
Cluster-Based Traffic Management for Optimizing Urban Congestion Using Unsupervised Learning on Real-Time Data Streams

Moath Alshorman¹ and Saif taleb²

¹ *Department of Computer Engineering, Hijjawi Faculty for Engineering Technology, Yarmouk university, Irbid, Jordan*

² *school of Information science and computing, Donetsk National Technical University, Lviv region, 82111, Ukraine*

ABSTRACT

Congestion, defined as delays and bottlenecks in traffic due to an excess of vehicles on the road and inadequate infrastructure, is an increasingly pressing issue in today's cities around the globe. Congestion in urban areas is still a significant problem that severely affects people's lives, the environment, and economic output. The current traffic management methods frequently depend on predetermined criteria, and the datasets available are insufficient to deal with complicated and ever-changing traffic situations as they occur in real time. To tackle this issue, this paper proposes CBTMULT, a cluster-based traffic management (CBTM) approach using unsupervised learning techniques (ULT) on real-time data streams from IoT-enabled sensors and traffic monitoring systems. CBTMULT aims to improve traffic flow in metropolitan areas by optimizing congestion management systems and dynamically segmenting traffic patterns. To classify comparable traffic situations according to vehicle density, velocity, and flow rates, the CBTMULT technique employs K-means clustering algorithms. To optimize signal timings, adaptive traffic signal control uses clusters of detected anomalies to guide its decisions. The results show that, on average, vehicle flow rates are 25% higher and that the amount of time vehicles spend in congestion is 25% shorter on simulated urban road networks. The system also outperformed conventional rule-based approaches regarding reaction time to unusual occurrences, which was 30% faster. CBTMULT framework for cluster-based traffic management offers a data-driven, scalable answer to the problem of optimizing urban congestion in real-time, making way for better, more sustainable traffic systems.

Keywords: Urban Congestion, Traffic Management, Unsupervised Learning, Real-Time Data Streams, K-means Clustering Algorithms, Adaptive Signal Control, IoT-Enabled Traffic Systems.

1. Introduction

One of the nagging and increasing problems that cities around the world are facing is urban congestion; the rapid urbanization, increased population, and hikes in vehicle ownership overflow the already existing transport infrastructures [1]. The increased demand for efficiency in mobility systems further exacerbates this problem, hence demanding the call for more intelligent traffic management solution developments [2]. Increased commute times, as a result of congested roads, have bearings on economic productivity; they increase fuel consumption and lead to logistical delays [3]. There is also considerable environmental impact due to congestion in terms of high emissions of greenhouse gases and air pollution. The continuous exposure to vehicle emissions and stress from traffic delays also impacts public health [4]. Traditional traffic control systems have been inefficient in handling these diverse problems associated with the legacy of fixed-signal timing and static route planning [5].

Innovative city initiatives and incorporation of advanced technologies, including IoT, big data analytics, and machine learning, have come up with new capabilities in traffic flow optimization [6]. IoT-enabled devices provide real-time traffic data, while the machine-learning algorithms support dynamic and adaptive traffic control strategies [7].

However, most of the existing traffic control strategies are usually coupled with their dependence on static rules and predefined data sets, which do not consider the dynamic nature of today's urban traffic [8]. These methods become ineffective in dealing with rising unexpected events such as accidents, peak-hour surges, and road closures—conditions that demand real-time interventions and adaptability [9]. Moreover, the volume and complexity of the traffic data generated by these IoT-enabled devices pose a huge challenge in scalability and data analysis in the traditional systems to give prompt and effective solutions [10]. This is even made worse by the lack of intelligent frameworks that couple real-time data analysis with adaptive traffic control [11]. This may call for the data-driven, highly scalable traffic management solution able to dynamically analyze traffic patterns and react to anomalies with the purpose of optimizing resource allocation for reducing congestion and enhancing mobility within cities [12].

CBTMULT is a cluster-based traffic management framework using unsupervised learning approaches for real-time optimization of urban traffic. It clusters traffic conditions dynamically using K-Means clustering based on critical metrics such as vehicle density, speed, and flow rates. Anomaly detection algorithms embedded into the system allow the identification and adaptation to unexpected disruptions such as accidents or road blockages. Adaptive traffic signal control strategies are put in place to adjust timings with respect to the clustered traffic patterns for optimum traffic flow. Further, real-time data acquisition is achieved through IoT-enabled sensors, and dimensionality reduction techniques ensure scalability to guarantee efficient processing. CBTMULT tries to offer a robust, adaptive, and scalable solution to cure congestion in urban areas effectively.

a) Contributions

- To propose CBTMULT, a cluster-based traffic management framework that optimises unsupervised learning for real-time urban congestion.
- To integrate K-Means clustering and anomaly detection for dynamic traffic condition analysis and disruption identification.
- To implement adaptive traffic signal control strategies to improve traffic flow and reduce congestion duration.
- To demonstrate significant improvements in congestion management, achieving a 25% increase in flow rates and a 30% faster response to anomalies.
- To provide a scalable and data-driven approach for enhancing urban traffic systems in smart cities.

First, an overview of urban congestion and related challenges will be outlined. Then, the proposed CBTMULT framework and methodology will be provided. This will be followed by discussions on the results and major findings highlighted with improved performance. Conclusions will summarize implications and limitations and suggest future research directions.

2. Related works

With increasing complexity in urban traffic systems and a high prevalence of disturbances such as accidents, roadblocks, and sudden congestion, the need for innovative approaches to anomaly detection has been driven. This aims to discuss the development of methodologies, from traditional statistical to current machine learning models, and their strengths, limitations, and suitability in

dynamic traffic scenarios. Special consideration is given to integrating IoT-enabled sensors and data-driven systems, which have significantly increased real-time monitoring capabilities. Taking lessons from these steps forward, the review highlights the current research gaps in adaptive and scalable traffic management solution requirements and points toward further opportunities. Table 1 shows the summary of the literature Review.

Table 1. Summary of the literature review

Author	Problem Definition	Methods Used	Result	Advantages	Disadvantages	Research Gap
Alkaissi, Zainab Ahmed [13]	Traffic congestion evaluation of urban streets	Fuzzy Inference System (FIS) and GIS application	Evaluation of traffic congestion using an integrated approach of FIS and GIS	Combines spatial and fuzzy logic effectively	Limited to evaluation; lacks real-time application	Extending the methodology for real-time traffic management and prediction
Yang, Bo, et al. [14]	Urban traffic congestion alleviation	Millimeter wave radar and improved probabilistic neural network (PNN)	Efficient congestion detection and alleviation mechanisms using radar and improved PNN	High accuracy and applicability in varying traffic conditions	Implementation cost is high due to hardware dependency	Scalability and optimization in resource-limited scenarios
Tsalikidis, Nikolaos, et al. [15]	Urban traffic congestion prediction using multiple data sources	Sensor data and weather information with a multi-step prediction approach	Improved prediction accuracy of traffic conditions over multiple steps	Multi-source data integration improves robustness	High computational overhead	Exploration of simpler, cost-effective prediction algorithms
Saillaja, V., et al. [16]	Enhancing roadwork safety using traffic cones	IoT-embedded cones with CNN-based object detection	Real-time detection of roadwork zones and safety improvement	Real-time safety enhancement	Limited focus on overall traffic management	Generalizing object detection for dynamic traffic environments

Bhardwaj, Akashdeep, et al. [17]	Traffic data analysis for smart IoT devices	IIoT for traffic flow analysis and modeling	Enhanced understanding of traffic patterns and flow	Efficient integration with smart city IoT infrastructure	Limited to descriptive analysis; lacks predictive capabilities	Extending into predictive traffic modeling
Chahal, Ayushi, et al. [18]	Traffic congestion prediction in IoT-enabled smart cities	Hybrid univariate model	Accurate congestion prediction in IoT-enabled settings	Tailored for IoT applications	May not generalize to non-IoT urban environments	Expanding the model for more complex, real-world scenarios
Luz, Hivez, et al. [19]	Dynamic traffic flow management	AI algorithms	Optimized traffic flow management	Effective adaptation to traffic dynamics	Lack of specificity in AI approaches used	More detailed validation and comparison of AI techniques in traffic management
Sahare, Mahendra, et al. [20]	Congestion avoidance in FANET and IoT environments	Fuzzy logic approach	Improved congestion avoidance mechanisms	Effective in specific IoT and FANET scenarios	Limited applicability outside FANET environments	Exploring broader applications and scalability for other networks

3. Proposed Scheme

a) Dataset Explanation

The Traffic Prediction Dataset is a real-world traffic dataset for modelling and predicting urban traffic patterns. The dataset includes information such as a timestamp, the number of vehicles, speed, and weather conditions like temperature and humidity to help analyze traffic flow under different scenarios. The data is time-series-based with multiple observations at regular intervals, which is very apt for machine learning tasks in traffic prediction and congestion management. It has been labelled in such a manner as to provide complete support in applications of predictive modelling, adaptive signal control, and real-time anomaly detection for smart city traffic systems [21].

b) *Proposed Scheme*

CBTMULT uses unsupervised learning via K-means clustering to analyze real-time data streams from IoT-enabled sensors and traffic monitoring systems. It will dynamically recognize different traffic states by partitioning traffic patterns into clusters based on variables such as vehicle density, speed, and flow rates; then, it uses these clusters to optimize traffic signal control by adjusting timings based on current traffic states. The primary CBTMULT rationale is orchestrated to enhance traffic flow in an urban environment via efficient congestion management and delay reduction through a general improvement in the efficiency of traffic performance in real-time, hence countering challenges brought about by urban congestion within dynamic environments. Figure 1 shows the proposed CBTMULT process.

c) *Data Acquisition and Preprocessing*

The data is acquired in real time from several input sources: IoT-enabled sensors, traffic cameras, and vehicle GPS systems installed at strategic locations along the urban road network. Other parameters like vehicle density, speed, flow rates, signal timings, and historical traffic patterns are collected to provide the data in a comprehensive manner. Preprocessing involves filtering noise and removing irrelevant data to increase the model's accuracy. This then uses data normalization to make the scaling consistent with different sensors and imputes missing data via interpolation or with the help of machine learning methods to ensure complete and reliable information for analysis.

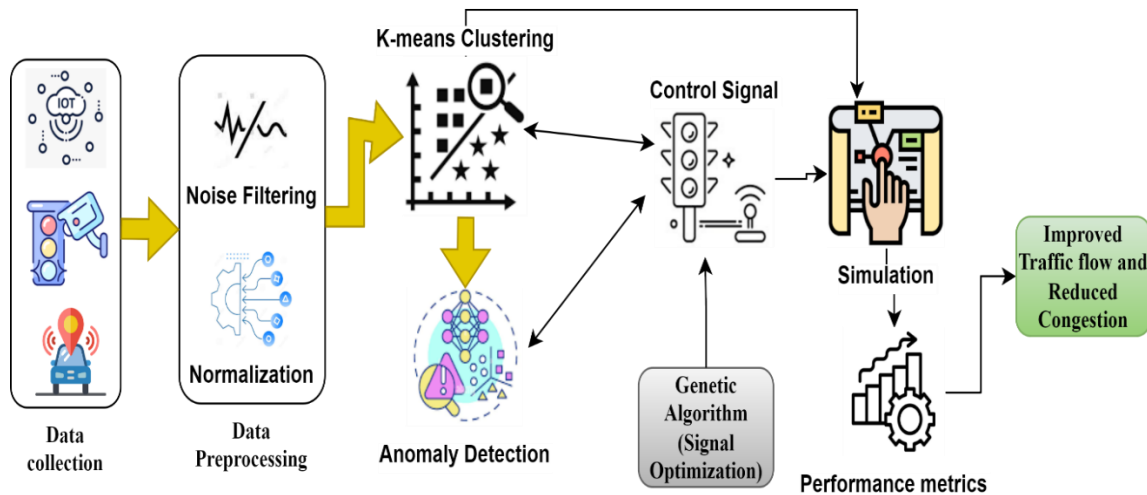


Figure 1. Process of CBTMULT method

d) *Traffic Pattern Clustering*

Unsupervised Learning Techniques: K-Means clusters traffic data into distinct groups based on vehicle density, average speed, and flow rates. This will enable the finding of patterns like "high traffic congestion," "moderate traffic flow," or "low traffic volume". The K-Means algorithm works fine for large datasets, which suits the urban traffic systems that produce sizeable real-time data streams. Since centroids' updating is done frequently, K-Means adapts to the dynamic nature of traffic, an essential aspect of real-time monitoring. K-Means is the foundation for organizing heterogeneous traffic data into manageable clusters, making further analysis and decisions easier.

$$J = \sum_{k=1}^K \sum_{i=1}^{n_k} w_k \cdot \frac{\|x_i^k - \mu_k\|^2}{\sigma_k^2} + \lambda \cdot \sum_{k=1}^K \frac{\sum_{i=1}^{n_k} f(x_i^{(k)}, t)}{n_k}$$

$$f(x_i^k, t) = \alpha \cdot (x_i^k - \mu_k) \cdot \Delta t \tag{1}$$

where K is the number of clusters (e.g., high congestion, moderate flow, low flow). x_i^k is the data point i in cluster k , characterized by features like vehicle density, speed, and flow rates. μ_k refers to the centroid of the cluster k . σ_k^2 is the variance within the cluster k , used to normalize cluster tightness. w_k refers to the weight assigned to the cluster k based on its relevance (e.g., higher for congested clusters to prioritize resolution). $f(x_i^k, t)$ is a dynamic function representing the temporal evolution of traffic data. α is a sensitivity parameter, and Δt represents the time interval for updates. n_k is the number of data points in the cluster k . λ is the regularization parameter balancing clustering objectives and dynamic adaptation.

e) Cluster Validation Metrics

i. Silhouette Score:

The Silhouette Score is crucial to CBTMULT clustering quality. A data point's fit into its cluster is measured by balancing cohesion, how close data points are, separation, or how diverse clusters are. A high Silhouette Score indicates well-defined clusters with data points around centroids and far from others. CBTMULT computes the Silhouette Score for various clusters (K) to find the best K without under- or over-segmentation. For congestion and free flow analysis, this groups traffic patterns meaningfully. The Silhouette Score helps fine-tune the clustering process for better traffic monitoring and decision-making in the CBTMULT architecture. Figure 2 (a) represents the Silhouette Score, and figure 2(b) illustrates the Davies-Bouldin Index.

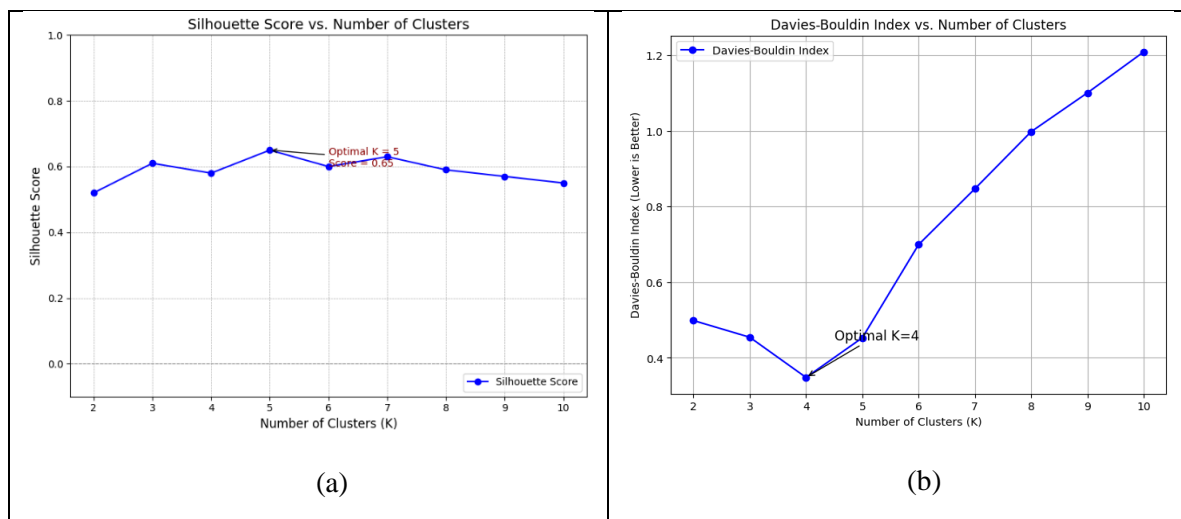


Figure 2. Silhouette Score and Davies-Bouldin Index analysis of traffic monitoring

ii. Davies-Bouldin Index

The Davies-Bouldin Index (DBI) is a significant CBTMULT clustering quality validation measure. Internal cluster similarity and inter-cluster distance are measured. Lower DBI scores mean better clustering, with data points well-grouped and segregated. This feature permits robust traffic pattern classification, separating high congestion, moderate flow, and low traffic volume. DBI also lets you compare clustering setups or techniques to find the optimal one. DBI assures that K-Means produces compact and distinct clusters in CBTMULT, improving traffic pattern detection accuracy and traffic management judgments.

f) *Anomaly Detection*

Traffic management anomaly detection detects sudden real-time traffic flow disturbances that could significantly impact flow. Examples include collisions that rapidly shift vehicle speed or density. Blocked roads or lanes produce strange and aberrant patterns, and severe weather deteriorations may cause abrupt congestion. This would allow real-time identification of anomalies and immediate measures, such as traffic rerouting or signal time adjustment, to minimize congestion and increase network efficiency and safety.

g) *Statistical Analysis:*

Statistical analysis in traffic management considers real-time traffic parameters, such as speed (s), flow (f), and density (d), to find the outliers that deviate significantly from normal traffic behaviour. Starting with data collection for these parameters, descriptive statistics are computed: mean (μ) and standard deviation (σ) that establish the regular traffic pattern. Similarly, anomalies can be identified by defining a threshold for deviation from the mean by computing an anomaly score, $A(x_i) = \frac{|x_i - \mu|}{\sigma}$. If the anomaly score exceeds a predefined threshold, the data point is marked as an anomaly; such anomalies can correspond to disruptions such as accidents or roadblocks.

h) *Machine Learning Models:*

Machine learning techniques can detect strong anomalies in high-dimensional traffic data. Two of the very popular models are Isolation Forests and DBSCAN.

i. *Isolation Forest:*

Isolation Forest (iForest) is a technique for detecting anomalies that use recursive data space partitioning to isolate them. It is based on the premise that outliers are denser and more compact than typical data points. The technique starts by building a set of randomly selected decision trees. Each node picks a feature at random and then picks a threshold at random to divide the data. The path length ($h(x)$) is then calculated for each data point (x), with anomalies being quickly isolated and thus having low path lengths. Finally, the anomaly score is given by $A(x) = 2^{-\frac{h(x)}{c(n)}}$ where ($c(n)$) is the average path length for regular points. iForest is good at handling high-dimensional data and can identify individual anomalies and subtle disruptions in traffic patterns, such as slight traffic slowdowns or sudden surges in vehicle density.

ii. *DBSCAN (Density-Based Spatial Clustering of Applications with Noise)*

DBSCAN is an anomaly detection approach that clusters data points based on their density and marks outliers as anomalies. It starts by calculating the ϵ -neighborhood for each data point (x), which denotes all points not farther than a distance (ϵ) from (x). A point (x) is considered a core point if the number of points in its ϵ -neighbourhood, ($|N_\epsilon(x)|$), is greater than or equal to a predefined threshold, MinPts. DBSCAN then groups density-connected cluster points into clusters and flags those not contained in any cluster as anomalies. This approach is efficient for non-uniformly distributed datasets and can find clusters of any shape, thus being particularly suitable for complex and dynamic traffic environments.

i) *Response mechanisms*

When the traffic system finds any anomalies, an immediate response mechanism is operated to manage the disruptions effectively. Starting with immediately sending alerts to traffic control centres,

navigation systems, and traffic applications (such as Waze or Google Maps) so that relevant parties are informed in real time. Optimized response protocols are activated by dynamic calculation of alternative routes for traffic diversion, signal timing adjustment to reduce congestion at incident areas, and emergency services dispatching, such as ambulances, tow trucks, or police, to incident scenes. The loop is closed with feedback: the system monitors the effectiveness of interventions already in place. It readjusts strategies as needed, driven by real-time data to ensure maximum congestion management effectiveness.

j) Adaptive Traffic Signal Control

The dynamic signal timing algorithm adjusts green light timings based on real-time traffic data and emergency vehicle priority to enhance traffic flow at a crossroads with Route A and Route B. Initial green time is 60 seconds for Route A and 30 seconds for Route B. The technology continuously monitors line lengths and densities to determine route congestion. If an ambulance is identified on Route B, the system overrides the signal plan and gives Route B full green time until the emergency clears. In the absence of an emergency, the system modifies green times dynamically based on congestion: Route A receives more time if it's congested, while Route B gets more if it clears. Green timings are proportional to both routes' queue lengths. Traffic stabilization (low congestion on both routes) resets the system to an optimum signal timing schedule. Traffic is managed efficiently, and emergency vehicles are prioritized in real time. Table 2 shows the Genetic algorithm for traffic signal optimization.

Table 2. Genetic Algorithm (GA) process to optimize traffic signal

```

N = 100 # Population size
max_iterations = 1000
Pc = 0.8 # Crossover rate
Pm = 0.1 # Mutation rate
w1,w2,w3 = 0.5,0.3,0.2 # Weight factors
population = initialize_population(N)
def fitness(individual):
    Tq,Td,Ev =
calculate_Tq_Td_Ev(individual) # Calculate queue length, delay, and priority
    return w1 * Tq + w2 * Td + w3 * Ev
fitness_values = [fitness(ind) for ind in population]
for iteration in range(max_iterations):
    selected_parents = select_population(population,fitness_values)
    offspring = []
    for i in range(0,N,2):
        if random() < Pc:
            child1,child2 = crossover(selected_parents[i],selected_parents[i + 1])

```



```
    offspring.extend([child1, child2])
  else:
    offspring.extend([selected_parents[i], selected_parents[i + 1]])
  end if
end for

for i in range(len(offspring)):
  if random() < Pm:
    offspring[i] = mutate(offspring[i])
  end if
end for

fitness_values_offspring = [fitness(ind) for ind in offspring]
combined_population = population + offspring
combined_fitness = fitness_values + fitness_values_offspring
sorted_indices = sorted(range(len(combined_population)), key =
lambda x: combined_fitness[x])
population = [combined_population[i] for i in sorted_indices[:N]]
fitness_values = [combined_fitness[i] for i in sorted_indices[:N]]
best_fitness = min(fitness_values)
print(f"Iteration {iteration + 1}: Best Fitness = {best_fitness}")
best_individual = population[fitness_values.index(min(fitness_values))]
print("Optimized Signal Timing Plan: ", best_individual)
```

4. Result and discussions

a) Performance Metrics

Comparing the CBTMULT method with other traditional ones, such as the Fuzzy Inference System and GIS Application (FIS-GIS) [13], the Millimeter Wave Radar and Improved Probabilistic Neural Network (IPNN) [14], and AI Algorithms for Dynamic Traffic Flow Management, AIDTFM [19]. The comparison is based on three key metrics: Congestion Duration Reduction, measuring the precision of each method in predicting traffic conditions; Vehicle Flow Rate Improvement, measuring efficiency in processing and delivering actionable results; and Response Time Improvement, measuring the feasibility and adaptability of each method for broader deployment. This analysis points out how CBTMULT can go further than the existing methods.

b) Congestion Duration Reduction (CDR):

CDR measures the percent reduction in average time vehicles are in congestion within a traffic system. It estimates a traffic control system's effectiveness in reducing delays due to heavy vehicle density and bottlenecks. This is calculated as in eqn 2.

$$CDR = \frac{\sum_{i=1}^n w_i (CD_{baseline,i} - CD_{method,i})}{\sum_{i=1}^n w_i \cdot CD_{baseline,i}} \times 100 \quad (2)$$

where w_i is the weight assigned to each zone based on traffic density or priority. $CD_{baseline,i}$ refers to the congestion duration in the zone i under traditional methods. $CD_{method,i}$ is the congestion duration in the zone i under CBTMULT. n is the total number of zones or intersections.

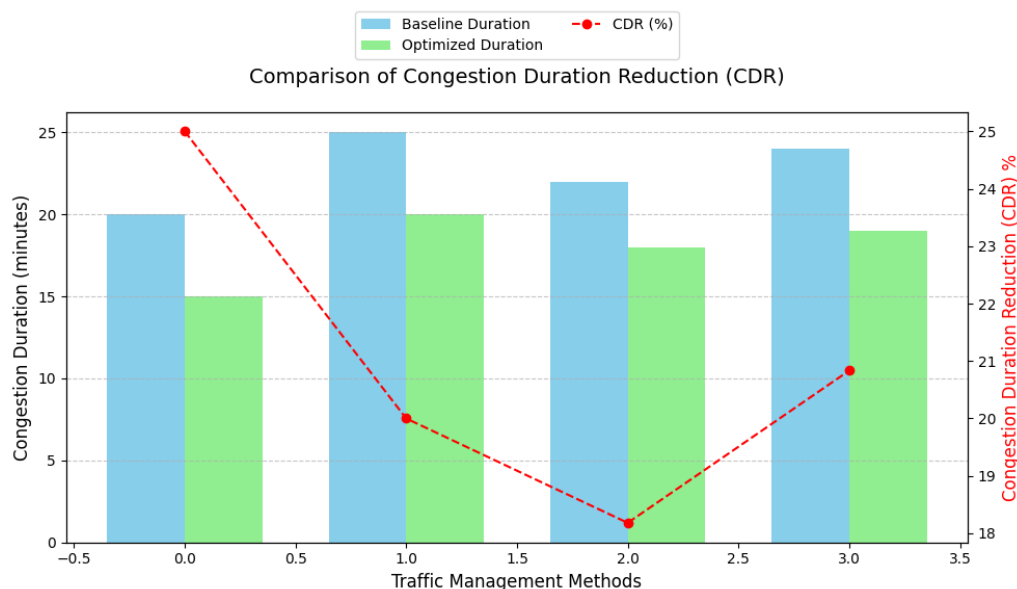


Figure 3. CDR Analysis

Figure 3 compares the performance of traffic management methods based on congestion duration in minutes and congestion duration reduction in percentage—CDR. The baseline and optimized congestion durations are represented by bars, while the CDR percentage is plotted as a line. The CBTMULT method has the shortest optimized congestion duration of 15 minutes and the highest CDR of 25%, indicating the best performance. FIS-GIS, IPNN, and AIDTFM have longer durations and lower reductions, which reflects their relative inefficiency. The dual y-axes show a comprehensive visualization of absolute durations and percentage improvements, emphasising how CBTMULT is effective in congestion management by minimizing delays through clustering and anomaly detection for real-time management.

c) Vehicle Flow Rate Improvement (VFRI):

VFRI is the percentage increase in the average number of vehicles that pass through a section of road in each period. This measure assesses how well a traffic control system maximizes road use and optimizes vehicle flow. This is obtained in the equation 4.

$$VFRI = \frac{\sum_{i=1}^m \frac{VFR_{method,i} - VFR_{baseline,i}}{Capacity_i}}{\sum_{i=1}^m \frac{VFR_{baseline,i}}{Capacity_i}} \times 100 \quad (3)$$

where $VFR_{method,i}$ refers to the vehicle flow rate on the road segment i under CBTMULT. $VFR_{baseline,i}$ is the vehicle flow rate on the road segment i under traditional methods. $Capacity_i$ is the maximum vehicle capacity of the road segment i . m is the total number of road segments.

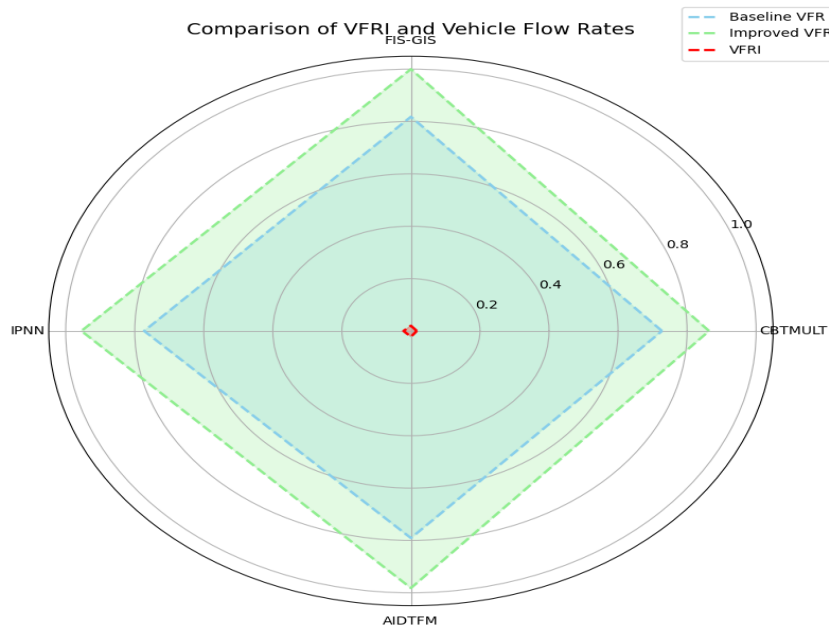


Figure. 4. VFRI Analysis

Figure 4 displays the Vehicle Flow Rate Improvement comparisons and the baseline and improved vehicle flow rates for each traffic management method: CBTMULT, FIS-GIS, IPNN, and AIDTFM. The chart's axes represent different methods, and the lines show normalized performance values so they can be compared clearly. The baseline VFR is blue for current performance, the improved VFR is green for enhancements, and the VFRI is red for the relative improvement percentages. Shaded areas show differences, so it is easy to see which methods work best at increasing traffic flow efficiency.

d) Response Time Improvement (RTI):

RTI is the time a traffic management system saves in responding to abnormal events, such as congestion or accidents, compared with a benchmark system. RTI is quantified as a percentage and estimated as in equation 4.

$$RTI = \frac{\sum_{j=1}^k \alpha_j \cdot (RT_{baseline,j} - RT_{method,j})}{\sum_{j=1}^k \alpha_j \cdot (RT_{baseline,j})} \times 100 \quad (4)$$

where α_j refers to the weight factor for anomaly type j (e.g., accidents, roadblocks) based on severity or frequency. $RT_{baseline,j}$ is the response time for anomaly j under traditional methods.

$RT_{method,j}$ refers to the response time for anomaly j under CBTMULT. k is the total number of anomaly types.

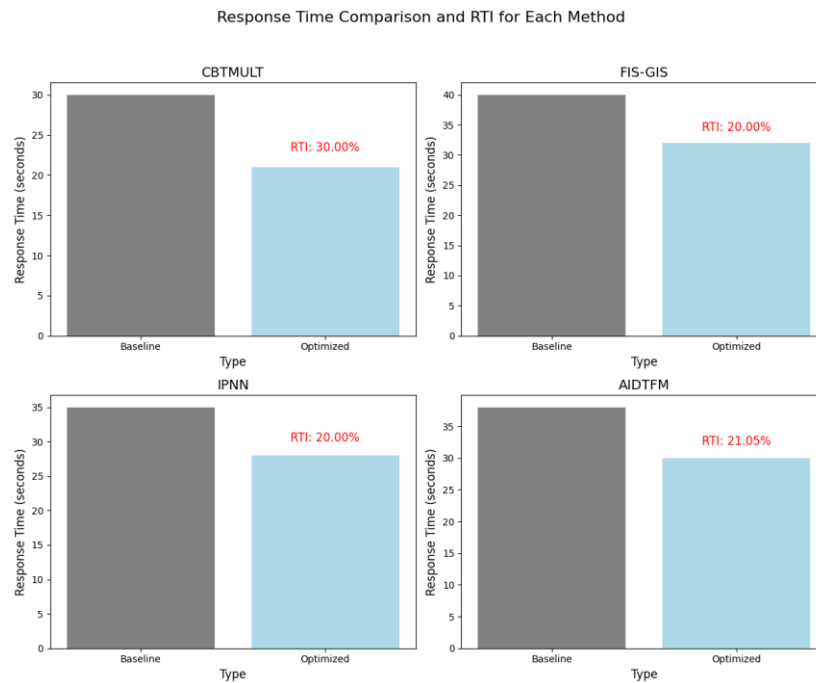


Figure. 5. RTI analysis

Figure 5 presents the performance of each traffic management method: CBTMULT, FIS-GIS, IPNN, and AIDTFM, in terms of response time. The bars in each subplot represent two types: Baseline Response Time and Optimized Response Time; on top of the optimized bar, RTI (%) is noted. The layout allows for the individual analysis of how much each method contributes to a reduction in response times and, accordingly, improvement of efficiency. CBTMULT has the shortest response time and the highest RTI, possessing the best real-time anomaly-handling ability. The visual arrangement lets one compare the methods clearly, showing the effectiveness of optimization strategies for urban traffic management.

5. Conclusion

The CBTMULT framework successfully utilizes unsupervised learning techniques, such as K-Means clustering, to enhance urban traffic management through dynamic segmentation of traffic patterns and adjustment of traffic signal timings in real-time. The system integrates anomaly detection methods, including statistical analysis, Isolation Forest, and DBSCAN, to efficiently recognize and respond to disruptions like accidents, roadblocks, and sudden congestion. The response mechanism ensures immediate action by alerting traffic control centres, rerouting traffic, adjusting signal timings, and deploying emergency services if needed, thus improving the flow of traffic and easing congestion. The system's feedback loop allows continuous monitoring and optimization for effective long-term management. This may also be pursued further by developing anomaly detection models that are more robust, scalable solutions for large cities and integration with machine learning techniques that will help predict future traffic patterns from historical data. Further, integrating the system with autonomous vehicles and advanced transportation technologies can make it even more effective and adaptive in the dynamic urban environment.

References

- [1]. Beojone, Caio Vitor, and Nikolas Gerolimimis. "On the inefficiency of ride-sourcing services towards urban congestion." *Transportation research part C: emerging technologies* 124 (2021): 102890.
- [2]. Zhong, Renxin, et al. "Special issue on methodological advancements in understanding and managing urban traffic congestion." *Transportmetrica A: Transport Science* 18.1 (2022): 1-4.
- [3]. Bendib, Abdelhalim. "The effects of spatial clustering of public facilities on social equity and urban congestion in the city of Batna (Algeria)." *GeoJournal* 87.2 (2022): 861-874.
- [4]. Rahman, Md Mokhlesur, et al. "Traffic congestion and its urban scale factors: Empirical evidence from American urban areas." *International journal of sustainable transportation* 16.5 (2022): 406-421.
- [5]. Cvetek, Dominik, et al. "A survey of methods and technologies for congestion estimation based on multisource data fusion." *Applied Sciences* 11.5 (2021): 2306.
- [6]. Li, Chenguang, et al. "Analysis of Urban Congestion Traceability: The Role of the Built Environment." *Land* 13.2 (2024): 255.
- [7]. Kavididevi, Venkatesh, et al. "IoT-Enabled Reinforcement Learning for Enhanced Cold Chain Logistics Performance in Refrigerated Transport." *2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS)*. IEEE, 2024.
- [8]. Oladele, Oluwaseyi Kolawole. "Internet of Things (IoT) integration and its impact on smart cities, automation, and data-driven decision-making." (2024).
- [9]. Hassan, Muhammad, et al. "Smart City Intelligent Traffic Control for Connected Road Junction Congestion Awareness with Deep Extreme Learning Machine." *2022 International Conference on Business Analytics for Technology and Security (ICBATS)*. IEEE, 2022.
- [10]. Shukla, Praveen, C. Rama Krishna, and Nilesh Vishwasrao Patil. "IoT traffic-based DDoS attacks detection mechanisms: A comprehensive review." *The Journal of Supercomputing* 80.7 (2024): 9986-10043.
- [11]. Dui, Hongyan, et al. "IoT-enabled real-time traffic monitoring and control management for intelligent transportation systems." *IEEE Internet of Things Journal* (2024).
- [12]. Firdhous, Mohamed Fazil Mohamed, B. H. Sudantha, and Naseer Ali Hussien. "A framework for IoT-enabled environment aware traffic management." *International Journal of Electrical and Computer Engineering* 11.1 (2021): 518.
- [13]. Alkaissi, Zainab Ahmed. "Traffic congestion evaluation of urban streets based on fuzzy inference system and GIS application." *Ain Shams Engineering Journal* 15.6 (2024): 102725.
- [14]. Yang, Bo, et al. "Urban traffic congestion alleviation system based on millimeter wave radar and improved probabilistic neural network." *IET Radar, Sonar & Navigation* 18.2 (2024): 327-343.
- [15]. Tsalikidis, Nikolaos, et al. "Urban traffic congestion prediction: a multi-step approach utilizing sensor data and weather information." *Smart Cities* 7.1 (2024): 233-253.
- [16]. Saillaja, V., et al. "IoT-Embedded Traffic Cones with CNN-based Object Detection to Roadwork Safety." *2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*. IEEE, 2024.
- [17]. Bhardwaj, Akashdeep, et al. "IIoT: traffic data flow analysis and modeling experiment for smart IoT devices." *Sustainability* 14.21 (2022): 14645.
- [18]. Chahal, Ayushi, et al. "A hybrid univariate traffic congestion prediction model for IOT-enabled smart city." *Information* 14.5 (2023): 268.
- [19]. Luz, Hives, et al. "Dynamic Traffic Flow Management Using AI Algorithms." (2025).
- [20]. Sahare, Mahendra, Priti Maheshwary, and Vinay Kumar Dwivedi. "A Fuzzy Approach for Congestion Avoidance in FANET and IoT." *Nine* 34.35 (2025): 36.
- [21]. <https://www.kaggle.com/datasets/hasibullahaman/traffic-prediction-dataset>