Journal of Applied Automation Technologies

Research Paper

Application of Cloud-Based Real-Time Data Processing in Automated Traffic Management Systems

Grażyna Grabiec[1], Sławomir Cyra[1,*] and Jarosław Bogdan Kalisz[1]

[1] Faculty of Mechatronics and Automation, Cracow University of Technology, Krakow, 31-155, Poland
*Corresponding author: slawomir.c@pk.edu.pl

Abstract. Autonomous traffic control requires a cloud-based real-time data processing system, and several data sources, including cameras, detectors, linked cars, and signal controllers in cities, continuously produce data. A cloud-edge cooperative processing framework for automated traffic management systems is proposed in this paper. The framework includes low-latency alert, multi-source traffic condition fusion, cloud-based stream analysis, edge processing, and roadside data gathering. Adaptive scheduling will be utilized to dynamically distribute computer resources according to event immediacy, queue length, and traffic stream priority. The suggested approach lowers the average end-to-end latency to 134 ms (from 286 ms) and boosts the processing throughput to 52,000 records/s while also reducing the average intersection delay by 18.6%, according to the experiment results in a simulated urban traffic environment. Based on the study, it is concluded that a scalable system capable of high-quality real-time traffic processing can be constructed by combining edge cooperation and flexible scheduling.

Keywords: *Real-Time Data Processing, Automated Traffic Management, Edge-Cloud Collaboration, Traffic Flow Analysis*

Received on 29 May 2022, Accepted on 23 December 2022, Published on 06 Jan 2023

Introduction

The automatic traffic management system now uses a variety of real-time data sources, including road cameras, loop detectors, connected cars, roadside units, cell phones, and signal controllers. Traffic flow, congestion development, accidents, lane occupancy, wait duration, vehicle speed, intersection status, etc. are all included in the data streams. The present urban transportation network's traffic-control platform must continuously handle a variety of data types and cannot rely solely on historical analysis when an issue arises. Large-scale traffic management issues can benefit from cloud computing, which offers dispersed analytics, unified data services, scalable computing resources, and elastic storage [1]. High-speed data capture and low-latency processing of data from field devices to the control center are also necessary for real-time awareness of traffic conditions, according to research on intelligent transportation [2]. In the transportation industry, stream computing solutions offer the technical support required for continuous event processing [3]. This function is further expanded by cloud-edge collaboration, which keeps large-scale optimization and historical analysis in the cloud and moves latency-sensitive computation closer to the road [4].

Nonetheless, there are still certain technical issues with processing traffic data in the cloud. Different frequencies, formats, noise levels, and spatial coverage are all present in traffic data. While detector recordings and vehicle probe data are modest but could be absent, video streams are comparatively voluminous and bandwidth-heavy. A central cloud pipeline may encounter queue aggregation, transmission delay, and an inconsistent response time when multiple crossings produce a spike in traffic at the same time [5]. Although edge computing requires job distribution and coordination with cloud resources, it has a modest local delay [6]. Although real-time stream processing frameworks are high-throughput ingestion devices, their scheduling policies need to be adaptable to deal with variations in traffic load, abrupt spikes brought on by incidents, and a variety of device workloads [7].

https://doi.org/10.64972/jaat.2023v1.280p2e:21-33

The majority of traffic management systems in use today prioritize either local reaction speed or cloud scalability, while autonomous traffic control necessitates both. As a result, a successful system should be able to combine low-latency event response, adaptive stream scheduling, and multi-source data fusion into a unified cloud-based architecture [8]. The combined effects of data quality, processing speed, and decision-feedback influence the system's overall performance, according to research on big data analytics for intelligent transportation [9]. Static batch learning is not an option for real-time traffic prediction and event detection, which also require constant feature updates [10]. Some tools for this are distributed resource allocation, microservice orchestration, and cloud-native data services [11]. Studies on traffic light regulation have demonstrated that processing delays directly cause queues to form and lower intersection efficiency [12]. As a result, the cloud platform's response quality under stress must be taken into account in addition to computing speed [13].

This research investigates cloud-based real-time data processing for automated traffic management systems. Create a cloud-edge cooperative processing framework to gather different traffic streams, combine traffic data from multiple sources, dynamically schedule processing jobs, and enable low-latency event detection and control response. Boost processing power, lower overall traffic event delays, and enhance response times in situations involving extensive metropolitan roads.

Related Work

Cloud-Based Traffic Data Processing

The intelligent transportation system includes centralized data storage and comprehensive analytical tools, and it is now simpler to grow and scale thanks to the cloud-based resources. Create a uniform data lake or streaming warehouse after gathering road recordings, detector logs, GPS trajectories, and connected-vehicle messages on a cloud platform. The cloud is appropriate for large-scale data storage and analysis of past trends, traffic forecasting, multi-region coordination, etc., according to research on cloud-based intelligent transportation systems. [14] Distributed processing can lessen the strain on central servers as the number of vehicles and sensors rises, according to research on Big Data traffic analysis [15]. The traffic data service can be separately deployed and updated using cloud-native frameworks built on microservices and containers without impacting the platform as a whole [16]. Hybrid cloud-edge architectures have been established because pure cloud processing may cause an unacceptable latency for event detection and signal response [17].

Real-Time Stream Computing in Intelligent Transportation

Continuous data collection, event windowing, state changes, and low-latency computing are all components of real-time stream processing. Stream processing is used by a traffic management system to identify unexpected speed reductions, forecast travel times, detect traffic congestion, and dynamically modify traffic light timings. Event-time windows, backpressure control, and operator parallelism are necessary for stable throughput, according to stream computing studies [18]. In a traffic scenario, the arrival rate of sensor data is not constant; instead, there will be abrupt surges in data because of accidents, rush hours, and other factors [19]. Simultaneously, the real-time traffic forecast method's features must be updated on a regular basis; otherwise, the control decision will be erroneous or delayed [20]. Although distributed stream engines can increase throughput, task scheduling and resource allocation in computing nodes have an impact on their actual performance [21].

Edge-Cloud Collaborative Traffic Management

Through local responsiveness and cloud-level coordination, cooperation between the edge and the cloud has begun to accomplish the goal of autonomous traffic control. While the cloud performs network-wide optimization and long-term learning, edge nodes temporarily store data locally and carry out lightweight video analysis and event filtering close to junctions. Local pre-processing can lower communication burden and speed up safety-critical event reaction, according to edge computing research in intelligent transportation [22]. Additionally, cooperative perception and low-latency vehicle-to-infrastructure services are supported by multi-access edge computing [23]. Task placement should take into account data quantity, compute cost, network latency, and event priority, according to research on cloud-edge orchestration [24]. The studies serve as the

https://doi.org/10.64972/jaat.2023v1.280p2e:21-33

foundation for developing a real-time traffic management architecture that collaboratively supports edge and cloud resources in automated traffic decision-making [25]. The traffic application requires collaboration between sensing, communication, and distributed computing, according to studies on Internet-of-vehicles [26]. According to research on smart-city data processing, scalable service orchestration is necessary for extensive road networks [27]. Studies on traffic flow prediction have demonstrated that the reliability of the subsequent control would be impacted by the quality of real-time data [28]. Uneven stream loads can be solved by research on cloud service scheduling [29]. The computational performance and traffic response quality of cloud platforms need to be taken into account, according to research on intelligent transportation analytics [30].

Cloud-Based Real-Time Traffic Data Processing Framework

System Architecture and Data Flow Modeling

Roadside data collection, edge preprocessing, cloud stream processing, traffic status fusion, event detection, and control feedback are the parts of this system. Data streams are continuously produced by linked cars, radar sensors, loop detectors, roadside cameras, and signal controllers. Timestamp alignment, noise filtering, lightweight feature extraction, and emergency event pre-screening are all carried out by edge nodes. After receiving a structured stream tuple in the cloud, carry out incident confirmation, global fusion, and control decision support. The cloud-edge collaborative architecture for autonomous traffic control is shown in Figure 1.

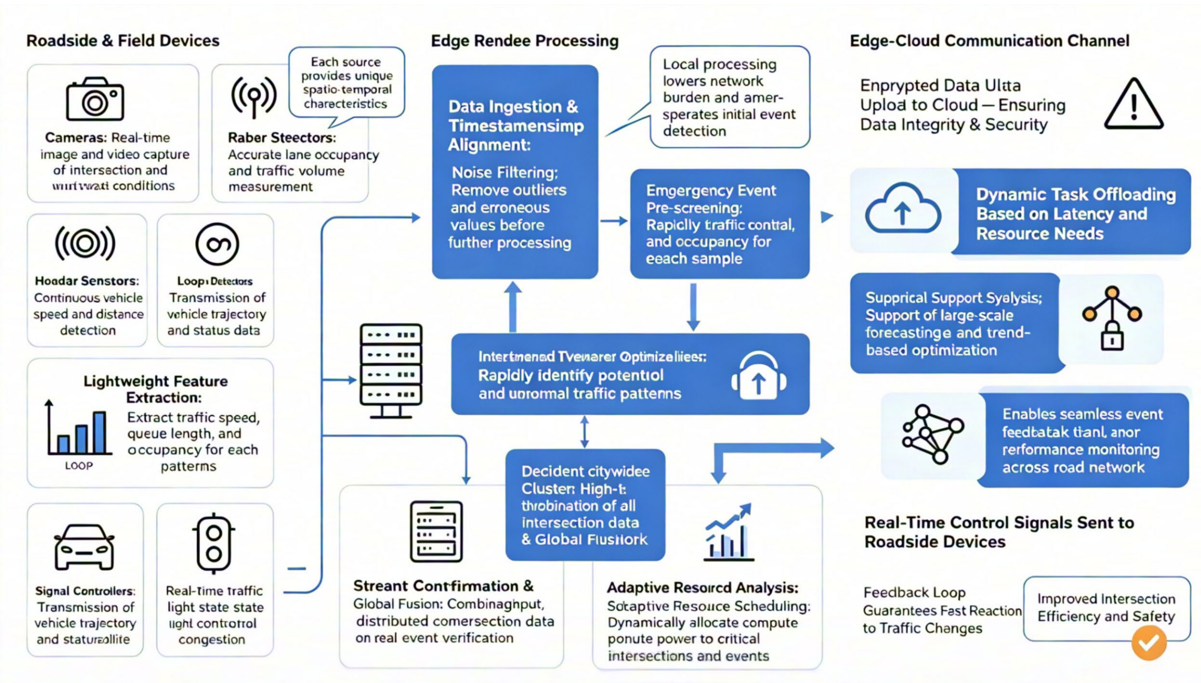

Figure 1. Cloud-edge collaborative architecture for automated traffic management

The incoming traffic stream is represented as a time-indexed collection of heterogeneous observations. Each observation contains source identifier, timestamp, location, feature vector, and priority label.

$$X_t = \{x_{i,t} \mid i = 1, \dots, M\} \qquad \text{Eq.(1)}$$

To describe end-to-end processing, the total delay is decomposed into acquisition, transmission, queuing, computation, and feedback components.

$$T_{\text{end}} = T_{\text{acq}} + T_{\text{net}} + T_{\text{queue}} + T_{\text{comp}} + T_{\text{fb}} \qquad \text{Eq.(2)}$$

The cloud-edge assignment variable determines whether a task is processed locally or transmitted to the cloud. This variable is optimized according to latency sensitivity and resource availability.

https://doi.org/10.64972/jaat.2023v1.280p2e:21-33

$$a_i = \arg\min_a \left(\lambda_T T_{i,a} + \lambda_C C_{i,a}\right) \quad \text{Eq.(3)}$$

Multi-Source Traffic Stream Fusion and Scheduling

A reliable traffic-state representation of noisy and partial sensor data is produced by combining many sources. Dense visual information and queuing data are provided by camera features; occupancy and speed are provided by radar and loop sensors; and trajectory and travel-time samples are provided by linked vehicles. Local intersections' fused traffic status is updated every five seconds, while the region's is updated every thirty seconds.

$$S_t = \sum_i w_{i,t} h_i(x_{i,t}) \quad \text{Eq.(4)}$$

The reliability weight of each source changes with packet loss, detection confidence, and temporal freshness. A source with delayed or noisy data receives lower weight during fusion.

$$w_{i,t} = \frac{\exp(-d_{i,t}) q_{i,t}}{\sum_j \exp(-d_{j,t}) q_{j,t}} \quad \text{Eq.(5)}$$

Adaptive stream scheduling assigns computing capacity to high-priority intersections and event streams. The scheduler increases resources for abnormal congestion, crash suspicion, and emergency vehicle priority.

$$R_{i,t} = R_0 + \eta P_{i,t} + \mu B_{i,t} \quad \text{Eq.(6)}$$

Queue accumulation is controlled by adjusting processing parallelism. When the queue length exceeds a threshold, the scheduler expands the number of operators allocated to the corresponding stream.

$$K_{t+1} = K_t + \left\lceil \frac{\max(0, Q_t - Q_{\max})}{Q_{\text{step}}} \right\rceil \quad \text{Eq.(7)}$$

Low-Latency Event Detection and Response Optimization

The event detection module determines whether congestion, an abrupt decrease in speed, an illegal halt, lane obstruction, or a malfunctioning sensor are the results of the combined traffic circumstances. The multi-source traffic stream fusion and event response module is shown in Figure 2.

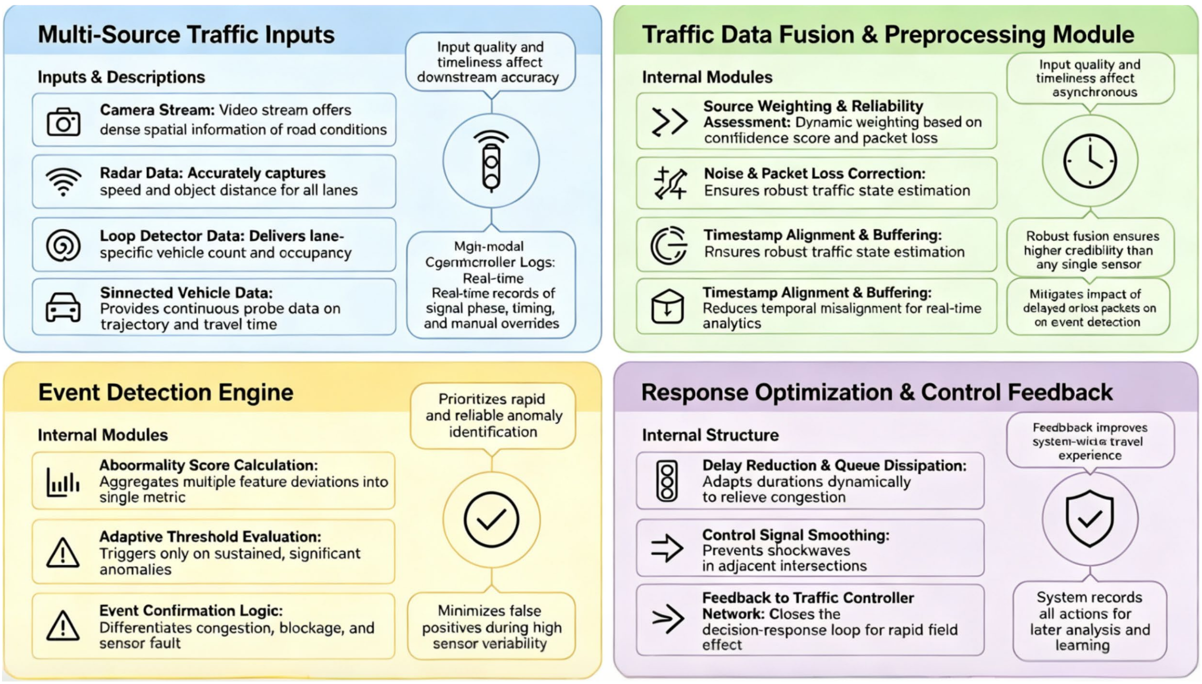

Figure 2. Multi-source traffic stream fusion and event response module

The abnormality score combines traffic speed deviation, queue growth rate, occupancy change, and detection confidence.

$$A_t = \beta_1 |v_t - v_{\mathrm{ref}}| + \beta_2 \Delta Q_t + \beta_3 O_t \quad \text{Eq.(8)}$$

An event is confirmed only when the abnormality score remains above the adaptive threshold for consecutive windows. This prevents short noise bursts from triggering false control actions.

$$E_t = \mathbf{I}(A_t > \theta_t \text{ and } A_{t-1} > \theta_{t-1}) \quad \text{Eq.(9)}$$

The response optimization objective balances delay reduction, queue dissipation, and control smoothness. It prevents abrupt signal changes that may destabilize neighboring intersections.

$$J = D_{\mathrm{avg}} + \alpha Q_{\mathrm{avg}} + \beta \|u_t - u_{t-1}\|^2 \quad \text{Eq.(10)}$$

Experimental Scheme and Evaluation Metrics

Experimental Scenario and Cloud Platform Configuration

There are 48 crossings, 192 cameras, 96 loop detectors, 64 radar sensors, and simulated connected-vehicle messages in the experimental example. Each edge node in the cloud platform's distributed stream processing nodes and containerized microservices is in charge of two or four intersections. During off-peak hours, there are 1.8 million traffic stream records each hour; during peak hours, that number can reach 4.6 million. Test the system with typical traffic, rush-hour traffic, event bursts, and sensor packet loss scenarios.

$$D = \{X_t, S_t, E_t, u_t\} \quad \text{Eq.(11)}$$

Latency, Throughput, And Traffic Response Metrics

Processing efficiency is measured by end-to-end latency, throughput, queue length, and resource utilization. The average latency is computed over all processed stream windows.

$$L_{\mathrm{avg}} = \frac{1}{N} \sum_{n} \left(t_{\mathrm{out},n} - t_{\mathrm{in},n}\right) \quad \text{Eq.(12)}$$

Throughput measures the number of processed traffic records per second. It reflects whether the cloud-edge system can absorb burst traffic streams.

$$P_{\mathrm{thr}} = \frac{N_{\mathrm{proc}}}{\Delta T} \quad \text{Eq.(13)}$$

Traffic response quality is evaluated by incident detection delay, average vehicle delay, queue length reduction, and control stability.

$$G_{\mathrm{resp}} = \frac{D_{\mathrm{base}} - D_{\mathrm{prop}}}{D_{\mathrm{base}}} \quad \text{Eq.(14)}$$

The comprehensive efficiency score combines real-time computing and traffic management benefits, but it is reported as a normalized index in the experimental section rather than used as a separate optimization equation.

Comparative Experimental Design

The four scenarios are contrasted with batch-oriented traffic analysis, fixed-rule stream scheduling, edge-only processing, and centralized cloud processing. The identical traffic inputs and detection algorithms are used by all techniques. The comparison items include end-to-end latency, throughput, packet-loss robustness, incident detection delay, queue length reduction, and resource utilization. For scalability tests, the gallery of situations now offers 48 intersections instead of just 12.

https://doi.org/10.64972/jaat.2023v1.280p2e:21-33

Result Interpretation and Comparative Analysis

Real-Time Processing Performance Analysis

The suggested cloud-edge framework has maintained a throughput of 52,000 records per second while lowering the average end-to-end latency in centralized cloud processing from 286 MS to 134 Ms. The real-time processing results are displayed in Figure 3. The average delay under the four processing architectures is shown in Figure 3(a), the throughput curve as a function of stream size is shown in Figure 3(b), and the queue accumulation graph during peak hours is shown in Figure 3(c). This rise is anticipated based on the findings of a study on edge-cloud collaboration that can lower communication overhead and preserve cloud-level coordination [31].

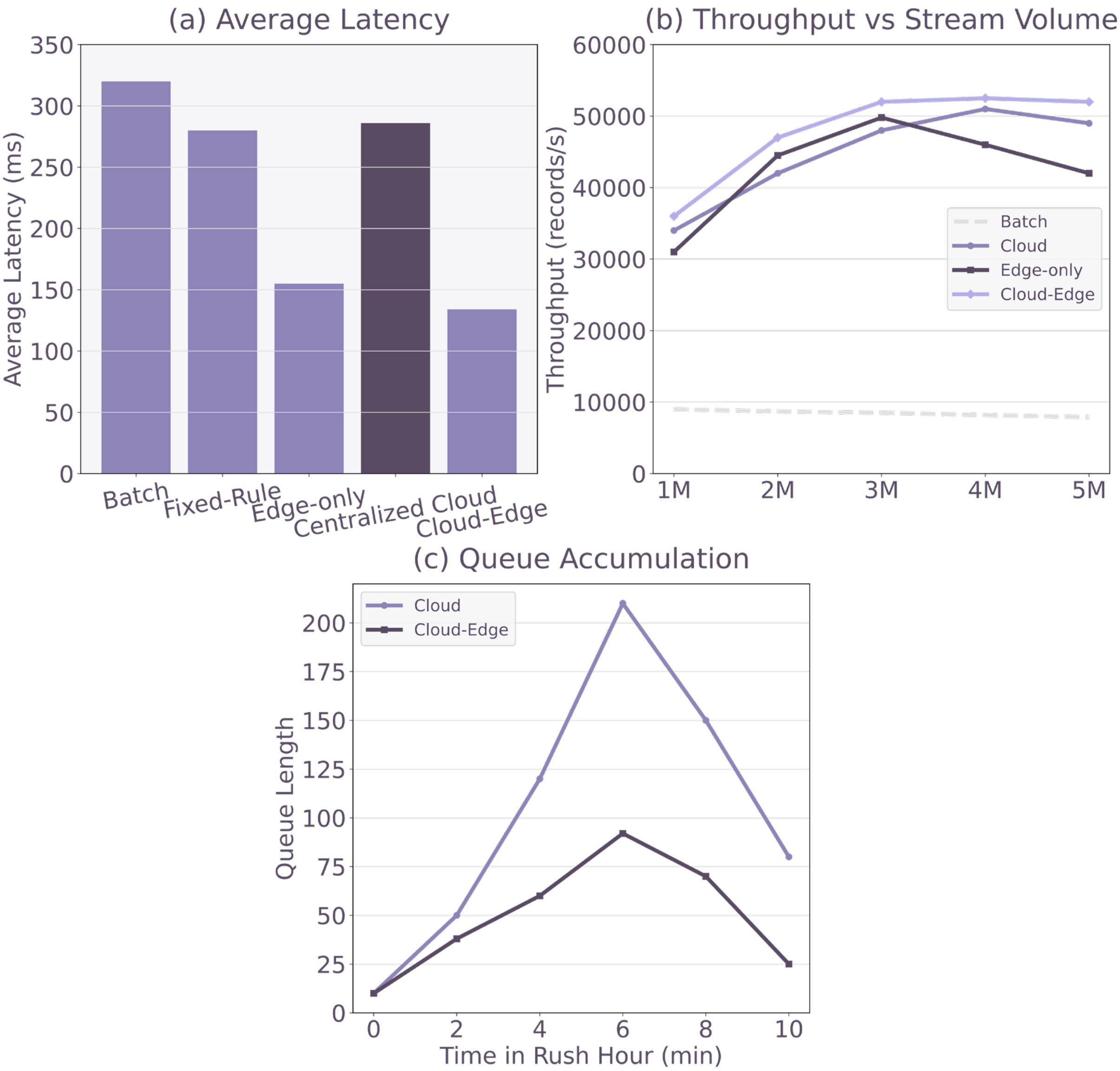

Figure 3. Real-time processing performance comparison. (a) Average latency comparison. (b) Throughput under increasing stream volume. (c) Queue accumulation during rush-hour bursts

Traffic Event Detection and Flow Control Analysis

Event detection performance can be enhanced and false alarms brought on by single-sensor noise can be decreased by using multi-source fusion. The suggested system's average lane obstruction detection time in the event burst scenario is 2.8 seconds, whereas the centralized cloud processing takes 5.9 seconds. The event

https://doi.org/10.64972/jaat.2023v1.280p2e:21-33

detection performance is depicted in Figure 4, with Figure 4(a) showing the detection delay, Figure 4(b) showing the false alarm rate, and Figure 4(c) showing the missed detection under packet loss. Research on cloud-native stream processing has also shown that event-driven applications need window scheduling and priority routing [32].

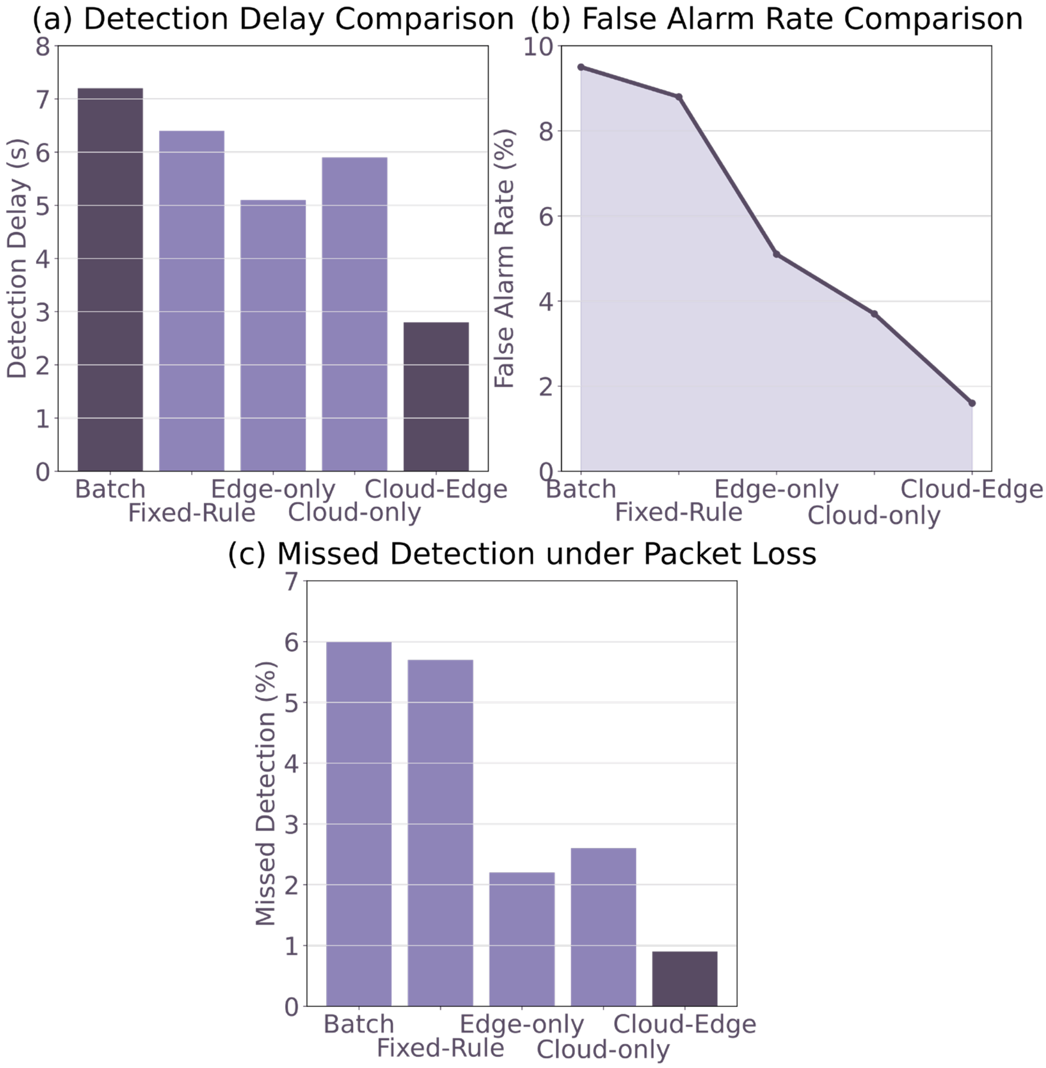

Figure 4. Traffic event detection performance under different scenarios. (a) Detection delay comparison. (b) False alarm rate comparison. (c) Missed detection under packet loss

The average vehicle delay, the length of the line, and the smoothness of the green-time adjustment are all considered aspects of traffic control responsiveness. The flow control analysis findings are displayed in Figure 5. The mean intersection delay is shown in Figure 5(a), the queue length reduction is shown in Figure 5(b), and the signal adjustment stability is shown in Figure 5(c). After implementing the new method, the average delay and queue length decreased by 18.6% and 22.4%, respectively, in comparison to the fixed-rule scheduling. The computation index and the transportation outcomes of decisions should be considered in real time, according to traffic analysis research [33].

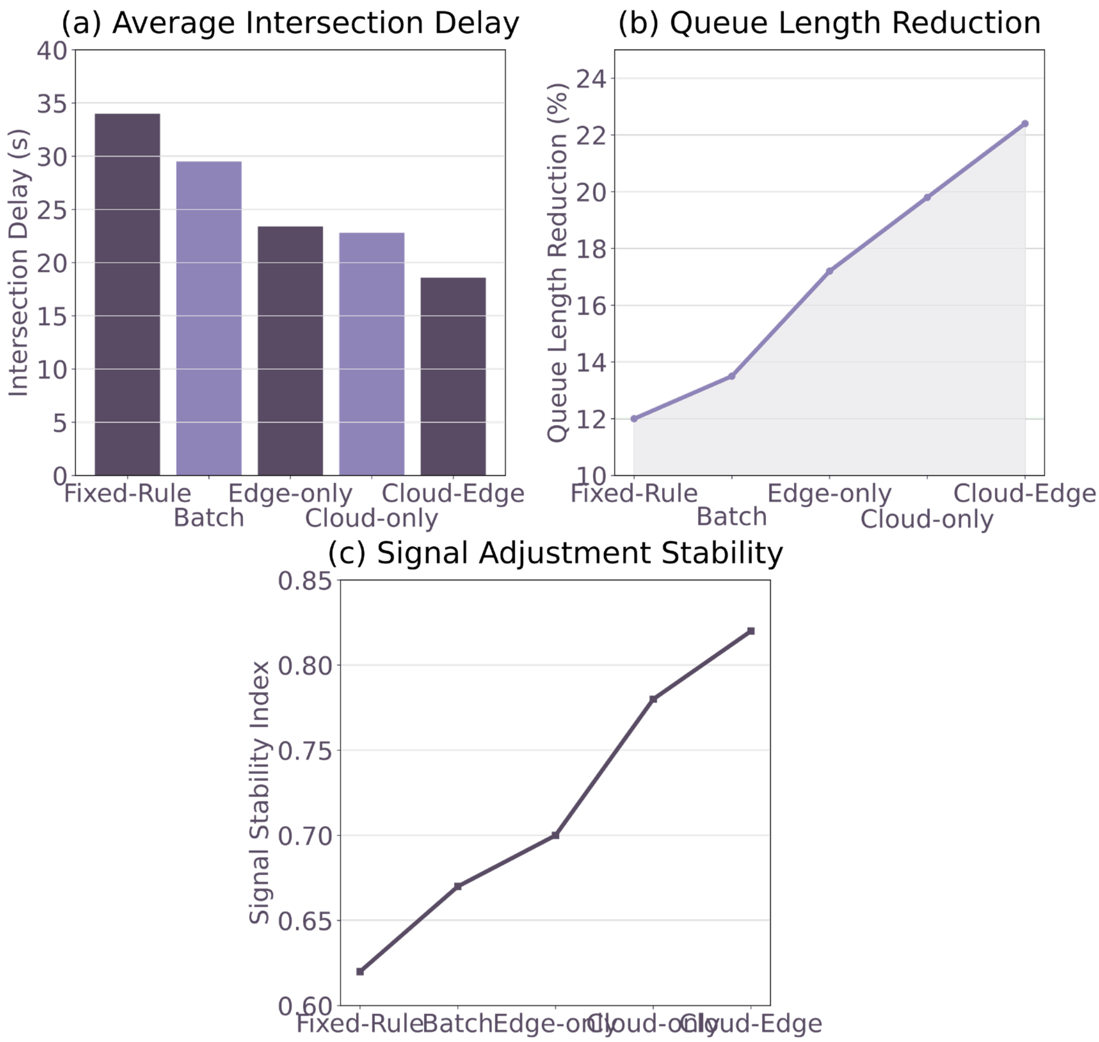

Figure 5. Traffic flow control response analysis. (a) Average intersection delay. (b) Queue length reduction. (c) Signal adjustment stability

Scalability and Deployment Efficiency Discussion

According to the scale test, the suggested system can continue to function normally when the number of junctions increases from 12 to 48. The scalability findings are displayed in Figure 6, where Figure 6(a) compares processing delay under various network sizes, Figure 6(b) reports CPU and memory use, and Figure 6(c) exhibits elastic operator expansion. Adaptive resource allocation is required for geographically non-uniform stream loads, according to research on distributed data processing [34].

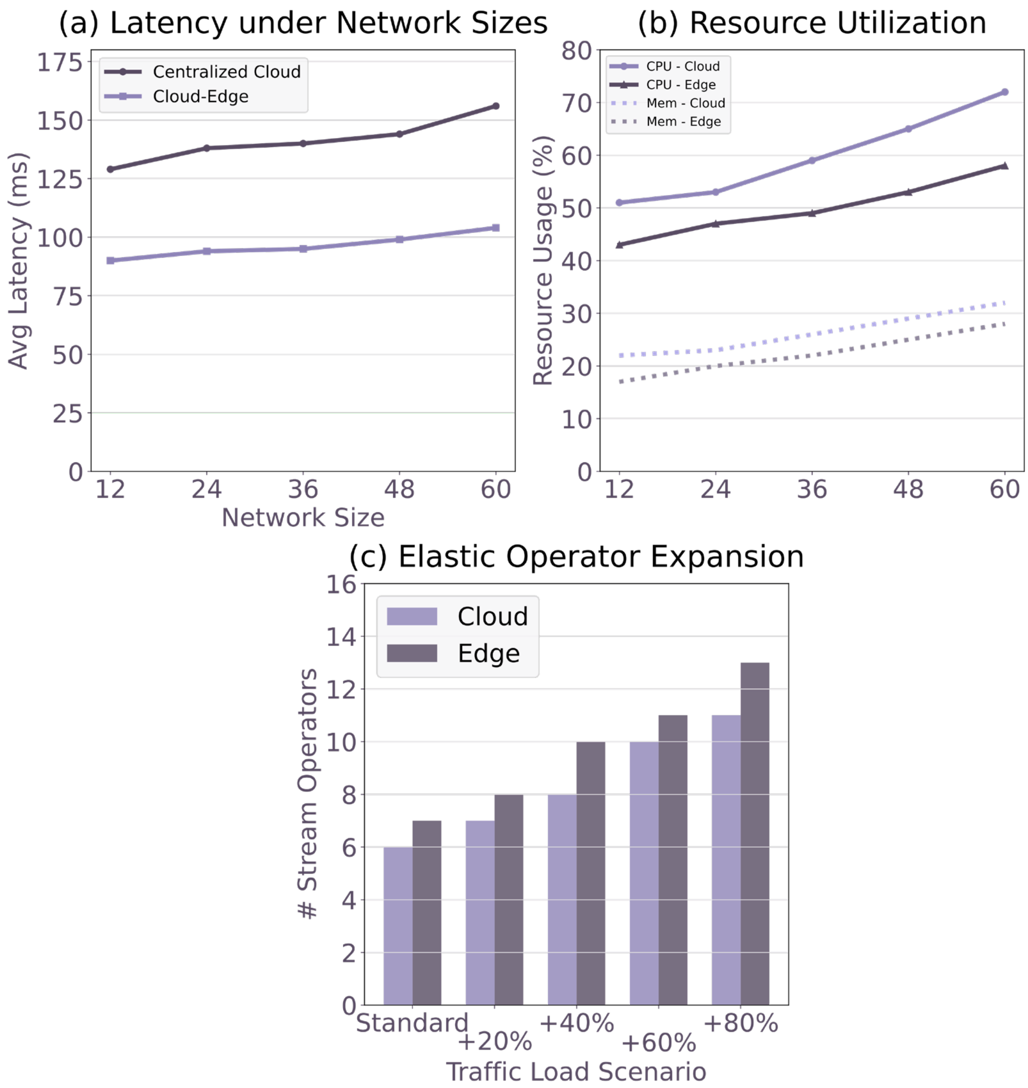

Figure 6. Scalability analysis of cloud-edge traffic processing. (a) Latency under different network sizes. (b) Resource utilization comparison. (c) Elastic operator expansion

Additionally, packet loss, cloud-node overload, and edge-node failure will be broken down by deployment efficiency. Figure 7 illustrates deployment robustness; Figure 7(a) depicts the outcome when packet loss occurs, Figure 7(b) contrasts the reaction under cloud overload, and Figure 7(c) illustrates the recovery following an edge failure. In the case of a single-edge failure, throughput drop is still less than 9.3%, with an average recovery time of 1.7 seconds. Resilient orchestration requires traffic systems with safety-critical decision-making skills, according to edge intelligence research [35]. Priority scheduling for burst events is also supported by real-time stream processing literature [36]. Research on cloud-native microservices has demonstrated that under dynamic loads, container-level scaling may preserve service continuity [37]. Recent studies on intelligent transportation have demonstrated that, in situations when sensor coverage is relatively low, the accuracy of traffic-state information can be increased by merging data from many sources [38]. A comparatively little increase in processing time can be exploited to shorten a vehicle backlog, according to research on urban traffic control [39]. As a result, the automated traffic management system may provide a framework for adaptive scheduling, cloud scalability, and edge responsiveness [40].

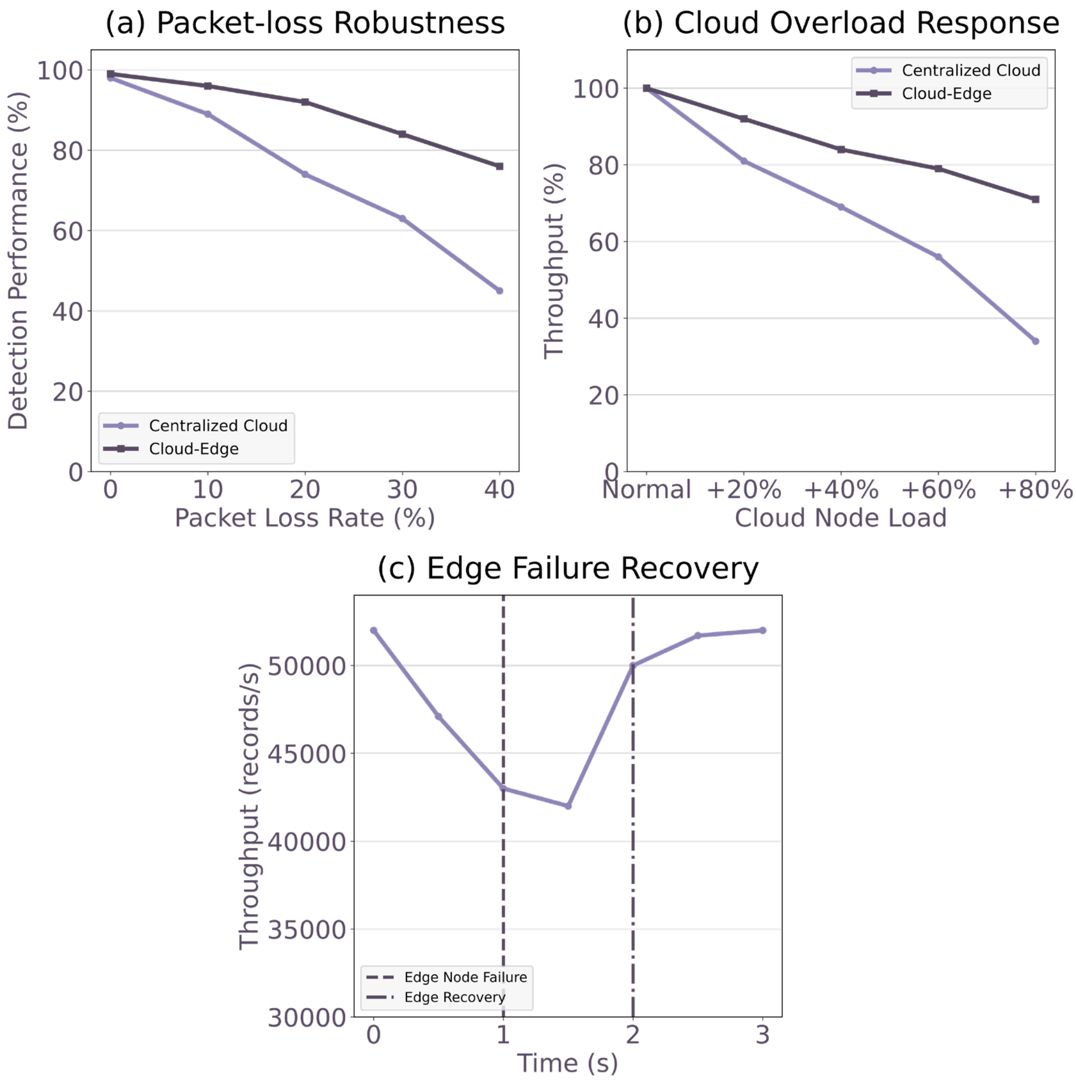

Figure 7. Deployment robustness under traffic management disturbances. (a) Packet-loss robustness. (b) Cloud overload response. (c) Edge failure recovery

Conclusion

This research presented a cloud-based real-time data processing architecture for an automated traffic control system. The system performs low-latency event response, combines traffic state data from many sources, integrates roadside sensing and edge pre-processing, and performs cloud-stream processing. It can simultaneously handle several types of traffic, grow in size, and function rapidly.

According to the findings, cloud-edge collaboration can improve stream and traffic event detection throughput and lower end-to-end latency when compared to centralized cloud processing and edge-only processing. Flexible scheduling will prevent lines from building up during periods of high traffic and incidents, and several sources of the event will boost its credibility.

Future studies can incorporate data from connected vehicles, signal control based on reinforcement learning, digital twin traffic simulation, and data sharing that protects privacy. Large urban transportation networks can also benefit from federated edge-cloud optimization and hardware-aware deployment.

Author Contributions

Grażyna Grabiec and Sławomir Cyra contribute to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. Jarosław Bogdan Kalisz contributes to data collection, draft preparation, manuscript editing. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Xie, B., Zhang, K., Zhao, Y., Zhang, Y., Cai, Y., & Wang, T. (2021). Self-adaptive trajectory prediction for improving traffic safety in cloud-edge based transportation systems. Journal of Cloud Computing, 10(1). https://doi.org/10.1186/s13677-020-00220-8

[2] Chinamanagonda, S. (2020). Edge Computing: Extending the Cloud to the Edge -Growth in IoT and Real-Time Data Processing Needs. International Journal of Science and Research (IJSR), 9(2), 1941-1949. https://doi.org/10.21275/sr24829170402

[3] Volkov, A. O., Korobkina, A. V., & Stepanov, S. N. (2022). Development of a Model and Algorithms for Servicing Real-Time and Data Traffic in a Cloud Computing System. 2022 Systems of Signals Generating and Processing in the Field of on-Board Communications, 1-5. https://doi.org/10.1109/ieeeconf53456.2022.9744289

[4] Wang, R., Mou, X., Sun, J., Liu, P., Guo, X., Wo, T., & Liu, X. (2020). Cloud-Edge Collaborative Industrial Robotic Intelligent Service Platform. 2020 IEEE International Conference on Joint Cloud Computing, 71-77. https://doi.org/10.1109/jcc49151.2020.00020

[5] Lu, S., Tang, X., Zhu, Y., & She, J. (2021). A Cloud-Edge Collaborative Intelligent Fault Diagnosis Method Based on LSTM-VAE Hybrid Model. 2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), 207-212. https://doi.org/10.1109/cscloud-edgecom52276.2021.00045

[6] Hou, X., & Zhao, H. (2021). Edge-cloud Collaborative Architecture for Scene Graph Application System. 2021 6th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), 299-304. https://doi.org/10.1109/iciibms52876.2021.9651576

[7] Du, B., Shan, N., & Zhou, S. (2019). An Efficient Mobile Cloud Service Model or Tactical Edge. Advances in Intelligent Systems and Computing, 422-430. https://doi.org/10.1007/978-3-030-29035-1_41

[8] Koutsovasilis, P., Venugopal, S., Gkoufas, Y., & Pinto, C. (2021). A Holistic Approach to Data Access for Cloud-Native Analytics and Machine Learning. 2021 IEEE 14th International Conference on Cloud Computing (CLOUD), 654-659. https://doi.org/10.1109/cloud53861.2021.00084

[9] Shaohua, H., & Nanfeng, X. (2020). Abnormal Traffic Monitoring Methods Based on a Cloud Computing Platform. 2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), 85-89. https://doi.org/10.1109/icccbda49378.2020.9095592

[10] Manchana, R. (2022). The Power of Cloud-Native Solutions for Descriptive Analytics: Unveiling Insights from Data. Journal of Artificial Intelligence & Cloud Computing, 1(4), 1. https://doi.org/10.47363/jaicc/2022(1)e139

[11] Han, R., & Zhang, C. (2019). Big Data Analysis on Economical Urban Traffic in Beijing: Organize overlapping transportation though the underground diameter line of Beijing railway hub. 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), 269-273. https://doi.org/10.1109/icccbda.2019.8725775

[12] Ozpinar, A., & Yarkan, S. (2019). Vehicle to Cloud. Web Services, 1223-1242. https://doi.org/10.4018/978-1-5225-7501-6.ch063

[13] Banihashemi, S., & Sohi, S. Z. (2022). Big Data in Urban Traffic and Transportation. Data-centric Regenerative Built Environment, 57-91. https://doi.org/10.4324/9781003139942-3

[14] Nampalli, R. C. R. (2021). Leveraging AI in Urban Traffic Management: Addressing Congestion and Traffic Flow with Intelligent Systems. Journal of Artificial Intelligence and Big Data, 1(1), 86-99. https://doi.org/10.31586/jaibd.2021.1151

[15] Yu, C., Chen, J., & Xia, G. (2022). Coordinated Control of Intelligent Fuzzy Traffic Signal Based on Edge Computing Distribution. Sensors, 22(16), 5953. https://doi.org/10.3390/s22165953

[16] Yang, J., & Suo, H. (2020). Intelligent Traffic Signal Control System Based on Edge Computing and Big Data. Journal of Physics: Conference Series, 1624(3), 032039. https://doi.org/10.1088/1742-6596/1624/3/032039

[17] Celtek, S. A., & Durdu, A. (2022). A Novel Adaptive Traffic Signal Control Based on Cloud/Fog/Edge Computing. International Journal of Intelligent Transportation Systems Research, 20(3), 639-650. https://doi.org/10.1007/s13177-022-00315-3

[18] Songlin Chen, S. C., Songlin Chen, H. W., & Hong Wen, J. W. (2022). Artificial Intelligence Based Traffic Control for Edge Computing Assisted Vehicle Networks. 網際網路技術學刊, 23(5), 989-996. https://doi.org/10.53106/160792642022092305007

[19] Kumar, D. (2022). Fog Computing and Vehicular Networks for Smart Traffic Control. Bio-Inspired Optimization in Fog and Edge Computing Environments, 139-146. https://doi.org/10.1201/9781003322931-8

[20] Yang, J. (2020). Design Traffic Signal Node Based on Edge Computing. Journal of Physics: Conference Series, 1575(1), 012198. https://doi.org/10.1088/1742-6596/1575/1/012198

[21] Sharma, A., & Diwakar, C. (2021). Future aspects on MEC (Mobile Edge Computing): Offloading Mechanism. 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), 34-39. https://doi.org/10.1109/ispcc53510.2021.9609511

[22] Nie, W., Duan, Y., Lee, V. C., & Liu, K. (2019). Supporting Data Stream Analytical Processing in Vehicular Sensor Networks. 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2127-2134. https://doi.org/10.1109/itsc.2019.8917225

[23] Lu, M., Xiong, C., & Fan, X. (2020). Erasure-Code Algorithm for Distributed Stream Processing. 2020 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS), 1-5. https://doi.org/10.1109/hpbdis49115.2020.9130569

[24] Weißbach, M., Hilbert, H., & Springer, T. (2020). Performance Analysis of Continuous Binary Data Processing using Distributed Databases within Stream Processing Environments. Proceedings of the 10th International Conference on Cloud Computing and Services Science, 138-149. https://doi.org/10.5220/0009413301380149

[25] Li, K., Liu, G., & Lu, M. (2019). A Holistic Stream Partitioning Algorithm for Distributed Stream Processing Systems. 2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), 202-207. https://doi.org/10.1109/pdcat46702.2019.00046

[26] K, S. (2020). Dynamic Operator Scaling for Distributed Stream Processing Systems for Fluctuating Streams. Journal of Advanced Research in Dynamical and Control Systems, 12(SP7), 2815-2821. https://doi.org/10.5373/jardcs/v12sp7/20202422

[27] Neto, A., Silva, T., Batista, T., Lopes, F., Delicato, F., & Pires, P. (2021). Optimizing Resource Allocation in Edge-distributed Stream Processing. Proceedings of the 17th International Conference on Web Information Systems and Technologies, 156-166. https://doi.org/10.5220/0010714700003058

[28] Dong, Z. (2022). Prediction of Ship Traffic Flow Based on Wavelet Decomposition and LSTM. 2022 7th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), 88-93. https://doi.org/10.1109/icccbda55098.2022.9778881

[29] Liu, J., Qu, H., Chen, M., & Gong, Y. (2022). Traffic Flow Prediction Based on Spatio-temporal Attention Block. 2022 International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM), 32-39. https://doi.org/10.1109/mlccim55934.2022.00014

[30] Liu, F., Ma, X., An, X., & Liang, G. (2020). Urban Traffic Flow Prediction Model with CPSO/SSVM Algorithm under the Edge Computing Framework. Wireless Communications and Mobile Computing, 2020, 1-11. https://doi.org/10.1155/2020/8871998

[31] Li, X. (2022). Research on Traffic Intelligence based on Traffic Flow Detection Algorithm. 2022 International Conference on Edge Computing and Applications (ICECAA), 1525-1529. https://doi.org/10.1109/icecaa55415.2022.9936548

[32] Aljulayfi, A., & Djemame, K. (2021). A Machine Learning based Context-aware Prediction Framework for Edge Computing Environments. Proceedings of the 11th International Conference on Cloud Computing and Services Science, 143-150. https://doi.org/10.5220/0010379001430150

[33] Ma, C., Hu, X., Liu, S., & Liu, L. (2021). Attention Based Multi-Unit Spatial-Temporal Network for Traffic Flow Forecasting. 2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), 225-230. https://doi.org/10.1109/cscloud-edgecom52276.2021.00048

[34] Ma, C., Qin, S., Hu, J., & Yan, L. (2021). Subway Flow Prediction Based on Improved Support Vector Machine. 2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), 115-120. https://doi.org/10.1109/cscloud-edgecom52276.2021.00030

[35] Sahil, & Sood, S. K. (2021). Smart vehicular traffic management: An edge cloud centric IoT based framework. Internet of Things, 14, 100140. https://doi.org/10.1016/j.iot.2019.100140

[36] G, D. K. (2022). Intelligent Traffic Management Using Internet of Vehicles: A Study into Opportunities and Challenges. Technoarete Transactions on Internet of Things and Cloud Computing Research, 2(2). https://doi.org/10.36647/ttitccr/02.02.art003

[37] Anchal, Mittal, P., Gulia, P., & Balkishan (2022). Applications of Machine Learning Techniques in SD-IoT Traffic Management. SDN-Supported Edge-Cloud Interplay for Next Generation Internet of Things, 171-200. https://doi.org/10.1201/9781003213871-9

[38] Absardi, Z. N., & Javidan, R. (2022). A QoS-Aware Traffic Management Policy for IoT-enabled Smart City Applications based on Edge Cloud Computing. 2022 Sixth International Conference on Smart Cities, Internet of Things and Applications (SCIoT), 1-7. https://doi.org/10.1109/sciot56583.2022.9953660

[39] Mohanty, S., Sahoo, B., Apat, H. K., & Sahoo, K. S. (2022). An Overview of Software-Defined Internet of Vehicles. SDN-Supported Edge-Cloud Interplay for Next Generation Internet of Things, 71-86. https://doi.org/10.1201/9781003213871-4

[40] Duque, A. O., Klein, C., Feng, J., Cai, X., Skubic, B., & Elmroth, E. (2022). A Qualitative Evaluation of Service Mesh-based Traffic Management for Mobile Edge Cloud. 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid), 210-219. https://doi.org/10.1109/ccgrid54584.2022.00030