira e analise os dados gerados pelos sensores de forma eficaz, pois é um problema importante Ao processar os dados.

Objetivo: propor uma nova estrutura para a análise dos dados dos sensores da IoT usando o aprendizado de máquina com base nas melhorias do Modelo de mistura Gaussian (GMM) para dados adquiridos em tempo real.

Metodologia: neste documento, modelos baseados em agrupamentos GMM são usados para encontrar padrões de densidade em um dia ou semanalmente para os requisitos do usuário. A plataforma em nuvem ThingSpeak usada para executar análises e visualizações.

Resultados: foi realizada uma análise do mecanismo proposto implementado em dados de tráfego em tempo real com precisão, recuperação e F-Score como medidas.

Conclusões: os resultados indicam que o mecanismo proposto é eficiente em comparação com o estado de Esquemas de arte

Originalidade: a aplicação da plataforma GMM e ThingSpeak Cloud para realizar análise de dados em tempo real da IoT é a primeira abordagem para encontrar padrões de densidade de tráfego em estradas movimentadas.

Restrições: É necessário desenvolver o aplicativo para usuários móveis para encontrar as rotas de tráfego ideais com base em padrões de densidade. Os autores não conseguiram desenvolver o aspecto de segurança para encontrar padrões de densidade.

Palavras-chave: IoT, análise de dados, aprendizado de máquina, GMM, ThingSpeak.

1. Introduction

In recent development, around 40 billion Internet of Things (IoT) sensor devices are connected to the Internet. By 2025, it is predicted that the number will grow between 80 and 120 billion devices connected to the internet, generating 180 trillion gigabytes of new sensor data that year. In this IoT and Cyber-Physical-Systems (CPS) digital era, a massive amount of sensor data has been generated. In order to perform data analysis and integration of IoT sensor data, the data analysis algorithms play a major role. To process and analyze sensor data is a challenging task. In earlier days, so many traditional algorithms were proposed and implemented for IoT data analysis. The traditional data analysis algorithms only dealt with static and spherical data. However, dealing with huge volumes and variety of dynamic data originating from IoT sensor devices is a challenging problem. In this paper, we used the machine learning based improved GMM with ROI parameter values for estimating the traffic density on busy roads while considering a smart traffic management application.

In recent years, the urban population growth is increasing tremendously. The global urban population is expected to grow 1.86% per year from 2015 to 2020. In

between 2020 to 2025, this increase will be 1.63% and from 2025 to 2030 it may rise again by approximately 1.44% according to the World Health Organization (WHO). Cars are popular in an urban population with many families owning a minimum of two cars. The popularity of private cars increases urban traffic and because of this, traffic is becoming one of the biggest problems in urban areas all over the world [1]. The heavy traffic in urban cities is leading to congestion, accidents that have caused the loss of property, wasted time, environmental pollution, and sometimes even the death of a person. Therefore, there is a big need for traffic monitoring and reduction systems in urban areas for them to become Smart cities [2]. The best approach to solve this problem is using the Internet of Things and it provides a new trend to intelligent traffic management. To improve the efficiency in data analysis, this paper intends to use the IoT Technology, Machine Learning, Cloud computing, Raspberry Pi and data analysis technologies. The actual scenario of this research is as follows: The information generated by IoT device data, collected from the roads through gateways, can be presented to all travelers and users. After collecting the real-time sensor data, the system can recognize the current traffic, traffic flow conditions and can predict future traffic in urban areas. After that, sensor data monitoring and sensor data detection have been measured for analyzing and visualization of the acquired data. Based on the system-generated data, it may useful to drivers when choosing optimal routes. Therefore, the system is dynamically administrative, controlling and monitoring moving cars. If intelligent traffic systems using IoT can be made, then there are a lot of potential benefits for users such as improving traffic conditions, fewer traffic jams and high reliability, traffic safety, lower management costs and independence of weather conditions. Bridges, roads, tunnels, vehicles, traffic signals and drivers are some of the traffic elements in IoT. All these items will be connected to the web for monitoring and identification through different types of IoT devices like RFID, sensors, actuators, Global Positioning Systems (GPS) and laser scanners. As of late within the private sector, vehicles are swarming urban activity ever more. As a result, analysis is getting to be noticeably one of the imperative issues in a smart city framework everywhere throughout the world. Some of these concerns are movement block and mischances that normally cause a critical exercise in futility, property harm, and environmental contamination. Any kind of congestion on the streets eventually 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 density [3].

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 be 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 the most recent continuous moving data that helps drivers pick ideal paths. Thusly, the framework can absolutely administrate, monitor and control moving vehicles.

Developing a smart framework in view of IoT has a number of advantages such as a change of activity conditions, lessening the car influx and administration costs, high unwavering quality, traffic security and freedom of climate conditions. Such undertaking of an IoT must incorporate each component of traffic such as streets, spans, burrows, traffic signs, vehicles, and drivers. 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. 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 in accessing the logistics and services available on the road. Therefore, the developed system will be more reliable and accurate towards traffic management. The IoT sensor 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 captured within urban cities. These types of model executions can be found in the Smart Santander EU venture.

These are the following contributions made on this proposed approach:

1. To propose machine learning based improved GMM with ROI for IoT sensor real-time data analysis.
2. To analyze the gathered sensor data on providing some relevant feature extractions to show the daily and weekly wise sensor density conditions.
3. To find the Accuracy, Precision, Recall, and F-score of the proposed mechanism.
4. To compare the obtained results with the existing most relevant state-of-the-art schemes in an efficient way.

The rest of this paper is organized as follows: Section 2 shows the related work of this study. The proposed machine learning based improved GMM for IoT sensor data analysis mechanism has been described in Section 3. Section 4 shows the

performance evaluation of proposed work and it includes the experimental results and analysis. Finally, Section 5 concludes this paper and suggests future enhancements.

2. Related Work

IoT sensor data acquisition and analysis are one of the biggest problems in this present living world. The majority of the researchers have dealt with and put their efforts into this problem. As a result, several types of approaches have been developed. Bhadra et al [5] have applied agent-based fuzzy logic technology involving multiple approaches and vehicle movements for traffic control situations. They used a fuzzy neural network, and observed a traffic prediction mechanism in chaotic traffic flow 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 enterprise services for effective integration of Service Oriented Architecture (SOA) and Internet of Things (IoT). Due to the revolution in IoT, many of the researchers shifted their attention. Zhang et.al, proposed the more needful and convenient environment for sensor data collection and analysis in smart city applications. 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. [10] proposed various applications applied in different domains like smart city, smart home, smart metering, smart transport and healthcare.

D. Singh et al. [11] discussed the existing status of the smart city transport management system and researchers have been working more on real-time transport systems using the Internet of Things (IoT). X. Yu 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 taken from streets by a camcorder. Activity thickness is isolated, which considers total range controlled by vehicles all over the place, similar to the total of pixels in a video layout, instead of figuring out a number of vehicles. Two parameters are set 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 sweeping in addition.

Therefore, M. Mazhar Rathore et al. [14] suggests that the opportunity has already come and gone for limited management of 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 is an effective procedure for sharp activity

e-ISSN 2357-6014 / Vol. 16, no. 1 / 2020 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

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 by A.P. Plageras et al. [15], which can be 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 can be recognized early and thus preventive measures can be taken, in this way saving time and cost to the driver. Q. Liu Qiang Liu a et al. [16] presented an approach called dynamic movement observing framework. It implies that different influencing factors should be investigated. Sharmad Pasha [17] developed a GPS based vehicle ensuing framework. It diminishes the short separation travel and in addition to the same substance is concentrated on different variables so as to be thinking about convenient information submitting by utilizing VSN's 'vector remove directing calculation' and gives a high dependable correspondence. It covers the range of extreme separation when gathering information. Akbar Adnan et al. [18] made an RFID based conceptual model on insight into motion control frameworks. It controls movement stream, decreasing mischances in rush hour gridlock spots and remote area transmission. Therefore, after careful observation of all these literature surveys, there is a need for analysis of IoT sensor data in an efficient manner.

3. Machine Learning based improved Gaussian Mixture Model for IoT Data Analysis

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

Here we have shown how to monitor the real-time traffic data on the busy highway and the real-time data collected on the cloud. Once stored, analytics can be performed on the aggregated data on the cloud for the edge device and finally an analysis can be performed on the cloud data. Fig.1 illustrates a framework of workflow analysis and performing online analysis on the stored data in repositories. This framework uses ThingSpeak for both storage and analysis.

e-ISSN 2357-6014 / Vol. 16, no. 1 / 2020 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

The proposed framework based on machine learning techniques may consist of four layers as shown in Fig.1 These are the Sensing layer, Data processing layer, IoT sensor data analysis layer, and Application layer.

1. **Sensing 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 acquired from heterogeneous IoT devices: sensors, actuators, protocols, and data formats, 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 collected raw data. The data processing layer mainly consists of two sub-modules, namely Feature extraction, and Foreground Detector 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 may be simple or sometimes may be complex. Therefore, based on the representation of sensor data, both feature extraction and foreground detector data are transfered to the next layer i.e. IoT sensor data analysis layer for performing meaningful data as well as interpretation by providing Simulink and data integration approaches.

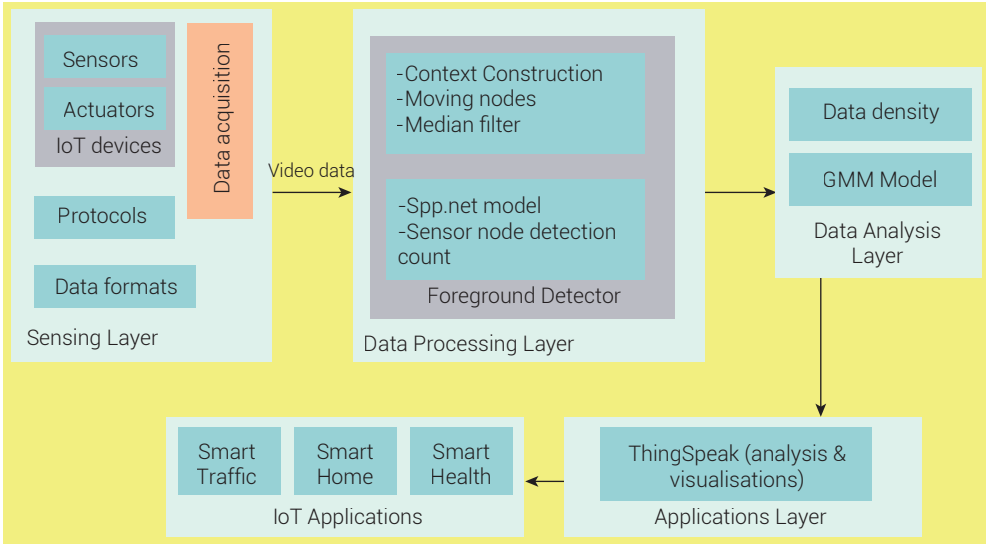

Figure1. The proposed framework for IoT sensor data analysis
Source: own work

e-ISSN 2357-6014 / Vol. 16, no. 1 / 2020 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

3. **Data analysis layer:** This is the crucial layer for this proposed framework. The Simulink model has been applied to this layer. It will take the data after pre-processing the data processing layer to annotate the IoT sensor data. The ambiguity of the foreground detector data is minimized by applying the Simulink model. The ThingSpeak based domain knowledge systems have used the context-aware data to predict the real-time traffic data for reducing the traffic jams and unforeseen delays.
4. **Application layer:** This layer is used to perform analysis and visualization of the acquired IoT sensor data to the users using the ThingSpeak cloud platform. This layer generates traffic alerts for vehicles in real-time situations. These analyzed and visualized data are applicable to IoT smart city applications of smart traffic, smart home, smart grid, and smart health etc.

3.1 Gaussian Mixture Model work process

The clustering based GMM techniques are used for finding the traffic density patterns using Foreground Detector and Median Filter approaches. Using Foreground Detector, the pixel location points to the value and identifies the frames. The arbitrary pixel point (a,b) consist of vector sequence and its value can be expressed as follows

$$\{a/, \dots \dots, a_n\} = \{g_t\,(a, b): 1 \leq t \leq n\} \tag{1}$$

Here $g_t\ (a,\ b)$ stands for the pixel grey scale value at time t. The GMM uses Gaussian distribution method to find that arbitrary value. Hence the pixel proportion g_c current pixel value:

$$k(a, b) = \sum_{i=1}^{p} q_i\,, c * m(a_c, u_n, \sum n, c) \tag{2}$$

Where at time t the value from i is the Gaussian distribution weight, it reflects the visibility Gaussian distribution proportion. The p denotes the number of distribution pixels at a. $m\ (a_c, u_n, \sum n, c)$ is the density function from Gaussian proportion at mean value of $u\ (i,\ c)$ at time t and the covariance $\sum i,\ c$. The function value $k\ (a_c)$ is the proportion of required time at pixel a.

The GMM process for the proposed work has been carried out in Fig.2. It clearly indicates how the video has been imported and how to find the traffic density-using threshold function. The flow of GMM starts with importing video streams from any

e-ISSN 2357-6014 / Vol. 16, no. 1 / 2020 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

dataset or real time data. The video input may contain the various resolutions like 180 pixels, 240 pixels, 360 pixels, 720 pixels, 960 pixels, and 1080pixels. Thereafter, the input video is extracted into different frames as per length of the video. These frames have been processed by model. The Foreground Detector is used to find the vehicle's status by observing whether it is moving, stopping, stationary, and via dynamic positions. Moreover, using a threshold function, the video frame has been converted into a binary image. The collection of pixels is called as objects. For detection of background objects a binary digit of 0 is used and for detection of foreground objects a binary digit of 1 is used. Once the foreground detection process is completed, the Median filter should take care of distinguishing detected objects and non-detected objects by applying filters. If the object is non-detected then it goes to the video frame level and proceeds with foreground detection. If the object is detected successfully, then the object is captured and the number of stopped objects, which was obtained by subtracting moving objects from total objects, can be found. Finally, the stopped objects are compared with a pre-defined threshold value and the traffic density, either low or high, can be found.

3.1.1 Object Detection with Region of Interest (ROI)

In our earlier work [20], we have focused on accuracy detection and presented an approach to address sensor data analysis. In this paper, we extend our initial approach to make it generic for IoT sensor data analysis with improved ROI parameters, finally extending our experimental evaluation and implementation with open source components, which were optimized for large-scale IoT real-time data. Fig.3 depicts the improved GMM with ROI parameter values for optimal counting of objects. The input video frame is taken from a dataset or real-time data after initializing the GMM model with current data. Thereafter the ROI adjustment parameter values are considered for optimal counting of objects, based on acceleration testing results.

e-ISSN 2357-6014 / Vol. 16, no. 1 / 2020 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

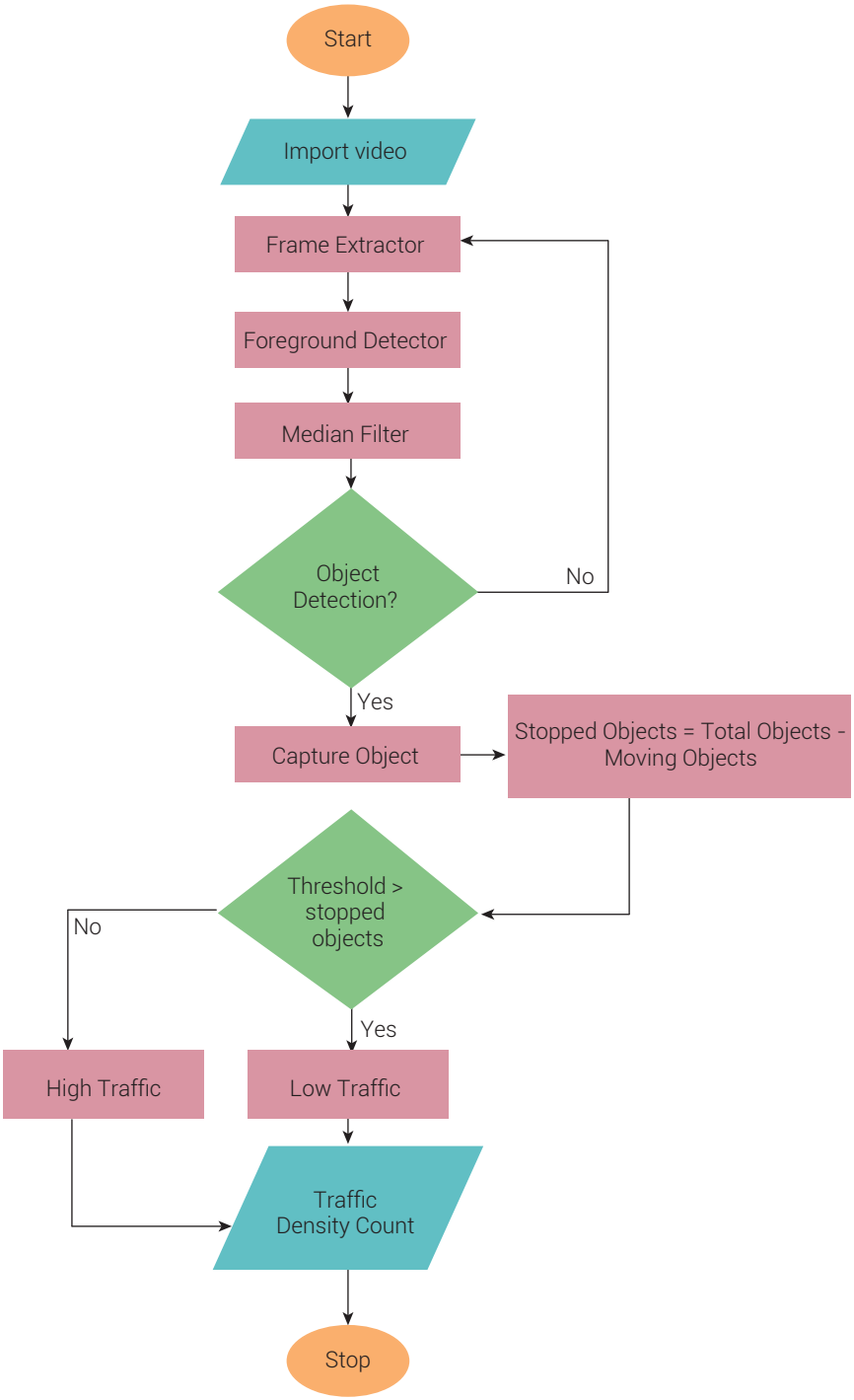

Figure 2. GMM Process Flow Chart
Source: own work

e-ISSN 2357-6014 / Vol. 16, no. 1 / 2020 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

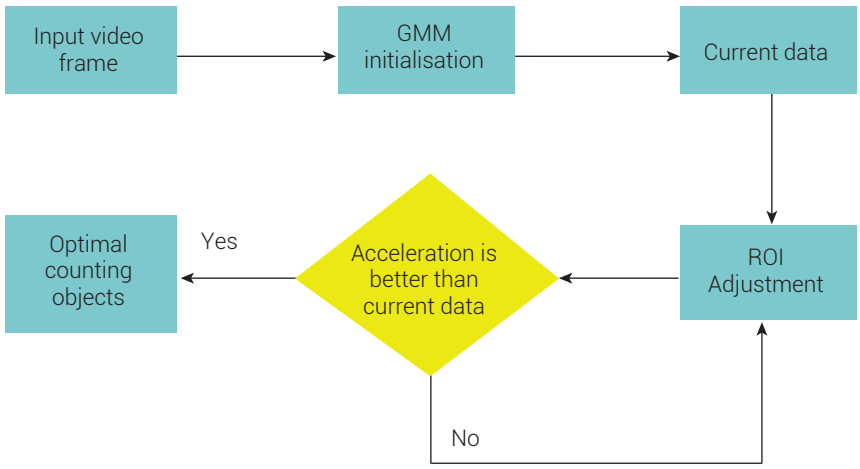

Figure.3. Traffic Density with ROI
Source: own work

3.1.2 Adjusting ROI Parameters

The adjusting ROI parameter values are used mainly for three locations; front face, middle face, and back face. The following pseudocode is used for processing the ROI adjustments in three locations.

 # initialize ROI for processing
Select the Shape1 by square border and int16 as the pixel value.
Select the Shape2 by rectangular border and int32 as the pixel value
Choose Shape3 by Opacity =1
setup ROI ***front*** *using step shape3*
setup ROI ***middle*** *using step shape3*
setup ROI ***back*** *using step shape3*
Finally display ROI = step shape1 or shape2

The list of values is chosen for different locations: Front face [20 85 115 40]; Middle face [54 25 58 17]; and Back face [60 5 30 15]. Additionally, the default parameter values are tested for the frame selections. Moreover, before ROI adjustment the Minimum (Min) and Maximum (Max) values are assumed for two kinds of object. For two-wheeler vehicle detections, the 'Min' value is 15 and the 'Max' value is 25. For four-wheeler vehicle detections, the 'Min' value is 150 and the 'Max' value is 1800. Similarly, for front face detection of vehicles, in the case of two-wheeler vehicles, the

'Min' value is set as 10 and the 'Max' value is set as 50. For front face detection of vehicles, in the case of four-wheeler vehicles, the 'Min' value is set as 100 and the 'Max' value is set as 2550. While both for back and middle faced two-wheeler detections, the 'Min' value is set as 5 and the 'Max' value is set as 10. In the case of four-wheeler detected vehicles, the 'Min' value is set as 15 and the 'Max' value is set as 150.

Algorithm 1 depicts the usage of the GMM model based on the sensor data analysis algorithm as shown below.

Algorithm 1: IoT Sensor Data Analysis

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
    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
```

3.1.3 Estimating Traffic Density

The estimation of traffic density is based on GMM (Gaussian Mixture Model) for finding situation tracking in real-time. Algorithm 1 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, the GMM and SPP-net models must be used.
2. **Calculate the stopped vehicles and moving vehicles from total count of vehicles**: In order to calculate 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 that of a stopped vehicle $C^t_{sv,}$ otherwise it is counted as a moving vehicle C^t_{mv}. The SPP-net model used for finding the count of moving vehicles is: $C^t_{sv} = C^t_{tv} - C^t_{mv}$, here t ranges between 1 less than or equal to n.

The proposed algorithm is incorporated into a GMM model with adjusted ROI parameter values and then exports the results for analysis and visualizations. It supports simulation, system-level design, automatic code generation, testing and visualization of IoT data. In this experiment, a model using Simulink and that runs on the Raspberry Pi 3 model B was developed. In this GMM model, an external capability of Simulink is used to develop the algorithm. In external mode, Simulink gathers the real-time sensor streaming data from Raspberry Pi and the users can see the video on desktop/mobile using the SDL video display while the GMM model is running.

The USB cam was placed at a particular region of highway where the vehicles are moving from left to right. The webcam connected to one USB port of the Raspberry Pi, captures video with a selected region. Next, the authors 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, the image was 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 in both the eastbound and westbound 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 is sent to ThingSpeak, the data aggregator for analyzing & visualizations. Here individually vehicle count values were sent to each eastbound and westbound region. Both east-bound traffic and westbound traffic values were sent to ThingSpeak (Channel Id) and the westbound values were stored in field1 whilst the eastbound values were stored in field2.

e-ISSN 2357-6014 / Vol. 16, no. 1 / 2020 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

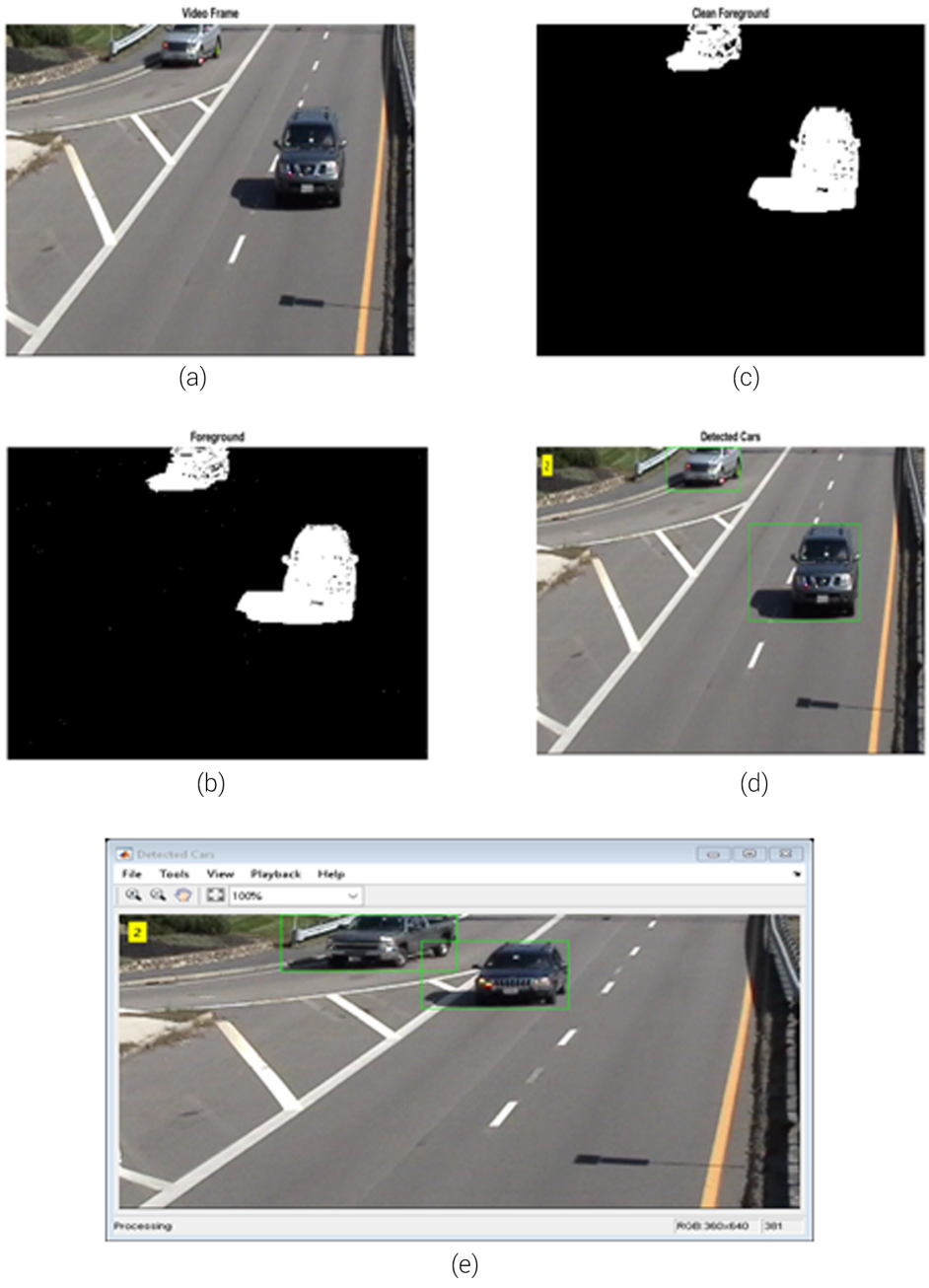

Figure. 4. Estimating total vehicle count for sensor data analysis (a) Video Frame (b) Foreground (c) Clean Foreground (d) Detected Objects (e) Final Detected Objects
Source: own work

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 assesses peak-times of activity on a day-by-day or weekly premise. The

created channel is situated on ThingSpeak. We can play out some examinations and perceptions, 24*7, conducted utilizing any web program running on a machine or laptop. With a specific end goal to see the traffic conditions on the web, go to ThingSpeak IoT Cloud platform.

3.2 Analyzing Data on ThingSpeak

The traffic-monitoring model is done by using Simulink deployed onto a Raspberry Pi hardware device. Then an analysis on real-time disseminate data, stored in the ThingSpeak cloud aggregator and sent by the Raspberry Pi 3 Model B fetching from ThingSpeak, can be started.

3.3 Online Analysis and Visualization: Creating Visualizations inside ThingSpeak

As of now, we had developed a traffic monitoring algorithm using a Simulink model deployed onto the Raspberry Pi 3 Model B. The traffic monitoring algorithm has been programmed to send the results to the ThingSpeak IoT Cloud platform for online analysis and visualizations. After that, one-week of traffic patterns were taken and analyzed. With this information, results were obtained such as; when the maximum peak times occurs and when the low peak time occurs on a daily base. We used east-bound traffic and westbound traffic for measuring the number of vehicles 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. The patterns will automatically update whenever the ThingSpeak IoT Cloud platform has viewed. The results are updated for every 15 seconds over the previous 48 hours. This allows for easy identification of the heavy traffic timings on this highway road as shown in Fig.4. Fig.5 show the 'week over week' traffic results comparison starting from 5th July 2019 to 10th July 2019. The traffic density conditions are very different from one day to another day and one week to another week. While observing the density patterns, the east bound and west bound places contain more traffic in the evening and morning. In the morning more people will come out of their home to go to school, college etc., and in the evening, they should return back to their homes. Moreover, in the day from after 10 AM to before 4 PM the traffic density is somewhat less when compared to early morning and evening timings.

e-ISSN 2357-6014 / Vol. 16, no. 1 / 2020 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

Figure 5. Traffic Density every 15 seconds for last 48 hours (a). On July 5 (b). On July 6 (c). On July 7 (d). On July 8 (e). On July 9 (f). On July 10

Source: own work

e-ISSN 2357-6014 / Vol. 16, no. 1 / 2020 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

Figure 6. Smoothed Traffic Density by week over week comparison (a). On July 5 (b). On July 6 (c). On July 7 (d). On July 8 (e). On July 9 (f). On July 10

Source: own work

e-ISSN 2357-6014 / Vol. 16, no. 1 / 2020 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

4. Performance Evaluation

The proposed framework has been implemented on ThingSpeak IoT cloud platform. This was programmed using MATLAB code with a DELL laptop of I3 Intel Pentium Processor, 4 GB RAM, and 1 TB HDD on the Windows 10 platform.

4.1 Performance Measures

To evaluate the performance of the proposed work, the following metrics are considered for measuring the proposed work. These metrics are generated from the confusion matrix as shown in Fig 7.

	Predicted as "YES"	Predicted as "NO"
Actually as "YES"	True Positive $[V_X \rightarrow V_X]$	False Negative $[V_X \rightarrow NV_X]$
Actually as "NO"	False Positive $[NV_X \rightarrow V_X]$	True Negative $[NV_X \rightarrow NV_X]$

Figure 7. Confusion Matrix
Source: own work

4.1.1. True positive

$V_X \rightarrow V_X$: This is an estimation of detected vehicles considered correctly as detected vehicles.

4.1.2. True negative

$NV_X \rightarrow NV_X$: This is an estimation of non-detected vehicles considered correctly as non-detected vehicles.

4.1.3. False positive

$NV_X \rightarrow V_X$: This is an estimation of non-detected vehicles considered incorrectly as detected vehicles.

4.1.4. False negative

$V_X \rightarrow NV_X$: This is an estimation of detected vehicles considered incorrectly as non-detected vehicles.

4.1.5. True Positive Rate (TPR)

TPR expresses the sensitivity and scales the proportion of correctly classified detected vehicles from the video as shown in Eq.3

$$TPR = \frac{V_x \rightarrow V_x}{[V_x \rightarrow V_x + V_x \rightarrow NV_x]} \tag{3}$$

4.1.6. True Negative Rate (TNR)

TNR expresses the specificity and scales the proportion of correctly classified non-detected vehicles from the video as shown Eq.4.

$$TNR = \frac{NV_x \rightarrow NV_x}{[NV_x \rightarrow NV_x + NV_x \rightarrow V_x]} \tag{4}$$

4.1.7. False Positive Rate (FPR)

FPR measures the significance level, which scales the proportion of non-detected vehicles that are interpreted as detected vehicles in the data acquisition process in an input video sequence as shown Eq.5

$$FPR = \frac{NV_x \rightarrow V_x}{[NV_x \rightarrow V_x + NV_x \rightarrow NV_x]} \tag{5}$$

4.1.8. False Negative Rate (FNR)

FNR scales the proportion of detected vehicles that are interpreted as non- detected vehicles in the data acquisition process as shown Eq.6

$$FNR = \frac{V_x \rightarrow NV_x}{[V_x \rightarrow NV_x + V_x \rightarrow V_x]} \tag{6}$$

4.1.9. Accuracy

Accuracy is the first step towards performance measurement where it defines the ratio between the total count of correctly detected vehicles made to a total count of detected vehicles made as shown Eq.7.

e-ISSN 2357-6014 / Vol. 16, no. 1 / 2020 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

$$Accuracy = \frac{(V_x \rightarrow V_x + NV_x \rightarrow NV_x)}{[V_x \rightarrow V_x + NV_x \rightarrow NV_x + NV_x \rightarrow V_x + V_x \rightarrow NV_x]} \quad (7)$$

4.1.10. Precision, Recall & F-measure

Precision discourses about the exactness of the data, and the Recall voices about completeness. Both precision and recall conclude over the accuracy of the system whereas the accuracy does not explain much about the false results. F-measure studies precision and recall to decide upon the score. Its harmonic mean over precision and recall are as shown in Eq.8-10

$$Precision = \frac{V_x \rightarrow V_x}{[V_x \rightarrow V_x + NV_x \rightarrow V_x]} \quad (8)$$

$$Recall = \frac{V_x \rightarrow NV_x}{[V_x \rightarrow V_x + V_x \rightarrow NV_x]} \quad (9)$$

$$F - measure = 2 * \left(\frac{Precision * Recall}{Precision + Recall}\right) \quad (10)$$

4.2 Result Discussions

The proposed framework has been compared with existing leading approaches: Zhang et.al, 2013 [8], M. Mazhar Rathore et.al, 2016 [14], A.P. Plageras et.al, 2017 [15], Sharmad Pasha, 2016 [17], A.R. Al-Ali, et. al, 2017 [18], László Lengyel et.al, 2015 [19], and Sivadi Balakrishna et.al, 2019 [20] by considering all the mentioned evaluation metrics in our proposed work.

Table 1 highlights the predominant Accuracy, Precision, Recall, F-Score, TNR, FPR, and FNR value over state-of-the-art schemes under our proposed MLIGMM_IoTDA mechanism. The result confirms that MLIGMM_IoTDA is able to maintain its accuracy percentage of 94.8 at varying accuracy results of data acquisition even when the false positive rate is increased. MLIGMM_IoTDA enables a superior accuracy value of 10% when compared to the Balakrishna et.al, FSDAA approach. The results prove that MLIGMM_IoTDA is capable of maintaining the average accuracy value of 10 % to 33% compared with the state-of-the-art schemes. Likewise, precision value achieves the superior 11%, recall value achieves the superior percentage of 4%, F-measure value achieves the superior 5%, TNR value achieves the superior 11%, FPR value achieves

the superior 2%, and FNR value achieves the superior 3% when compared to the state-of-the-art schemes mentioned in Table 1. The results shown in Table 1 of our proposed MLIGMM_IoTDA is considered at 10 objects.

In the first experimental investigation of our proposed mechanism with dataset of video frames, Table 2 shows the detection of traffic in either low traffic or high traffic of ten video frames under various metrics. By considering low traffic data video frames, the average accuracy achieves 95.8%, average precision achieves 100%, average TPR achieves 92%, and average TNR achieves 100%. The success rate is more because the input video data contains the low or medium number of vehicles observed. Therefore, the low traffic with highest success rates are achieved. While considering high traffic video frames, the average accuracy achieves 94.2%, average precision achieves 94.8%, average TPR achieves 93.6%, and average TNR achieves 94.8%. Finally, the results indicate that the proposed MLIGMM_IoTDA mechanism is good for detecting the vehicles on busy highway roads.

Table 1. Comparison of the various state of the art schemes with proposed MLIGMM_IoTDA

Approach	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	TNR (%)	FPR (%)	FNR (%)
Zhang et.al, 2013 [8]	61.28	64.36	59.45	57.25	65.04	27.12	32.15
M. Mazhar Rathore et.al, 2016 [14]	70.64	74.08	80.58	78.06	74.15	24.35	27.32
A.P. Plageras et.al, 2017 [15]	74.29	76.28	78.26	74.12	67.10	22.61	25.67
Sharmad Pasha, 2016 [17]	68.16	70.15	73.64	58.26	62.35	34.15	32.15
Akbar Adnan, et. al, 2018 [18]	80.14	79.05	86.01	82.14	81.39	21.68	29.31
László Lengyel et.al, 2015 [19]	76.20	79.35	81.29	80.65	74.36	23.16	26.74
Balakrishna et.al, FSDAA 2019 [20]	84.56	86.21	88.60	84.91	86.34	15.42	18.22
MLIGMM_IoTDA (Proposed)	94.8	97.6	92.7	89.64	97.5	13.16	15.20

Source: own work

e-ISSN 2357-6014 / Vol. 16, no. 1 / 2020 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

Table 2. Detected object result on both low and high traffic conditions

S.No	Video Name	Traffic Status	Real Object	Detected Object				Accuracy (%)	Precision (%)	TPR (%)	TNR (%)
				TP	TN	FP	FN				
1	v1.mov	low traffic	5	5	5	0	0	100	100	100	100
2	v2.mov	low traffic	7	6	7	0	1	92.8	100	85.7	100
3	v3.mov	low traffic	3	3	3	0	0	100	100	100	100
4	v4.mov	low traffic	4	3	4	0	1	87.5	100	75	100
5	v5.mov	low traffic	6	6	5	0	0	100	100	100	100
Average								**95.8**	**100**	**92.1**	**100**
6	v6.mov	high traffic	12	9	8	0	1	94.4	100	90.0	100
7	v7.mov	high traffic	16	13	10	0	0	100	100	100	100
8	v8.mov	high traffic	13	10	11	1	2	87.5	90.9	83.3	91.6
9	v9.mov	high traffic	17	14	12	1	0	96.2	93.3	100	92.3
10	v10.mov	high traffic	21	18	19	2	1	92.5	90.0	94.7	90.4
Average								**94.2**	**94.8**	**93.6**	**94.8**

Source: own work

In the second experimental evaluation, the GMM with improved ROI is implemented and tested on traffic video datasets consisting of various data video frames. The video frames resolution 180 x 240 pixels is considered. Table 3 depicts the two-wheeler results of various video data frames before adjusting ROI values. In this experiment, the actual minimum density count of 67 and the maximum density count of 168 is observed. It generates the different accuracy results for different video data. On an average, the accuracy for two-wheeler vehicle result achieved 81%.

Table 3. Two-wheeler result before adjusting ROI parameters

Video Data	Actual	Two-Wheeler Count	
		Density count	Accuracy (%)
video1.mov	67	59	88.05
video2.mov	85	72	84.70
video3.mov	79	64	81.01
video4.mov	126	98	77.77
video5.mov	168	127	75.59
Average			**81.42**

Source: own work

In Table 4, the results are provided after adjusting the ROI parameters of two-wheeler with three facial directions. Here the same video data frames used

e-ISSN 2357-6014 / Vol. 16, no. 1 / 2020 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

before adjusting ROI values are also considered, nevertheless the accuracy results are improved. The detection of vehicles is improved when front face sided, followed by middle and back directions. In the front face direction, the highest accuracy 94% was observed at video1 and the lowest accuracy 84% was observed at video3. In the middle view direction, the highest accuracy 82% was observed at video1 and the lowest accuracy 49% was observed at video5. In the backward direction, the highest accuracy 50% was observed at video1 and the lowest accuracy 30% was observed at video5. On an average, the accuracy achieved 89% at the front side direction, 65% at the middle side direction, and 45% at the backward direction. Therefore, after careful observation of before and after ROI adjustment values, the average accuracy improvement of 7% was achieved.

Table 4. Two-wheeler result after adjusting ROI parameters

Video Data	Actual	Front	Accuracy (%)	Middle	Accuracy (%)	Back	Accuracy (%)
video1.mov	67	63	94.02	55	82.08	34	50.74
video2.mov	85	78	91.76	64	75.29	42	49.41
video3.mov	79	67	84.81	55	69.62	39	49.36
video4.mov	126	115	91.26	63	50.00	53	42.06
video5.mov	168	146	86.90	83	49.40	64	38.09
Average			**89.75**		**65.27**		**45.93**

Source: own work

Similarly, Table 5 shows the four-wheeler result before adjusting ROI parameter values on various video data frames. The actual data of the video frames and correspondingly the density count data is noted with accuracy. In this experiment, the minimum actual density value 19 and the maximum actual density value 42 were observed. The highest accuracy 79% was maintained at video1 while minimum accuracy 52% was maintained at video5. It is observed that the average accuracy 66% was achieved by considering all five video frames.

e-ISSN 2357-6014 / Vol. 16, no. 1 / 2020 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

Table 5. Four-wheeler result before adjusting ROI parameter values

Video Data	Actual	Four-Wheeler Count	
		Density count	Accuracy (%)
video1.mov	24	19	79.16
video2.mov	36	24	66.6
video3.mov	19	15	78.94
video4.mov	37	20	54.05
video5.mov	42	22	52.38
Average			**66.22**

Source: own work

Table 6 shows the detected results after adjusting ROI parameter values of four-wheeler vehicles. In the front face direction, the highest accuracy 89% was observed at video3 and the lowest accuracy 59% was observed at video5. In the middle view direction, the highest accuracy 78% was observed at video3 and the lowest accuracy 58% was observed at video1. In the backward direction, the highest accuracy 81% was observed at video4 and the lowest accuracy 36% was observed at video3. On an average, the accuracy achieved 76% at the front side direction, 71% at the middle side direction, and 58% at the backward direction. Therefore, after careful observation of before and after ROI adjustment values, the average accuracy improvement of 10% was achieved.

Table 6. Four-wheeler result after adjusting ROI parameters

Video Data	Actual	Front	Accuracy (%)	Middle	Accuracy (%)	Back	Accuracy (%)
video1.mov	24	20	83.33	14	58.33	18	75.00
video2.mov	36	26	72.22	28	77.77	17	47.22
video3.mov	19	17	89.47	15	78.94	7	36.84
video4.mov	37	28	75.67	27	72.97	30	81.08
video5.mov	42	25	59.52	30	71.42	22	52.38
Average			**76.04**		**71.88**		**58.50**

Source: own work

Fig 8 depicts the comparison results of two-wheeler vehicle detection by various directions. The five video data frames were taken and correspondingly measure the vehicle detections. The results are 'Actual', 'Front', 'Back', and 'Middle' directions. Similarly, Fig 9 shows the comparison results of four-wheeler vehicle detection by same directions used in the two-wheeler vehicles.

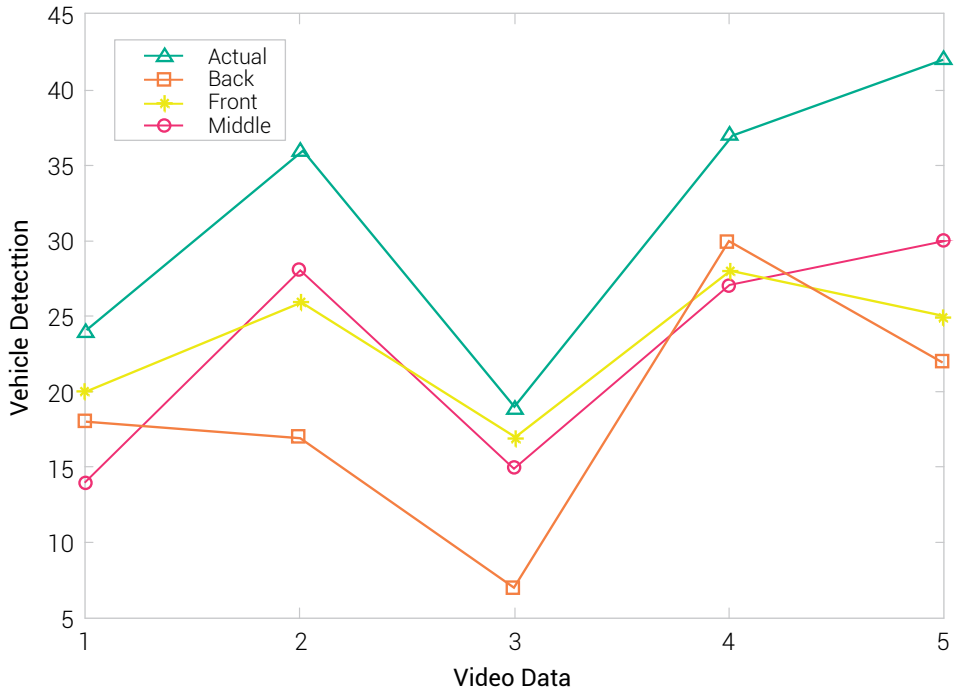

Figure 8. Two-wheeler vehicle detection
Source: own work

Generally, after adjusting ROI parameter values with GMM, the Foreground detection and Medianfilter techniques count the traffic density more accurately. Interestingly, in some video frames, the 'Back' face directed adjusted ROI values produce the superor results when compared with 'Front' and 'Back' directions. In addition, the proposed mechanism achieves more efficient results in two-wheeler vehicles compared to four-wheeler vehicles.

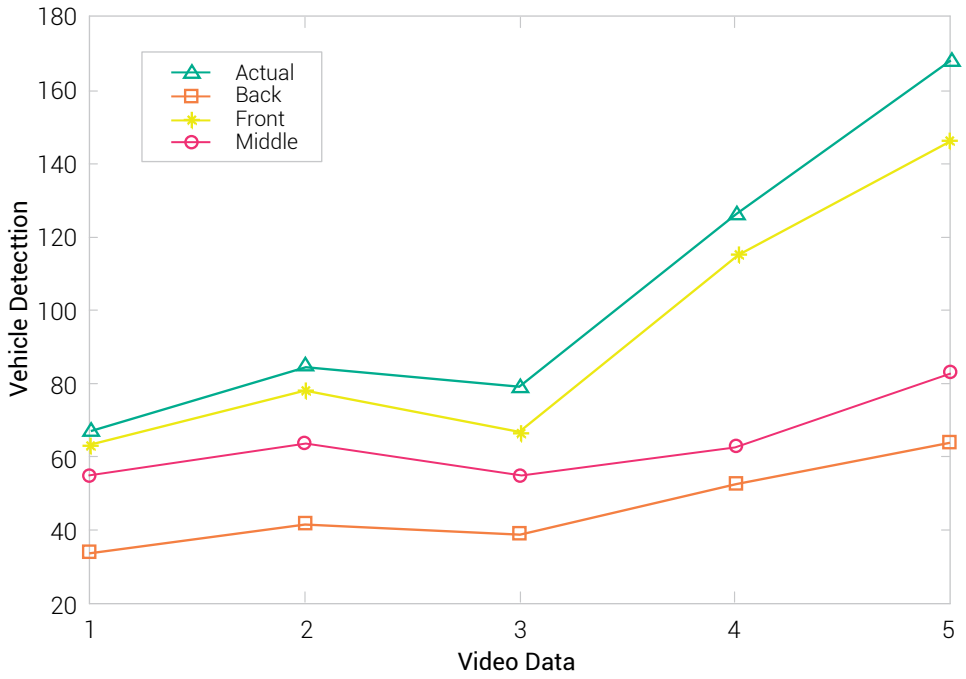

Figure 9. Four-wheeler vehicle detection
Source: own work

The motivation behind the superior result produced in two-wheeler vehicles compared to the four-wheeler vehicles is the number of adjusting ROI parameter values are more and the feature selection with GMM models are accurate. The authors in [20] proposed a framework FSDAA for performing acquisition and analysis on real-time traffic data with a Simulink model and ThingSpeak Cloud platform. They did not used the ROI for detecting the traffic density. Therefore, our proposed MLIGMM_IoTDA mechanism produced the superior results in terms of Accuracy, Precision, F-Score, FPR, TNR, and FNR. In addition, the reasons behind the success of the MLIGMM_IoTDA scheme when compared to the baseline approaches considered for analysis are due to the efficient use of the Simulink model and testing samples inducing greater accuracy, true positive and true negative since they employ better combinations for testing such that maximum alternatives are used for detecting the moving vehicles.

4.3 Time Complexity

In addition, the time complexity of the MLIGMM_IoTDA mechanism is determined to be $T_{Complex} = O(MS) = O(n^2)$. This time complexity is initially derived as $T_{Complex} = O(MS)$

since the processing time majorly depends on the acquiring of the input video sequence 'M' and the number of detected sensor data 'S' in each input video. Further, the number of optimal features influencing the decision making depends on the number of clusters 'C' formed from the input vector of size 'M'. Thus S = O(N) and hence the total time complexity of MLIGMM_IoTDA is $T_{Complex} = O(n^2)$.

5. Conclusions and Future Directions

Data analytics exist everywhere in the Internet of Things (IoT) with huge volumes and variety of data being generated daily. Therefore, how to acquire the real-time sensor data and performing analysis on the sensed data is a challenging task. In this paper, the authors have proposed the MLIGMM_IoTDA mechanism for IoT sensor real-time data analysis. The proposed framework has been implemented on a ThingSpeak IoT cloud platform. A Simulink model was also developed for data analysis and visualizations of the sensed information in real-time. The MLIGMM_IoTDA has been evaluated with accuracy, precision, recall, and F-Score and yields encouraging results compared with state-of-the-art schemes.

The subsequent work of this proposed work is as follows: Firstly, in this framework we didn't provide any security and confidentiality for the MLIGMM_IoTDA, therefore we will accomplish it; Secondly, the same framework has been implemented through the app for mobile users to identify the traffic jam and suggest optimal routes. However, in the future, the authors will be considering these issues to evaluate the proposed MLIGMM_IoTDA mechanism performance.

References

[1] S. Balakrishna and M. Thirumaran, "Semantic Interoperable Traffic Management Framework for IoT Smart City Applications," *EAI Endorsed Transactions on Internet of Things*, vol. 4, no. 13 pp. 1-17, 2018. [Online]. doi: 10.4108/eai.11-9-2018.15548

[2] S. Balakrishna and M. Thirumaran, "Towards an Optimized Semantic Interoperability Framework for IoT-Based Smart Home Applications," in *Internet of Things and Big Data Analytics for Smart Generation.* Balas V., Solanki V., Kumar R., Khari M. (Eds). Intelligent Systems Reference Library, vol 154. Springer, Cham, pp 185-211, 2019.

[3] S. Balakrishna and M. Thirumaran, "Programming Paradigms for IoT Applications: An Exploratory Study", in *Handbook of IoT and Big Data*. Solanki, V., Díaz, V., Davim, J. (Eds.). Boca Raton: CRC press, Taylor & Francis Group, pp 23-57, 2019.

[4] S. Balakrishna, V. Kumar Solanki, V. Kumar Gunjan and M. Thirumaran, "Performance Analysis of Linked Stream Big Data Processing Mechanisms for Unifying IoT Smart Data" *International Conference on Intelligent Computing and Communication Technologies (ICICCT)*, Springer, pp. 680-688, 2019.

[5] S. Bhadra, A. Kundu and S. K. Guha, "An Agent based Efficient Traffic Framework using Fuzzy", *Fourth International Conference on Advanced Computing & Communication Technologie*s, pp. 57, 2014.

[6] A. Mallik, H. Ghosh, S. Chaudhury and G. Harit, "MOWL: An Ontology Representation Language for Web-based Multimedia Applications", *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 10, no. 1, pp. 21, 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), IEEE,* p. 10, 2013.

[8] Q. Zhang, T. Huang, Y. Zhu, and M. Qiu, "A case study of sensor data collection and analysis in smart city: provenance in smart food supply chain," *International Journal of Distributed Sensor Networks*, vol. 9, no. 11, pp 382-132, 2013.

[9] K. Kotis and A. Katasonov, "Semantic Interoperability on the Web of Things: The Smart Gateway Framework", *Proceedings of the Sixth International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2012)*. Palermo, pp 25-35, 2012.

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

[11] 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*, pp. 15-20, 2017.

[12] 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), IEEE*, pp. 251-257, 2013.

[13] 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, IEEE, pp. 216, 2012.

[14] M. Mazhar Rathore, P. Anand, A. Awais and R. Suengmin , "Urban planning and building smart cities based on the internet of things using big data analytics," *Computer Networks*, Elsevier, pp. 1-22, 2016. [Online]. doi: 10.1016/j.comnet.2015.12.023

[15] A.P. Plageras, K.E. Psannis, C. Stergiou, H. Wang and B.B. Gupta, Efficient IoT-based sensor BIG Data collection-processing and analysis in Smart Buildings, *Future Generation Computer Systems,* Elsevier, pp 1-15, 2017. [Online]. https://doi.org/10.1016/j.future.2017.09.082

[16] Q. Liu Qiang Liu, Y. Ma, M. Alhussein, L. Peng and Y. Zhang, "Green data center with IoT sensing and cloud-assisted smart temperature controlling system," *Computer Networks*, Elsevier, pp 1-11, 2015. [Online]. http://dx.doi.org/10.1016/j.comnet.2015.11.024

[17] S. Pasha, "Thingspeak Based Sensing and Monitoring System for IoT with Matlab Analysis," *International Journal of New Technology and Research (IJNTR)*, vol. 2, no. 6, pp. 19-23, 2016.

[18] A. Adnan, G. Kousiouris, H. Pervaiz, J. Sancho, P. Ta-Shma, F. Carrez, and K. Moessner, "Real-time probabilistic data fusion for large-scale IoT applications," *IEEE Access*, vol 6, no 4, pp 10015-10027, 2018.

[19] L. Lengyel, P. Ekler, T. Ujj, T. Balogh, and H. Charaf, "SensorHUB: An IoT Driver Framework for Supporting Sensor Networks and Data Analysis," *International Journal of Distributed Sensor Networks, Hindawi*, vol 11, no. 3, pp 1-12, 2015.

[20] S. Balakrishna, M. Thirumaran, and V. Kumar Solanki, "A Framework for IoT Sensor Data Acquisition and Analysis," *EAI Endorsed Transactions on Internet of Things*, vol 18, no. 1, pp 1-13, 2019. [Online]. http://dx.doi.org/10.4108/eai.21-12-2018.159410