International Journal of Hyperconnectivity and the Internet of Things
Volume 7 • Issue 1

An Efficient Machine Learning-Based Cluster Analysis Mechanism for IoT Data

Sivadi Balakrishna, Vignan's Foundation for Science, Technology, and Research, India*

ABSTRACT

The prevailing developments in internet of things (IoT) and other sensor technologies such as cyber physical systems (CPS) and wireless sensor networks (WSNs), the huge amount of sensor data has been generating from various IoT devices and protocols. Making predictions and finding density patterns over such data is a challenging task. In order to find the density patterns and make analysis over real-time dynamic data, the machine learning (ML) based algorithms are widely used to deal with the IoT data. In this article, the authors proposed an efficient ML-based cluster analysis mechanism for finding density patterns in IoT dynamic data effectively. In this proposed mechanism, the k-means and GMM models are used for clustering data analysis. The proposed mechanism has been implemented on ThingSpeak Cloud platform for analysing the data efficiently on daily and weekly basis. Finally, the proposed mechanism acquired superior results than the existing benchmarked mechanisms over all the performance evaluation metrics used for analysis over IoT dynamic data.

KEYWORDS

data analysis, Internet of Things (IoT), Machine Learning (ML), sensor data, ThingSpeak

1. INTRODUCTION

In recent years, the urban population growth is increased tremendously. As per the recent statistics of the World Health Organization (WHO), From 2015 through 2020, the global urban population will expand by 1.86 percent annually. This growth is expected to be 1.63% between 2020 and 2025 and 1,44% between 2025 and 2030. In urban areas, a considerable proportion of cars are owned by a single home, and at least two cars are owned by a single home. Private cars are becoming increasingly popular with urban traffic. This means that transport in metropolitan areas all over the world is becoming one of the biggest concerns (Shafiq et al. 2020). The large majority of individuals trafficking in urban areas leads to congestion, loss of property, waste of time, damage to the environment, and occasionally to the next level of human mortality. As a result, there is a significant need for smart traffic monitoring and strategies for reduction in cities (Zong et al., 2020). The IoT and ML approaches are the best way to overcome this challenge. It ushers in a new era of intelligent traffic control by effectively aggregating travel times.

DOI: 10.4018/IJHIoT.330680 *Corresponding Author

This paper aims to utilize IoT, cloud computing, Raspberry Pi, and ML approaches to enhance collection and data analysis. This research is based on existing scenarios: information generated by IoT device data gathered from roads and gates is accessible to all passengers and users (Liu et al., 2020). The system will be able to detect existing traffic, traffic and anticipate future traffic to urban areas when it collects real-time sensor data using unsupervised learning techniques. After that, sensor data monitoring and sensor data detection have been measured for analyzing and visualizing the acquired data. Drivers can use the system-generated data for the optimum route selection. As a result, the system is dynamically administrated, controlling, and monitoring moving cars. The boundary time and conditions of different traveler times may be vague significantly (D. Singh et al., 2017).

The unsupervised learning-based clustering techniques are significantly applied in transport research areas for identifying travel patterns (Pyykonen et al., 2013; Yu et al., 2012). The k-means is a crisp clustering algorithm based on partitioning and the Gaussian Mixture Model (GMM) is a fuzzy-based clustering technique, which is widely used for grouping transport patterns during peak and off-peak hours. To measure the number of clusters necessary for optimal transportation data clustering, this research uses the silhouette coefficient and elbow method.

The rest of this paper is organized as follows: Section 2 shows the related work of this study. Section 3 describes the proposed IoT data acquisition sensor and cluster analysis framework, Section 4 illustrates the outcome evaluation and the outcomes of tests and analysis of the proposed work. Section 5 finally concludes this paper and other improvements are highlighted.

2. LITERATURE SURVEY

In this section, the related works included over the recent years discussed and inferred the limitations found over it. (Nallaperuma et al., 2019) have addressed various traffic management system limitations and proposed a Smart Traffic Management Platform (STMP) for making quicker decisions, traffic flow forecasting, concept-drift detection, and optimized traffic control decisions by effective utilization of Artificial Intelligence (AI) techniques over the IoT big data streams. The STMP potentially integrates the various IoT sensor data and is able to predict traffic patterns quickly. However, this platform does not address data acquisition and analysis of dynamic data environments. (Puschmann et al., 2017) have proposed a method for the adaptive adjustment of clusters for IoT dynamic data streams. Which gain useful insights from various data sources and make real-time observations and predictive decisions. Moreover, this method significantly detects drifts by dynamically adjusting clusters over the various data segments. They applied their proposed method in a use-case scenario by considering real-time traffic data through potential utilization clustering algorithms. (Lu et al., 2018) have propounded a parallel approach for clustering large traffic data streams with the utilization of moving object density estimation. In addition, they intend to improve cluster efficiency in data streams, they proposed a modern parallel computation framework with high volume, high-speed traffic stream, and minimum delay with optimal clusters. (Nandurge et al., 2017) intend to overcome the limitations of road traffic accidents data analysis, they used a k-means unsupervised clustering algorithm for segmentation of road accident data. Further, the association rule mining technique has widely applied to discover the predictive traffic data patterns occurred over heterogeneous nature of road accident data.

Similarly, in the studies of traffic analysis over network edge, (Hafeez et al., 2020) have proposed anomaly detection mechanism to predict data analysis at edge level. The IoT-KEEPER enforces malicious network activity detection by utilizing fuzzy c-means clustering algorithm. They evaluated their proposed mechanism by utilizing parameters accuracy and false positive rate with comprehensive dataset. (Dommaraju et al., 2020) have introduced a Deep Learning (DL) based technique for traffic prediction accuracy in big data environment. They are processed and analyzed data through multiple perceptron layers such as input, hidden and output. Further, the activation applied on output layer to predict network traffic based on similarity measurement functions. (Shridevi et al., 2019) have

propounded a ML-based approach for on road intelligent path prediction over vehicle trajectory data. They used two ensemble algorithms such as random forest and AdaBoost for model training and testing.

Therefore, the acquisition and analysis of IoT sensor data in an effective method is necessary after careful observation of all these literature reviews. The following are the contributions made to this proposed mechanism:

- To propose a novel mechanism for acquiring and analysing data from IoT sensors by obtaining real-time data.
- To analyze the gathered sensor data on providing some relevant feature extractions to show the daily and weekly wise sensor density conditions.
- To predict the transport patterns, the k-means and GMM techniques applied.
- To assess the accuracy and performance of the proposed mechanism, precision, recall, and f-measure of the proposed mechanism.
- To compare the obtained results with the existing and most relevant state of the art mechanisms in an efficient way.

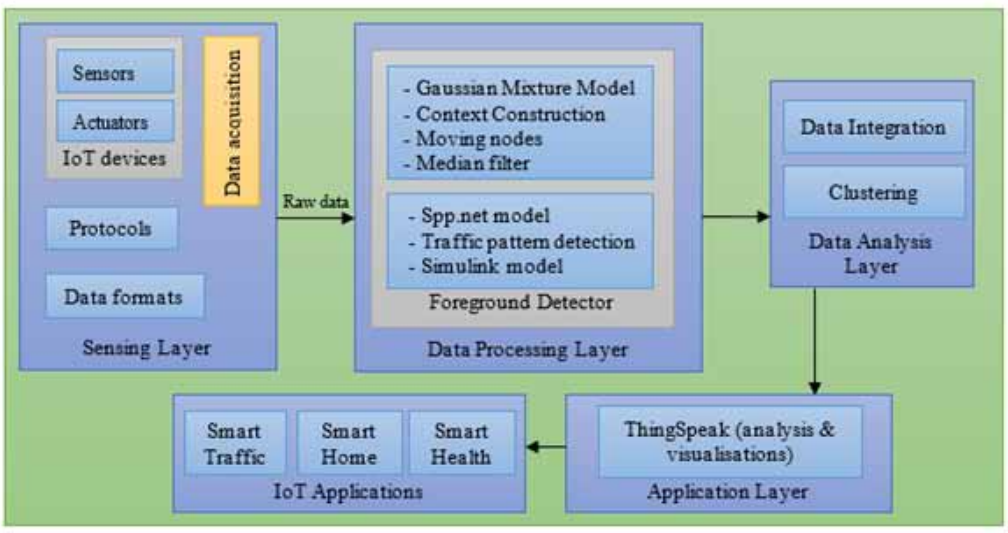
3. PROPOSED MECHANISM FOR CLUSTER ANALYSIS IN IOT DATA

This paper incorporates a ThingSpeak, a Raspberry Pi 3 Model B, and a webcam were used to analyse traffic on a major highway. to develop an intelligent IoT-based traffic monitoring system in real-time. The Raspberry Pi device is then connected to the computer for deployment and the traffic-monitoring algorithm is configured. ThingSpeak is a cloud aggregator that stores, analyses, and visualizes data online.

Once the data collected has been saved, the cloud data is able to analyse the edge device and then analyse the cloud data. Fig.1 presents a workflow analysis architecture and conducts an online examination of the cloud data saved. ThingSpeak is used both for storage and analysis in this context.

As presented in Fig. 1, the workflow of the cluster analysis based on the Internet of Things include four levels. These are the sensor, data processing layer, data analysis layer, and application layer.

Figure1. Work flow of the proposed ML-based cluster analysis mechanism for IoT data



1. **Sensing layer:** This layer is in charge of gathering data from a variety of IoT devices. In the proposed framework, this is the lowest layer seen. It collects a variety of heterogeneous forms of data from IoT devices such as sensors, actuators, protocols, and data formats, among other things. Later, the data has been moved to Data processing layer to process data efficiently.
2. **Data processing layer:** Once the required data has been collected from the physical layer. The data processing layer mainly includes two-submodule such as data extraction and foreground data detection. The extraction of feature is used to extract raw data through the use of camera vision methods such as SPP-net and GMM in order to manage smart traffic data in real time and to identify highway traffic congestion data. Sometimes the data collected might be basic or complicated. Accordingly, both the feature extraction and foreground sensor data are transferred into the next layer on the basis of the representation of sensor data, i.e. IoT sensor data analysis for the performance and interpretation of relevant data using Simulink's and data integration techniques.
3. **Data analysis layer:** For this proposed framework, this is the important layer. This layer was covered with the Simulink model. It is necessary to annotate the IoT sensor data after pre-processing on data processing layer. Applying the Simulink model minimises the ambiguity of the foreground detector data. The domain knowledge systems based on ThingSpeak have utilized context-conscious data to prevent traffic data in real-time to reduce traffic interrupts and unexpected delays.
4. **Application layer:** This layer allows users to analyse and visualise the acquisition of IoT sensor data using a cloud platform from ThingSpeak. This layer creates traffic warnings, number of cars in real time. These evaluated and presented data are valid for intelligent traffic, smart homes, smart grids and smart medical applications.

3.1 Hardware Setup

We have utilized Raspberry Pi 3 Model B and a USB webcam to build a proposed framework. In the second storey of the building facing National Highway 2, Vinukonda, Guntur, A.P., India, we mounted a USB webcam near a window. The camera can position both sides of the highway to provide an excellent view of the vehicles.

3.2 Deploying the Algorithm to the Hardware

We used a template available in MATLAB 2016a to build the traffic monitoring techniques such as Simulink, Image Processing Toolbox, Computer Vision System Toolbox, and Simulink Package for Raspberry Pi 3.

In a Simulink model, the proposed technique is included and the analysis and visualization result are exported. It offers simulation, system design, automatic generation of code, IoT data testing, and visualization. In this context, we construct a Simulink-based model that runs on model B of the Raspberry Pi 3.

It provides IoT data simulation, system-level design, automated code generation, testing, and visualisation. The Raspberry Pi 3 model B was developed using Simulink in this study. An external feature of Simulink for developing the algorithm is employed in this Simulink model. Simulink gathers real-time sensor streaming data from the Raspberry Pi in external mode, and users may see the video on their desktop or mobile device while the Simulink model is running using the SDL video display.

The USB camera is configured for a certain length of highway, with vehicles travelling from left to right. The Raspberry Pi's USB webcam, which is connected to one of the Raspberry Pi's USB ports, records video in a certain area. In this experiment, the Foreground Detector was used to evaluate vision problems and pixels in a sequence of video captured by a USB camera. It generally uses Gaussian mixture model to estimate and decrease excessive noise caused by travellers in the Foreground Detector. The Car Density block keeps track of how many vehicles are travelling eastward and westbound. The video scene is traffic guided. This Car Density Block separates the location of

video capture into two halves with the median highway. The data streaming in real-time supplied to the data aggregator ThingSpeak for analysis & visualisations is thereafter provided. Here, each eastbound or westbound area received a vehicle count value. We transmit traffic to ThingSpeak (channel id) both eastbound and westbound may place field 1 and field2 respectively.

We've centered on accurate detection and provided a strategy for dealing with traffic congestion. In this paper, we broaden our experimental evaluation and implementation with open source components optimised for large-scale IoT data streams, improving our initial method to make it general for heterogeneous IoT sensor data collecting and processing. As illustrated below, Algorithm 1 demonstrates the use of this Simulink model based on sensor data collecting and processing.

In addition, $T_{Complex} = O(MS) = O(n2)$ is the temporal complexity of the MSDACA. $T_{Complex} = O(MS)$ is the first derivation of this time complexity since the processing time is heavily dependent on the acquisition of input video sequence 'M' and the number of detected sensor data 'S' in each input video. In addition, $T_{Complex} = O(MS) = O(n2)$ is the temporal complexity of algorithm 1. TComplex = O(MS) is the first derivation of this time complexity since the processing time is heavily dependent on the acquisition of input video sequence 'M' and the number of detected sensor data 'S' in each input video. Furthermore, the number of optimum features impacting decision-making is proportional to the number of clusters 'C' obtained from the input vector of size 'M', thus $S = O(N)$, and therefore the overall temporal complexity of MSDACA is $T_{Complex} = O(n2)$.

3.4 Analyzing Data on ThingSpeak

The Simulink traffic monitoring model is used in the hardware Raspberry Pi. Then, using the Raspberry Pi 3 Model B to acquire data from ThingSpeak, we can begin analysing, and ThingSpeak cloud aggregator stores real-time event data.

3.4.1 Reading One Week of Data Into MATLAB

ThingSpeak can receive data from the Raspberry Pi every 15 seconds, as we've already mentioned. Furthermore, it only captures around 40,000 data points. In each cycle, all traffic data and time stamps are divided into two vectors, and then the traffic and time are added. Plotting traffic data is the first step, and then we must plot and name the graph. The daily changes scarcely differ between weekdays and non-weekdays. On weekdays (18/12-22/22) we have a different routine than on weekends (16/12 and 17/12).

3.5 Traffic Monitoring in East and Westbound Regions

To illustrate (or quantify) daily traffic density, we need to first average observations on traffic density. A bar chart is a type of data visualization tool. On Monday traffic has been found to be greater than the eastern and western traffic following the weekend. By observing Fig.2, from the start date 16 December to the end date 23 December, the total number of moving vehicles must be counted as both West and Eastbound information on traffic density. The traffic counters are presented in figures 3 (a) and (b) respectively by the east and westbound densities.

3.5.1 Estimating Traffic Density

The real-time traffic tracking system was designed to estimate traffic density by using the DBN model. This process can be done based on the detection of both motional and non-motional vehicles using Machine Learning models such as GMM and SPP.net. Later, we need to find the stopped vehicles as well as motion vehicles under the total list of visible objects.

Thereafter, First, figure out the speed of the vehicles to compute both stationary vehicles and moving vehicles. If the velocity in position is zero, the vehicle is said to be stationary. C^t_{sv}, else the moving vehicle C^t_{mv} is considered. The SPP-net model is used to calculate the number of moving vehicles as:

$C^t_{sv} = C^t_{tv} - C^t_{mv}$, here t ranges from 1 to less than n or equal to n.

The real-time raw traffic data that results is difficult to read and highly spiky. As a result, to see the maximum volume in a single day, we must examine the data on a time scale greater than 15

Figure 2. Density estimation of traffic from first week

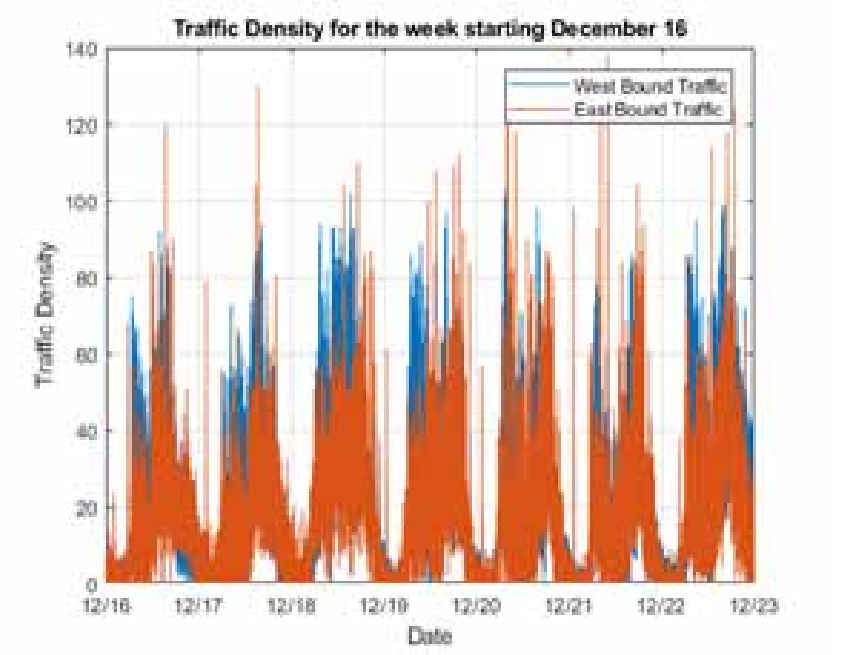
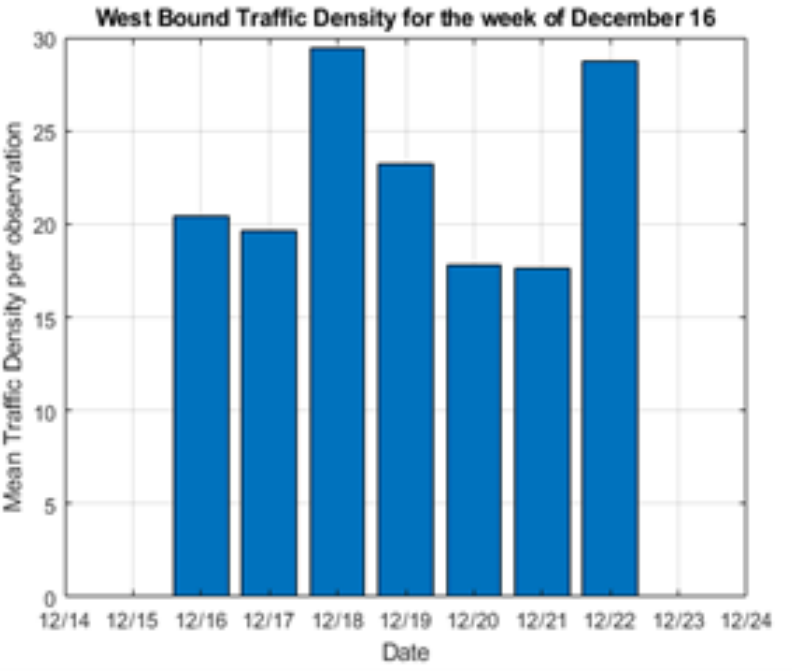


Figure 3. (a) West bound traffic density; (b) East Bound traffic density



seconds throughout a particular timeframe. The 24-hour day in two hour segments will be calculated for the split. Here every part starts at the top of the hour and is 30 minutes later at the completion of

time. Figure 6 displays the traffic information for each day. We have already built an algorithm for traffic monitoring using the Simulink model and installed it on the Pi 3 Model B Raspberry. The information was then uploaded to the ThingSpeak IoT Cloud platform for analysis and visualisation online. Then we analysed and ran a one-week of traffic patterns analyses on them to determine when the largest peak hours occur and when the lowest peak hours occur on a daily basis. In order to measure a number of cars on both sides we employed eastbound and westbound traffic.

For daily and weekly traffic observations, the mean value may be calculated easily. To achieve this, we have a threshold value in our deep observations for discovering modest and/or strong traffic patterns. Whenever the platform ThingSpeak IoT Cloud is visited, this code is immediately updated. Now, as illustrated in Fig.4, we can readily discover the heavy traffic times on this road.

3.6 Dynamic Data Visualizations With ML Models

The dynamic visualizations of the traffic density patterns are viewed through ML based clustering algorithms such as K-means and GMM. Fig.5 and Fig.6 show the clustering results of two clustering algorithms namely k-means and GMM respectively. The color of each scatter data represents which cluster it goes. Based on the observations of road traffic density, the number of clusters are required for clustering dynamic data determined by the elbow method.

During off-peak hours over a week duration, the results obtained by k-means is intends to higher for producing cars than the GMM algorithm at particular period. In addition, the k-means algorithm achieved the higher travelling estimate time (fig. 7). However, overall, the GMM got notable results when compared with the k-means algorithm over different patterns of data. Whereas in peak-hours, the results of the GMM is slightly encourageable than the k-means because of its fuzzification over cluster boundaries. Therefore, for estimation and prediction of traffic patterns over sensor data achieved better results over track changes is quite well in all the days.

The average spot speed of all travellers' data is estimated through clustering algorithms over various time segments for predicting traffic density patterns. In order to examine the clustering algorithms, the road user's traffic log was observed repeatedly at various data segments. On average, 1200 travel users' log was considered and grouped into several clusters based on their density patterns. After careful examination of this data analysis, the GMM achieved higher clustering pattern results than the k-means algorithm. We also tested the obtained results with 1% level of significance through the ANOVA statistical testing tool. This result intends to show that GMM is the more flexible and consistent mechanism for finding traffic density patterns of various road users during off-peak hours.

Figure 4. IoT cloud-based ThingSpeak visualizations and analysis

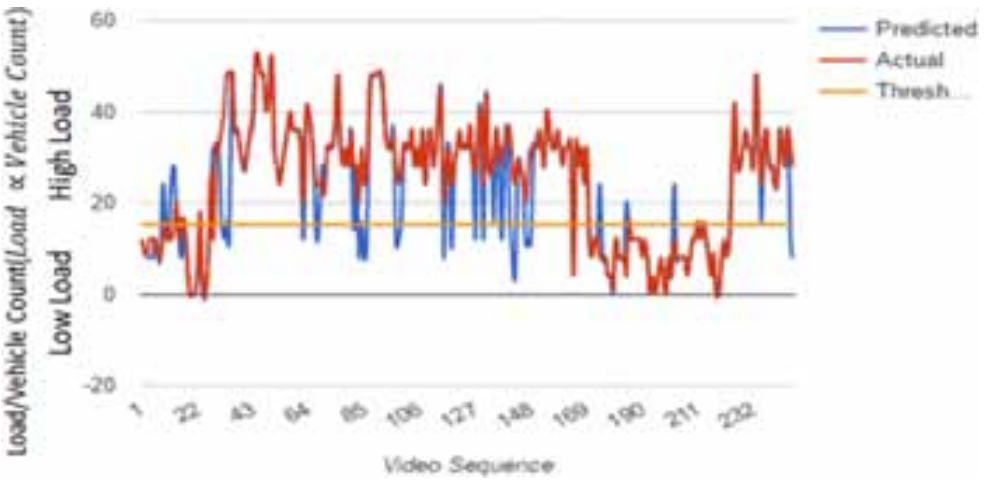


Figure 5. Clusters of traffic density estimates for week

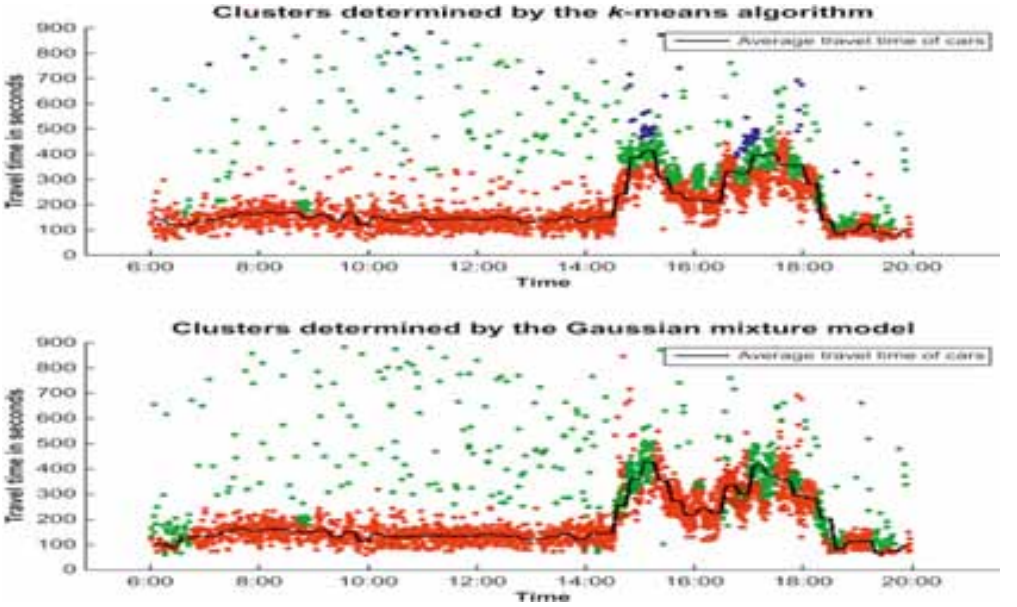
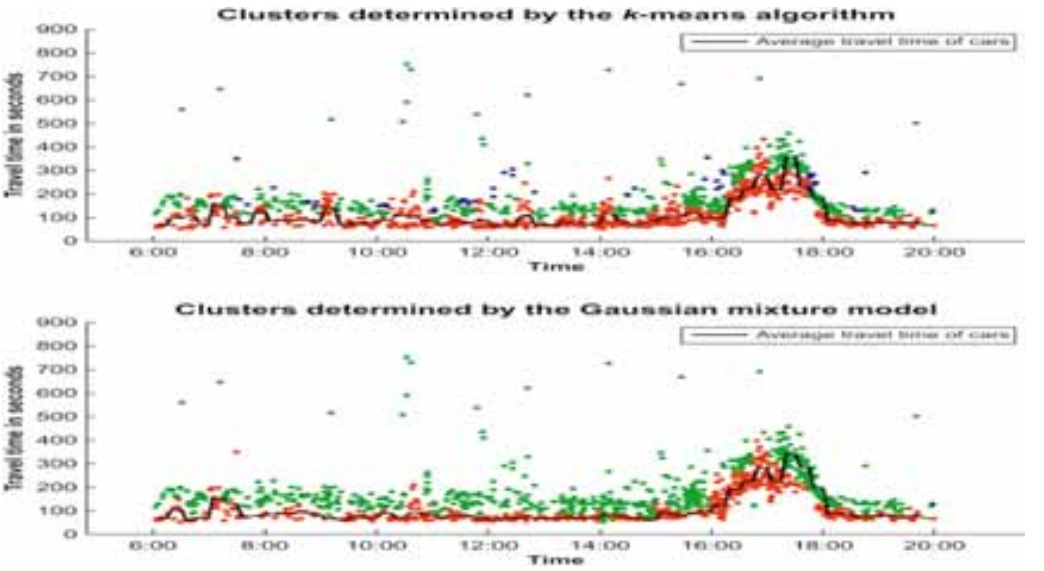


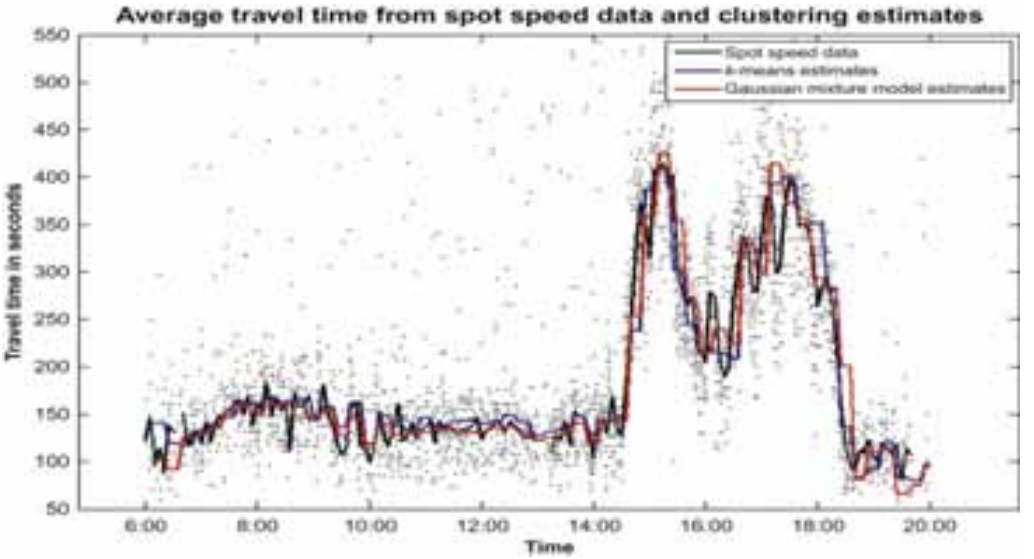
Figure 6. Clusters of traffic density estimates for single day



4. PERFORMANCE EVALUATION

The proposed cluster analysis mechanism was built using the ThingSpeak IoT cloud platform. This has been written using MATLAB code on an I7 Intel Pentium Processor DELL laptop, 8 GB RAM, and 1 TB HDD on the Windows 10 platform.

Figure 7. Travel time estimates of the clustering algorithms over the spotted area



4.1 Performance Metrics

The following metrics are used to measure the performance of the proposed framework in order to evaluate its performance. These measures are derived from the confusion matrix as seen in Table 1.

4.1.1. True Positive

$V_x \rightarrow V_x$: This is an estimate of the vehicles spotted accurately as vehicles detected.

4.1.2. True Negative

$NV_x \rightarrow NV_x$: This is an estimate of vehicles not spotted that are rightly termed undetected vehicles.

4.1.3. False Positive

$NV_x \rightarrow V_x$: This is an estimate of non-detected vehicles misclassified as detected vehicles.

4.1.4. False Negative

$V_x \rightarrow NV_x$: This is an estimate of the number of identified vehicles that were wrongly classified as non-vehicles.

Table 1. Confusion matrix

	Predicted as "YES"	**Predicted as "NO"**
Actually as "YES"	True Positive $\left[V_x \rightarrow V_x\right]$	False Negative $\left[V_x \rightarrow NV_x\right]$
Actually as "NO"	False Positive $\left[NV_x \rightarrow V_x\right]$	True Negative $\left[NV_x \rightarrow NV_x\right]$

4.1.5. True Positive Rate (TPR)

As demonstrated in Eq.1, TPR scales the proportion of correctly classified detected vehicles from the video and reflects the sensitivity.

$$TPR = \frac{V_x \rightarrow V_x}{\left[V_x \rightarrow V_x + V_x \rightarrow NV_x\right]} \tag{1}$$

4.1.6. True Negative Rate (TNR)

As indicated in Eq.2, The percent of accurately recognized non-detected vehicles from the video is represented by TNR.

$$TNR = \frac{NV_x \rightarrow NV_x}{\left[NV_x \rightarrow NV_x + NV_x \rightarrow V_x\right]} \tag{2}$$

4.1.7. False Positive Rate (FPR)

As demonstrated in Eq.3, FPR is a metric that measures the percent of non-detected cars in an input video sequence that are treated as detected vehicles during the data collection process.

$$FPR = \frac{NV_x \rightarrow V_x}{\left[NV_x \rightarrow V_x + NV_x \rightarrow NV_x\right]} \tag{3}$$

4.1.8. False Negative Rate (FNR)

In the data acquisition process, FNR scales the percent of identified vehicles that are regarded as non-detected vehicles, as illustrated in Eq.4

$$FNR = \frac{V_x \rightarrow NV_x}{\left[V_x \rightarrow NV_x + V_x \rightarrow V_x\right]} \tag{4}$$

4.1.9. Accuracy

The first step toward a performance metric is Accuracy, in which the ratio between the entire number of accurate vehicles recognized and the overall number of vehicles identified as shown in Eq. 5 is specified.

$$Accuracy = \frac{(V_x \rightarrow V_x + NV_x \rightarrow NV_x)}{\left[V_x \rightarrow V_x + NV_x \rightarrow NV_x + NV_x \rightarrow V_x + V_x \rightarrow NV_x\right]} \tag{5}$$

4.1.10. Precision, Recall & F-Measure

Precision is focused with the accuracy of the data, whereas Recall is concerned with completeness. The accuracy of the systems ends both with precision and with a reminder although accuracy does not much explain the false results. To determine the score, the F-measure examines accuracy and memory. Its harmonic mean over accuracy and reminder is illustrated in Eq. 6-8.

$$Precision = \frac{V_x \rightarrow V_x}{\left[V_x \rightarrow V_x + NV_x \rightarrow V_x\right]} \tag{6}$$

$$Recall = \frac{V_x \rightarrow NV_x}{\left[V_x \rightarrow V_x + V_x \rightarrow NV_x\right]} \tag{7}$$

$$F - measure = 2 * \left(\frac{Precision * Recall}{Precision + Recall}\right) \tag{8}$$

4.2 Result Discussions

The proposed framework has been compared with existing leading approaches- (Zhang et al., 2013; M. Mazhar Rathore et al., 2016; A.P. Plageras et al., 2017; Sharmad Pasha 2016; Akbar Adnan et al., 2017; and László Lengyel et al., 2015) by considering all the mentioned evaluation metrics in our proposed work.

Table 2 highlights the predominance of Accuracy, precision, recall, f-measure, TNR, FPR, and FNR value of Mechanism for Sensor Data Acquisition and Cluster Analysis (MSDACA) over state-of-the-art schemes under our proposed MSDACA mechanism. The result confirms that MSDACA is able to maintain its accuracy percentage of 84.65 at varying accuracy results of data acquisition even when the false positive rate is increased.

In compared to M. Mazhar Rathore et al., 2016, the MSDACA technique provides a greater accuracy value of 3%. The findings show that MSDACA is able to retain an average precision of 3 to 23% compared to state-of-the-art schemes. Likewise, the accuracy value is higher by 4%, the recall value reaches a greater percentage of 2%, the F-measure is higher than 5%, the TNR is higher than 5%, the FPR is superior to 3%, and the FNR is higher than 5% the existing mechanisms while compared in Table 2. The results shown in Table 2 of our proposed MSDACA are considered as 10 nodes.

Table 3 depicts the results of the various metrics by varying nodes with proposed MSDACA in percentage at the selected region of this paper. The number of vehicles simply called nodes varies starting from 10 to 50. On average, the increase of TPR is 23% by changing the nodes from 10 to 50. TNR is reduced by 86-53%, FPR has dropped by 15-67%, and FNR has declined at the same time by

Table 2. Comparison of the various state-of-the-art schemes with proposed MSDACA in percentage

Mechanism	Accuracy	Precision	Recall	F-Measure	TNR	FPR	FNR
Zhang et.al, 2013	61.28	64.36	59.45	57.25	65.04	27.12	32.15
M. Mazhar Rathore et.al, 2016	70.64	74.08	80.58	78.06	74.15	24.35	27.32
A.P. Plageras et.al, 2017	74.29	76.28	78.26	74.12	67.10	22.61	25.67
Sharmad Pasha, 2016	68.16	70.15	73.64	58.26	62.35	34.15	32.15
Akbar Adnan, et. al, 2018	80.14	79.05	86.01	82.14	81.39	21.68	29.31
László Lengyel et.al, 2015	76.20	79.35	81.29	80.65	74.36	23.16	26.74
MSDACA (Proposed)	**84.65**	**86.24**	**88.75**	**84.85**	**86.40**	**15.54**	**18.10**

Table 3. Results of the various metrics at different nodes with proposed MSDACA in percentage at selected region

Nodes	TPR	TNR	FPR	FNR	Accuracy	Precision	Recall	F-Measure
10	82.11	86.34	15.42	18.22	84.56	86.21	88.60	84.91
20	80.25	75.64	23.34	24.76	81.23	82.45	79.68	81.55
30	73.45	69.82	32.10	29.48	76.02	78.94	71.33	75.61
40	67.96	60.43	41.26	38.36	71.52	71.22	65.20	69.21
50	61.47	53.15	67.00	75.33	65.46	64.30	60.95	62.84

18-75% from 10 nodes to 50 nodes. The percentages of FPR and FNR decreasing means an increase in success rate because the false value in decrease means an increase in the success rate.

As a result of the effective use of the Simulink model and test samples, the reasons for the success of the MSDACA system in comparison with the baseline methodologies for analysis are more accurate, true positive and true negative, since they better combine tests so that maximum alternatives are applied for the detection of moving vehicles.

5. CONCLUSION AND FUTURE DIRECTIONS

The data analysis is available all around the IoT. Nowadays, a massive amount and variety of data are being generated. Therefore, acquiring the sensor data and performing analysis of the sensed data is a challenging task. In this work, the authors presented the MSDACA approach for the collection and analysis of IoT sensors. ThingSpeak IoT cloud platform with Matlab code has implemented the proposed framework. Also developed a Simulink model for data analysis and visualizations of the sensed information in real-time. For finding traffic patterns effectively, ML-based clustering algorithms are used. The MSDACA has been evaluated with accuracy, precision, recall, and f-measure, it yields the encourageable results than the benchmarked mechanisms.

The experiment's next work will be as follows: Since we have not addressed any security and confidentiality issues for the MSDACA, we will do so by implementing possible algorithms. Secondly, the MSDACA is being implemented through the app for mobile users to identify the traffic jam and suggest optimal routes. In future, however, these challenges will be considered for the evaluation of the proposed MSDACA mechanism.

REFERENCES

Adnan, A., Kousiouris, G., Pervaiz, H., Sancho, J., Ta-Shma, P., Carrez, F., & Moessner, K. (2018). Real-time probabilistic data fusion for large-scale IoT applications. *IEEE Access : Practical Innovations, Open Solutions*, *6*(4), 10015–10027.

Dommaraju, V. S., Nathani, K., Tariq, U., Al-Turjman, F., Kallam, S., & Patan, R. (2020). ECMCRR-MPDNL for Cellular Network Traffic Prediction with Big Data. *IEEE Access : Practical Innovations, Open Solutions*, *8*, 113419–113428. doi:10.1109/ACCESS.2020.3002380

Hafeez, I., Antikainen, M., Ding, A. Y., & Tarkoma, S. (2020). IoT-KEEPER: Detecting Malicious IoT Network Activity Using Online Traffic Analysis at the Edge. *IEEE Transactions on Network and Service Management*, *17*(1), 45–59. doi:10.1109/TNSM.2020.2966951

Kamble, S. J., & Kounte, M. R. (2019). On Road Intelligent Vehicle Path Predication and Clustering using Machine Learning Approach, *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC),* (pp. 501-505). IEEE. doi:10.1109/I-SMAC47947.2019.9032648

Lengyel, L., Ekler, P., Ujj, T., Balogh, T., & Charaf, H. (2015). SensorHUB: An IoT Driver Framework for Supporting Sensor Networks and Data Analysis, *International Journal of Distributed Sensor Networks*. *Hindawi*, *11*(3), 1–12.

Liu, C., Liu, C., Liu, F., & Hu, J. (2020). Clustering Analysis of Urban Fabric Detection Based on Mobile Traffic Data. *Journal of Physics: Conference Series*, *1453*(1), 012158. doi:10.1088/1742-6596/1453/1/012158

Lu, J., Feng, J., Zhang, J., Xia, P., & Xiao, X. (2018). A Parallel Approach on Clustering Traffic Data Stream Based on the Density, *2018 Sixth International Conference on Advanced Cloud and Big Data (CBD)*, (pp. 281-286). IEEE. doi:10.1109/CBD.2018.00057

Nallaperuma, D., Nawaratne, R., Bandaragoda, T., Adikari, A., Nguyen, S., Kempitiya, T., De Silva, D., Alahakoon, D., & Pothuhera, D. (2019). Online incremental machine learning platform for big data-driven smart traffic management. *IEEE Transactions on Intelligent Transportation Systems*, *20*(12), 4679–4690. doi:10.1109/TITS.2019.2924883

Nandurge, P. A., & Dharwadkar, N. V. (2017). Analyzing road accident data using machine learning paradigms, *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC).* Palladam. doi:10.1109/I-SMAC.2017.8058251

Pasha, S. (2016). Thingspeak Based Sensing and Monitoring System for IoT with Matlab Analysis, *International Journal of New Technology and Research (IJNTR), 2*(6), 19-23.

Plageras, A. P., Psannis, K. E., Stergiou, C., Wang, H., & Gupta, B. B. (2017). *Efficient IoT-based sensor BIG Data collection-processing and analysis in Smart Buildings, Future Generation Computer Systems*. Elsevier.

Puschmann, D., Barnaghi, P., & Tafazolli, R. (2017). Adaptive Clustering for Dynamic IoT Data Streams. *IEEE Internet of Things Journal*, *4*(1), 64–74. doi:10.1109/JIOT.2016.2618909

Pyykonen, P., Laitinen, J., Viitanen, J., & Eloranta, P. and Korhonen (2013). IoT for Intelligent Traffic System, *International Conference on Intelligent Computer Communication and Processing (ICCP).* IEEE.

Rathore, M. M., Paul, A., Ahmad, A., & Rho, S. (2016). *Urban planning and building smart cities based on the internet of things using big data analytics, Computer Networks*. Elsevier. doi:10.1016/j.comnet.2015.12.023

Shafiq, M., Tian, Z., Bashir, A. K., Jolfaei, A., & Yu, X. (2020). *Data Mining and Machine Learning Methods for Sustainable Smart Cities Traffic Classification: A Survey. Sustainable Cities and Society*. Elsevier. doi:10.1016/j.scs.2020.102177

Singh, D., Tripathi, G., & Jara, A. J. (2017). A survey of Internet-of-Things: Future Vision, Architecture, Challenges and Services. *IEEE World of Forum on Internet of Things*, (pp. 1-8). IEEE.

Yu, X., Sun, F., & Cheng, X. (2012). Intelligent Urban Traffic Management System Based on Cloud Computing and Internet of Things, *International Conference on Computer Science & Service System*. IEEE. doi:10.1109/CSSS.2012.539

Zhang, Q., Huang, T., Zhu, Y., & Qiu, M. (2013). A case study of sensor data collection and analysis in smart city: Provenance in smart food supply chain. *International Journal of Distributed Sensor Networks*, *9*(11), 382132. doi:10.1155/2013/382132

Zong, W., Chow, Y.-W., & Susilo, W. (2020). Interactive three-dimensional visualization of network intrusion detection data for machine learning. *Future Generation Computer Systems*, *102*, 292–306. doi:10.1016/j.future.2019.07.045

Sivadi Balakrishna is currently working as an Associate Professor in Department of Advanced Computer Science and Engineering in Vignan's Foundation for Science, Technology & Research (VFSTR). He has pursued Full-time Ph.D in the Department of Computer Science and Engineering, Pondicherry University (A Central University) in 2020, Puducherry, India. He received his Bachler of Technology (B.Tech) in the Department of Computer Science and Engineering from Jawaharlal Nehru Technological University (JNTU) in 2010 and Master of Technology (M.Tech) in the Department of Computer Science and Engineering from Jawaharlal Nehru Technological University (JNTU) in 2013, Kakinada, AP, India. He has qualified NET (National Eligibility Test) in Dec-2018, which was conducted by UGC. He has published more than 30 research articles in International journals and contributed chapters to several books. He has also presented papers at several International conferences. His current research interests are Machine Learning, Computer Vision and Artificial Intelligence. He may contact at: drsivadibalakrishna@gmail.com