frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 17 September 2025
DOI 10.3389/fcomp.2025.1564270

OPEN ACCESS

EDITED BY
Yuanyuan Huang,
Chengdu University of Information Technology, China

REVIEWED BY
Richard Kotter,
Northumbria University, United Kingdom
Jiazhong Lu,
Chengdu University of Information Technology, China
Olfa Souki,
University of Sfax, Tunisia

*CORRESPONDENCE
Afzal Badshah
✉ afzalbadshahkhattak@gmail.com
Ali Daud
✉ alimsdb@gmail.com

RECEIVED 21 January 2025
ACCEPTED 27 August 2025
PUBLISHED 17 September 2025

CITATION
Badshah A, Alsahfi T, Alesawi S, Alfakeeh A, Bukhari A and Daud A (2025) Regional computing for VBD offloading in next-generation vehicular networks. *Front. Comput. Sci.* 7:1564270. doi: 10.3389/fcomp.2025.1564270

COPYRIGHT
© 2025 Badshah, Alsahfi, Alesawi, Alfakeeh, Bukhari and Daud. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Regional computing for VBD offloading in next-generation vehicular networks

Afzal Badshah[1]*, Tariq Alsahfi[2], Sami Alesawi[3], Ahmed Alfakeeh[4], Amal Bukhari[2] and Ali Daud[5]*

[1]Department of Software Engineering, University of Sargodha, Sargodha, Pakistan, [2]Department of Information Systems and Technology, College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia, [3]Department of Computer Science, Faculty of Computing and Information Technology in Rabigh, King Abdulaziz University, Rabigh, Saudi Arabia, [4]Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia, [5]Faculty of Resilience, Rabdan Academy, Abu Dhabi, United Arab Emirates

The rapid growth of autonomous and Connected Vehicles (CVs) has led to a massive increase in Vehicular Big Data (VBD). While this data is transforming the Intelligent Transportation System (ITS), it also poses significant challenges in processing, communication, and resource scalability. Existing cloud solutions offer scalable resources; however, incur long delays and costs due to distant data communication. Conversely, edge computing reduces latency by processing data closer to the source; however, struggles to scale with the high volume and velocity of VBD. This paper introduces a novel Regional Computing (RC) paradigm for VBD offloading, with a key focus on adapting to traffic variations during peak and off-peak hours. Situated between edge and cloud layers, the RC layer enables near-source processing while maintaining higher capacity than edge or fog nodes. We propose a dynamic offloading algorithm that continuously monitors workload intensity, network utilization, and temporal traffic patterns to smartly offload tasks to the optimal tier (vehicle, regional, or cloud). This strategy ensures responsiveness across fluctuating conditions while minimizing delay, congestion, and energy consumption. To validate the proposed architecture, we develop a custom Python-based simulator, RegionalEdgeSimPy, specifically designed for VBD scenarios. Simulation results demonstrate that the proposed framework significantly reduces processing latency, energy usage, and operational costs compared to traditional models, offering a scalable and effective alternative for next-generation vehicular networks.

KEYWORDS

vehicular big data, regional computing, network optimization, intelligent transportation systems, edge computing

1 Introduction

The rapid evolution of autonomous and CVs technologies has led to a substantial increase in data generation. This proliferation is driven by the deployment of high-resolution video cameras for visual scene analysis, LiDAR for 3D environmental mapping, radar for long-range object detection, ultrasonic sensors for short-distance awareness, and GPS modules for precise localization (Lee et al., 2023). The resulting data surge, widely recognized as *VBD*, shown in Figure 1, forms the backbone of modern ITS. While VBD offers transformative benefits for vehicular safety, automation, and decision, it also presents serious challenges in terms of delay, communication bandwidth, and system scalability.

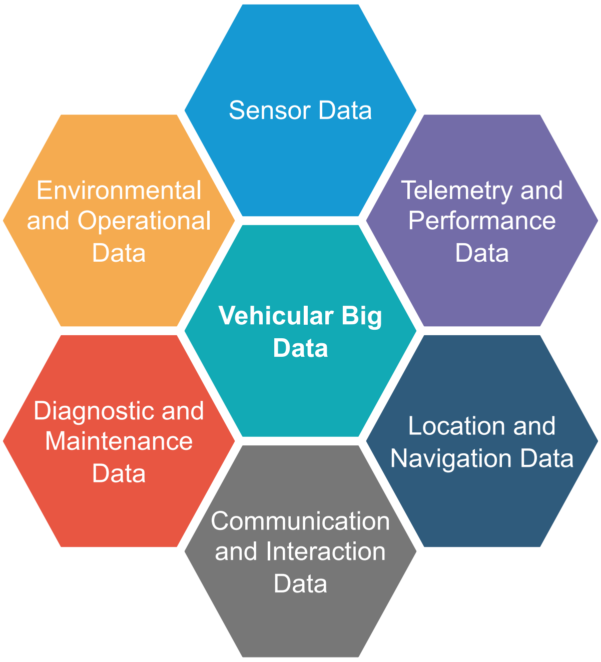

Sensor Data
Environmental and Operational Data
Telemetry and Performance Data
Vehicular Big Data
Diagnostic and Maintenance Data
Location and Navigation Data
Communication and Interaction Data

FIGURE 1
Classification of vehicular big data.

Autonomous Vehicles (AVs) are forecasted to reach widespread adoption by 2030 (Statista, 2024), with each vehicle generating as much as 4 TB of data per day (Mordorintelligence, 2024). This data supports crucial operations such as real-time sensing, perception, planning, and inter-vehicular coordination (Padmaja et al., 2023; Khan et al., 2024). As Figure 2 illustrates, the market growth for AVs is accelerating rapidly. Industry leaders, including Tesla,[1] Waymo,[2] Motional,[3] and Ford[4] are actively pushing forward developments in autonomous driving, supported by integrated sensing, onboard intelligence, and data-driven strategies. For clarity, Table 1 summarizes the acronyms used throughout this paper.

Cloud Computing (CC) provides the necessary scalability and computational capacity for big data storage, analysis, and collaborative learning (Badshah et al., 2022a, 2021; Alharbey et al., 2024). However, its reliance on distant servers incurs high latency and substantial communication overhead (Kumer et al., 2021; Waqas et al., 2024), limiting its effectiveness for safety-critical AVs applications. In contrast, Edge Computing (EC) brings processing closer to data sources, reducing transmission time and response latency (Garg et al., 2019). However, EC suffers from limited computational power and scalability, making it insufficient for handling continuous VBD workloads, particularly under peak-hours conditions (Ometov et al., 2022). To further address latency concerns, Mobile Edge Computing (MEC) has been introduced as an enhancement to EC, deploying compute resources at base stations within the radio access network. While MEC offers improved responsiveness for time-sensitive vehicular tasks, it still faces challenges such as narrow geographic coverage, high deployment costs per site, and the lack of coordination across broader regions (Zhang et al., 2024). These shortcomings in both EC and MEC necessitate the introduction of a more balanced architectural solution (Badshah et al., 2024).

To overcome these limitations, we propose a *RC* framework that integrates an intermediate computing tier between EC and CC. While fog computing is a middle-tier approach, it is deployed within or near edge facilities, sharing the same premises and constrained in resources. In contrast, RC servers are deployed regionally, spanning city or national scales, and possess enhanced storage and processing capabilities. This makes RC more suitable for managing the scale and dynamics of VBD in next-generation vehicular networks. As illustrated in Figure 3, the RC layer facilitates near-source processing during peak hours and enables delayed offloading of non-critical data to the cloud during offpeak hours, thereby improving bandwidth usage and minimizing congestion (Badshah et al., 2022b).

At the core of this framework is a *dynamic, traffic-aware offloading strategy* that adapts in real-time to workload characteristics, task urgency, network congestion, and usage patterns. This approach ensures time-sensitive processing is prioritized close to the vehicle, while deferrable data is transmitted during periods of reduced network activity.

To validate the proposed approach, we present *RegionalEdgeSimPy*, a Python-based simulator specifically designed for vehicular big data environments. Unlike existing tools such as EdgeCloudSim (Sim, 2024), our simulator models vehicular mobility, dynamic workload generation, congestion levels, and time-sensitive offloading behavior, enabling detailed performance analysis for delay, energy consumption, and cost efficiency.

The key contributions of this paper are as follows:

- We propose a novel RC framework for next-generation vehicular networks, bridging the performance gaps between traditional EC and CC systems.
- We design a *dynamic offloading strategy* that responds to temporal traffic patterns such as peak and off-peak hours, task characteristics, and system conditions to optimize workload placement.
- We develop and utilize *RegionalEdgeSimPy*, a customized simulator for evaluating vehicular big data offloading across multi-tier architectures.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 presents the proposed framework and methodologies. Section 4 provides simulation setup and evaluation results. Section 5 discusses implementation challenges, and Section 6 concludes the paper with directions for future work.

2 Related work

The rise of connected and autonomous vehicles has led to an exponential increase in VBD, intensifying challenges related to

[1] https://www.tesla.com/

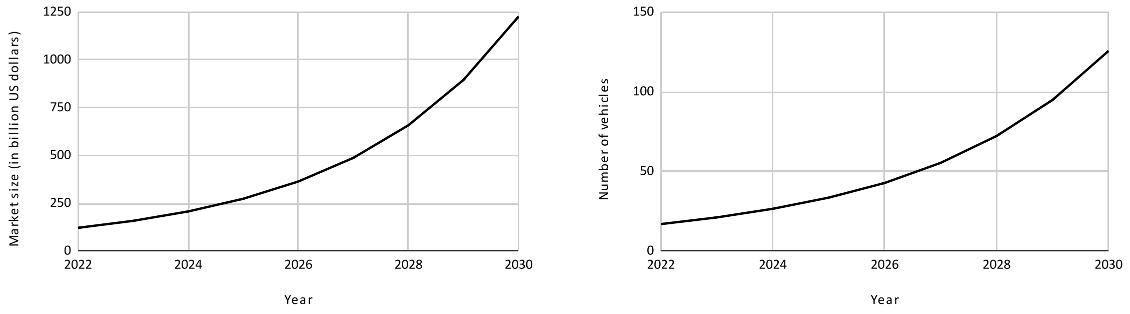
[2] https://waymo.com/

[3] https://motional.com/

[4] https://ford.com/

TABLE 1 Acronyms, notations, and symbols used in this article.

Symbol	Representation	Symbol	Representation
CC	Cloud computing	EC	Edge computing
RC	Regional computing	DC	Data center
ITS	Intelligent transport system	AV	Autonomous vehicle
VBD	Vehicular big data	ADAS	Advanced driving assistant system
GPS	Global positioning system	BS	Base station
RSU	Roadside unit	IoV	Internet of vehicles
VANET	Vehicular *ad hoc* network	DCC	Decentralized congestion control
VM	Virtual machine	MIPS	Million instructions per second
SINR	Signal-to-interference-plus-noise ratio	D_{net}	Networking delay ($D_{tran} + D_{prop} + D_{hovr}$)
D_t	End-to-end delay	D_{prop}	Propagation delay
D_{tran}	Transmission delay	D_{que}	Queuing delay
D_{proc}	Processing delay	D_{hovr}	Handover delay
Con_{Prob}	Congestion probability	θ	Congestion threshold
W	Measurement window (100 ms)	busy(W)	Busy time in window W
W_s	Stability window	ε	Utilization variance threshold
δ_{net}	Delay budget threshold	B_{avail}	Available bandwidth
C_{reg}	Regional server capacity	R_{T_i}	Resource demand of task T_i
L_{total}	Total load in window	P_i	Power at stage i
$T_{i,i+1}$	Transmission time between stages i and $i+1$	$D_{i,i+1}$	Distance between stages i and $i+1$
$E_{tran}(i, i+1)$	Transmission energy between stages i and $i+1$	E_{tran}	Total transmission energy
R	Acceptable signal quality	V	Vehicle velocity



(a) Autonomous vehicles market size

(b) Autonomous vehicle forecast

FIGURE 2
Autonomous vehicle statistics (Statista, 2024). **(a)** Autonomous vehicles market size. **(b)** Autonomous vehicle forecast.

its real-time transmission, processing, and storage. While many existing studies focus on big data analytics, comparatively fewer address the infrastructural burden of offloading and managing this massive data (Ramirez-Robles et al., 2024; Amin et al., 2024). With vehicles generating as much as 4 TB of data per day (Lee et al., 2023), including sensor feeds, telemetry, and environment perception data, this information must be processed onboard, shared with Roadside Units (RSUs), and transmitted to the cloud for advanced analysis and navigation support (Amin et al., 2024).

Recent studies explore the role of the Internet of Vehicles (IoV) in managing VBD by enabling scalable data collection, adaptive protocol design, and traffic-aware resource utilization. For instance, Xu et al. (2018) highlights how IoV enhances performance optimization through real-time data exchange. Authors in Daniel et al. (2017) emphasize the classification and prioritization of

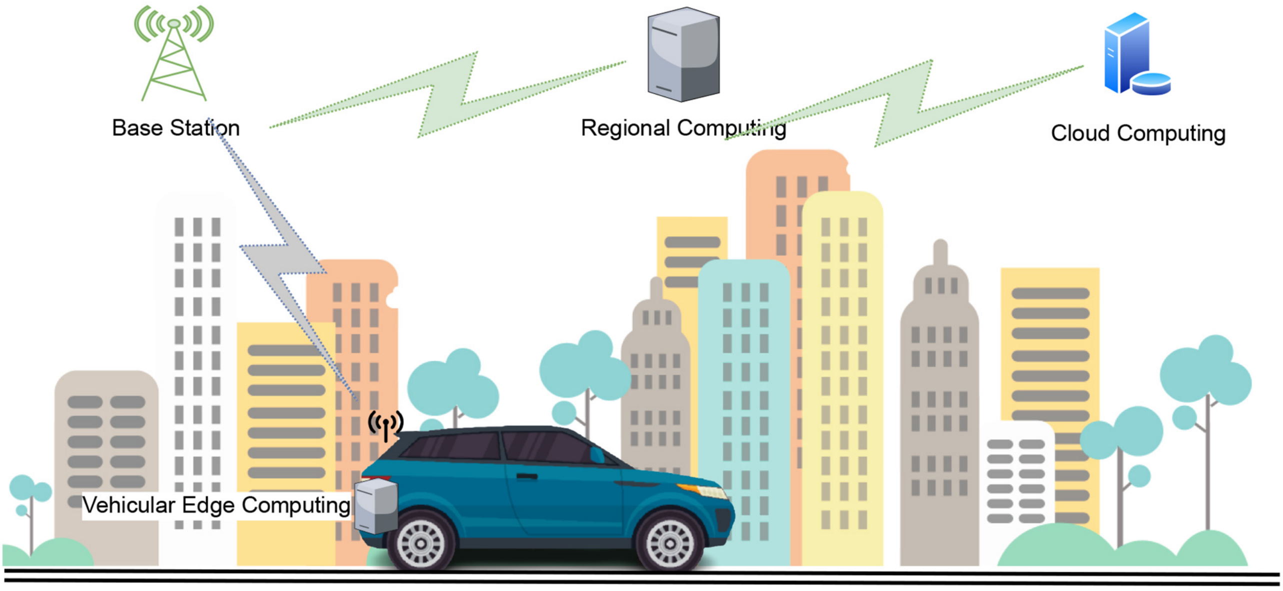


FIGURE 3
Regional computing for Vehicular Big Data (VBD).

vehicular data streams to ensure low-latency responsiveness in safety-critical scenarios.

Vehicular communication often depends on Vehicular *Ad Hoc* Networks (VANETs), which, despite their low-latency design, face challenges when scaling to dense traffic or data-intensive use cases. To address this, machine learning analysis is used in Manikandan et al. (2025) to detect unfavorable communication scenarios. Alternatively, Murk et al. (2019) explores a vehicle-assisted model for data offloading that reduces dependence on network infrastructure and lowers environmental impact.

To address network bottlenecks and reduce transmission delays, distributed and edge computing models have been investigated. The work in Alexakis et al. (2022) proposes a layered architecture supporting distributed processing and cloud analytics for vehicular data. Similarly, Garg et al. (2019) utilizes edge nodes deployed alongside road infrastructure, using 5G networks to minimize cloud latency. Similarly, authors in Rajput et al. (2023) explore utilizing idle or parked vehicles as temporary edge processors to decentralize computing tasks.

Distributed computing frameworks have also emerged as viable strategies for scaling vehicular data operations. For example, Sivakumar et al. (2025) proposes a hybrid edge and cloud model that adaptively distributes computational load based on network congestion and data proximity. Similarly, Chen et al. (2021) introduces interconnected systems for dynamic resource sharing among vehicles to reduce service latency and enhance responsiveness.

MEC extends CC capabilities to the edge of mobile networks by placing resources at the base stations. In vehicular settings, MEC supports low-latency offloading by enabling vehicles to connect with nearby servers through the radio access network However, MEC's limited coverage and high deployment cost per site constrain its scalability (Santos et al., 2025). Recent studies propose online MEC offloading for Vehicle to Vehicle (V2V) systems to improve responsiveness under dynamic traffic. These limitations underscore the need for a broader layer like RC, which enables coordinated processing across regions and facilitates off-peak cloud transfer while maintaining responsiveness (Santos et al., 2025).

Despite these advancements, researchers increasingly acknowledge the limitations of relying solely on edge or cloud paradigms. For example, Prehofer and Mehmood (2020) observes that high vehicle mobility often causes vehicles to move beyond the communication range of edge servers, disrupting data continuity. In response, Rajput et al. (2023) proposes a cooperative approach using parked vehicles as relay units or temporary RSUs to process data locally. While effective to some extent, such approaches still lack the scalability and coordination of a centralized system. This further reinforces the need for a robust intermediary layer like regional computing.

Emerging wireless technologies offer new possibilities for real-time Vehicle to Everything (V2X) communication. In particular, 5G and future 6G networks are seen as essential for latency-sensitive applications, such as those in Cellular Vehicle-to-Everything (C-V2X) environments (He et al., 2020). While 5G currently enables improved responsiveness through enhanced bandwidth and network slicing, 6G is expected to introduce unprecedented capabilities, including ultra-reliable low-latency communication (URLLC), intelligent resource orchestration, and integration with AI-driven decision-making. These features will further strengthen the effectiveness of regional computing by enabling faster task distribution, optimized network utilization, and seamless coordination across edge, regional, and cloud layers. Similarly, Liang et al. (2025) explores the use of THz communication bands to handle high-throughput data sharing between vehicles, while Du et al. (2020) examines the role of millimeter waves in boosting transmission capacity.

Security and privacy concerns also feature prominently in recent research. Study Liu et al. (2024) outlines security protocols for IoV big data exchange, while Guo et al. (2017) suggests a cloud model that authenticates vehicles before allowing data exchange. Additionally, Nan et al. (2023) proposes a regional cloud system in which computing resources are registered in a centralized index to ensure optimal access across the vehicular network.

Despite these contributions, existing literature seldom considers the temporal dynamics of data offloading, such as the distinction between peak and off-peak traffic periods. There is limited exploration of systems that adaptively shift workload placement based on real-time congestion and task criticality. Furthermore, simulation tools like EdgeCloudSim (Sim, 2024) lack capabilities to model vehicular mobility and time-sensitive task distribution, limiting their applicability for next-generation vehicular networks. To fill these gaps, this work introduces a novel RC framework paired with a dynamic, traffic-aware offloading algorithm. To support experimentation and benchmarking, a custom simulation environment, *RegionalEdgeSimPy*, has also been developed. Unlike prior tools, this simulator captures the temporal, spatial, and workload variability inherent in vehicular big data systems, offering a scalable foundation for evaluating offloading strategies under realistic operating conditions.

3 Proposed model

The ITS becomes smarter as data from each vehicle is added to the joint system. Uploading every vehicle's data to the cloud servers is necessary. However, every vehicle may generate one Terabyte (TB) of data per day, which is impossible for the existing system to transport, process, and store (Badshah et al., 2023). As shown in Figure 4, the proposed system aims to handle the VBD near the road infrastructure to minimize the delay and cost, and subsequently transfer this data to the cloud in off-peak hours. This will cause a minimum load on the network, as we usually see that the network gets congested in peak hours and underutilized in off-peak hours.

The proposed method works on three layers: the vehicular Edge Computing Layer, the Regional Computing Layer, and the Cloud Computing Layer.

The edge computing layer is the vehicle's server processing the sensor data in real-time. Regional computing layer processes and stores the VBD before transferring it to the cloud. The cloud computing layer stores this big data and coordinates among all vehicles.

3.1 Vehicular edge layer

The vehicle Layer includes the vehicle's sensors, processing, actuators, and storage system. It also includes the local edge servers and vehicle communication to surrounding vehicles, RSUs, pedestrians, etc. This internal system is utilized during driving for tasks such as obstacle detection and avoidance. The sensors, e.g., Lidar, Radar, Sonar, cameras, and navigation system, give input to the system. Along with the internal system, the surrounding vehicles and RSUs also assist in the vehicle's navigation. The internal computers process the data with powerful algorithms using the already-built data and Artificial Intelligence (AI) and direct the actuators that keep the vehicle on the road.

AVs need 5G internet connections to upload their internal data to the cloud or regional servers, or receive the navigation information, road status, and other particular spots and congestion data from the cloud servers. The vehicle is connected to the base station, communicating its data to the regional layer.

$$D_t = D_{tran} + D_{prop} + D_{proc} + D_{que} + D_{hovr} \quad (1)$$

Where D_{tran} represents the transmission delay, D_{prop} stands for the propagation delay, D_{proc} signifies the data processing delay, D_{que} denotes the queuing delay, and D_{hovr} corresponds to the handover delay.

The transmission delay (D_{tran}) accounts for the time taken to transmit data to the transmission medium. It is influenced by factors such as workload (W), channel bandwidth capacity (B), signal-to-noise ratio (SNR), modulation efficiency (ME), and error rate (ER). The calculation is given by:

$$D_{\mathrm{tran}} = \frac{W \times \mathrm{SNR} \times \mathrm{ME}}{B \times \mathrm{ER}} \quad (2)$$

The propagation delay (D_{prop}) represents the time for data to travel on the transmission medium from source to destination. It depends on distance (Dis) and transmission speed (tr_s), and is calculated as:

$$D_{\mathrm{prop}} = \frac{\mathrm{Dis}}{\mathrm{tr}_s} \quad (3)$$

Data processing delay (D_{proc}) reflects the time taken for the system to process the data. It is determined by the size of the data (S) and the processing rate of the machine (P_r). The formula is:

$$D_{\mathrm{proc}} = \frac{S}{P_r} \quad (4)$$

Queuing delay (D_{que}) indicates the time data spends waiting in a queue for processing. This delay depends on packet length (L), arrival rate of packets (a), and packet processing rate (R). The equation is given by:

$$D_{\mathrm{que}} = \frac{L \times a}{R} \quad (5)$$

Handover delay (D_{hovr}) represents the delay in connecting to the next base station due to signal strength. It is influenced by acceptable signal quality level (R), vehicle velocity (V), and Signal-to-Interference-plus-Noise Ratio ($SINR$). The calculation is expressed as:

$$D_{\mathrm{hovr}} = \frac{R}{V \times \mathrm{SINR}} \quad (6)$$

These individual components collectively contribute to the overall delay (D_t) experienced by the data during its journey to the edge server.

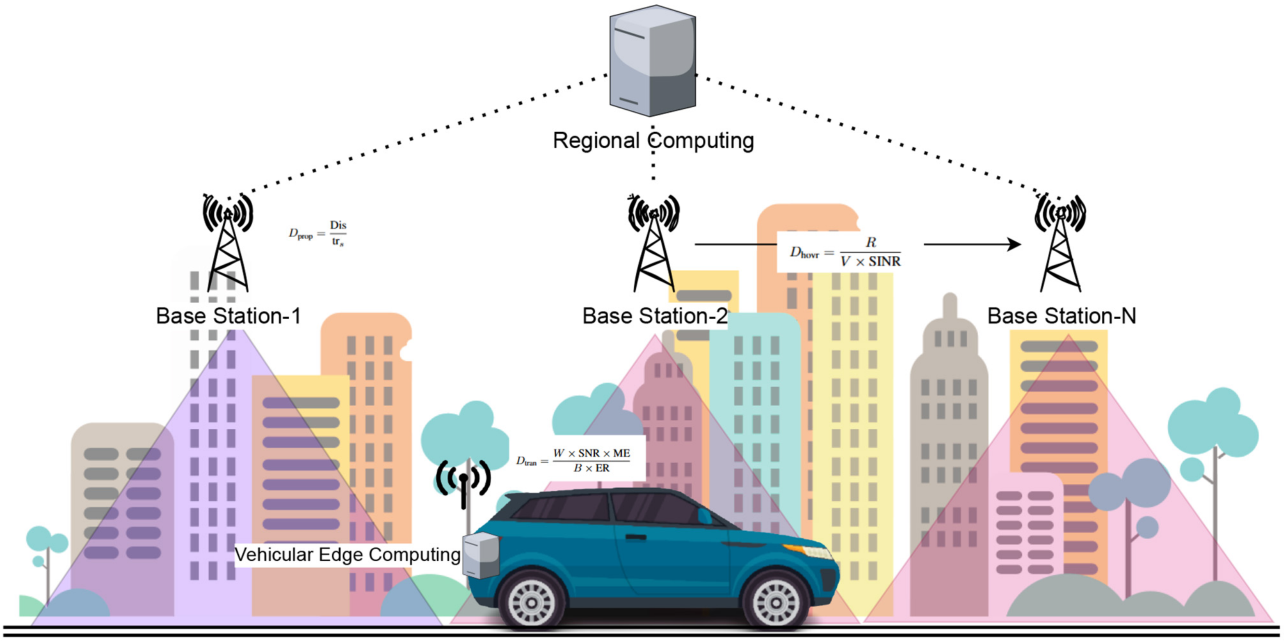

Regional Computing
$D_{\text{prop}} = \frac{\text{Dis}}{\text{tr}_s}$
$D_{\text{hovr}} = \frac{R}{V \times \text{SINR}}$
Base Station-1
Base Station-2
Base Station-N
$D_{\text{tran}} = \frac{W \times \text{SNR} \times \text{ME}}{B \times \text{ER}}$
Vehicular Edge Computing

FIGURE 4
Illustration of the communication delay faced by the VBD across the tiers.

Similarly, data communication cost increases as either the transmission distance or the data size grows;

$$C_t = C_{\text{tran}} + C_{\text{prop}} \tag{7}$$

The equation represents the total cost (C_t) experienced by data to reach the edge server, which is the sum of transmission cost (C_{tran}) and propagation cost (C_{prop}).

3.2 Regional layer

Regional servers within a specific region process and store vehicle data. They act as local hubs for collecting information from vehicles in their vicinity. Each vehicle within the region sends its data to the regional servers. This data may include parameters such as GPS location, speed, acceleration, sensor readings, video, and other relevant information about the vehicle's status and surroundings.

The regional servers process the received vehicle data to extract useful information. This can involve analyzing traffic patterns, road conditions, and congestion levels and identifying potential road issues or hazards. The regional service can provide guidance and information to other regional vehicles based on the processed data. This can include real-time updates about road conditions, traffic congestion, accidents, or any relevant information to help drivers make informed decisions.

The VBD is temporarily stored on this server in peak hours and sent to the cloud in off-peak hours to minimize the congestion on a public network. Furthermore, as we can see from equation 1, the overall delay depends on the distance (Dis) and workload (W). So, if this big data is processed and stored locally in peak hours, this minimizes the total delay, and the vehicle will get a real-time response. Similarly, the public network is not overburdened.

3.3 Cloud layer

The regional layer plays an active role in autonomous driving and real-time operations, while the cloud layer operates passively. The regional layer depends on the cloud for computing resources and transfers its data to the cloud during off-peak hours to leverage the cloud's capabilities for processing and analysis.

Compared to the regional layer, the cloud possesses a vast amount of VBD from AVs worldwide. It serves as a repository for processing and storing this extensive dataset.

$$T_{\text{cloud}} = \frac{W}{R} \tag{8}$$

Equation 8 represents the processing time (T_{cloud}) for collectively processing all vehicular workload at cloud servers. The variables are defined as follows: T_{cloud} is the processing time, W is the total amount of vehicular workload to be processed, and R is the processing rate of the cloud servers, indicating the amount of data processed per unit of time.

The total vehicular workload (W) is calculated as the summation of the amount of data produced by each vehicle (W_i) from 1 to n, as shown in Equation 9:

$$W = \sum_{i=1}^{n} W_i \tag{9}$$

Examining equation number 1 reveals that the roundtrip to the cloud will incur more delay than anticipated. The increased delay is attributed to the cloud having to process a substantial volume of vehicular big data (W). Additionally, the propagation delay and queuing delay experience escalation in tandem with the growing workload and distance.

We can see that cloud analyzes the data to gain insights into traffic patterns, congestion, and road conditions. This information can be used to optimize traffic flow and provide real-time updates to autonomous vehicles regarding alternative routes or potential hazards. The cloud employs machine learning and artificial intelligence algorithms to extract valuable information from the data. This enables development and improvement of autonomous driving algorithms, predictive maintenance models, and other intelligent systems.

The cloud leverages the vehicular data to enhance safety and security measures. It can identify and mitigate potential risks, detect anomalies or malicious activities, and provide early warnings to vehicles and authorities. By analyzing the data, the cloud can identify areas for improvement in Advanced Driving Assistance System (ADAS), such as fuel efficiency, route planning, and vehicle performance. This optimization can lead to cost savings and enhanced overall performance. The cloud's vast dataset is a valuable resource for researchers, engineers, and developers to study and innovate in autonomous driving. It enables the exploration of new algorithms, technologies, and applications to advance the capabilities of autonomous vehicles. This data may also be used to train autonomous driving models and algorithms.

3.4 Energy calculation

Similar to delay, energy consumption is directly proportional to distance, impacting operational costs for data transfer. The energy consumption in the autonomous vehicles' communication environment is computed as follows:

$$E_{\text{tran}} = \sum_{i=1}^{n} E_{\text{tran}}(i, i+1) \tag{10}$$

Here, $E_{\text{tran}}(i, i+1)$ represents the power consumption between consecutive stages, with power consumption increasing as the stages progress.

Where

$$E_{\text{tran}}(i, i+1) = P_i \cdot T_{i,i+1} \tag{11}$$

Where $D_{i,i+1}$ is the distance, P_i is the power and $T_{i,i+1}$ is the time. The above equations show that as the distance increases the energy consumption increases, which is directly proportional to cost.

$$E_{\text{other}} = E_{\text{pro}} + E_{\text{stor}} + E_{\text{col}} + k \tag{12}$$

Similarly, E_{other} represents energy consumption in other activities, including processing (E_{pro}), storing (E_{stor}), and cooling the data centers (E_{col}).

Therefore, the total energy consumption of communication is calculated as follows:

$$E_{\text{total}} = E_{\text{tran}} + E_{\text{other}} \tag{13}$$

Here, E_{total} indicates the total power usage, while E_{tran} denotes the total power consumption on data transfer, encompassing wires, switches, routers, and other devices.

It is also acknowledged that,

$$Cost_{\text{oper}} \propto E \tag{14}$$

The power consumption (E) is directly proportional to the operational cost ($Cost_{\text{oper}}$); hence, operational costs increase with rising power consumption.

```
Input: Vehicular big data D from region's vehicles; network & regional state
Output: Optimal offloading decision (RC, CC)
foreach task $T_i \in D$ do
    if $T_i$ is required for cross-regional training/model sharing then
        Route $T_i$ to cloud;
    end
    else
        Compute $D_{net} \leftarrow D_{tran} + D_{prop} + D_{hovr}$;
        Estimate $Con_{Prob} \leftarrow busy(W)/W$;
        Compute $\overline{util}$ and $Var(util)$ over $W_s$; let $stable \leftarrow (Var(util) \le \varepsilon)$;
        if $(Con_{Prob} \le \theta)$ and $(D_{net} \le \delta_{net})$ then
            if $C_{reg} \ge R_{T_i}$ and stable then
                Process $T_i$ at RC;
            end
            else if Task arrival time ∈ off-peak hours and $C_{reg} \ge R_{T_i}$ then
                Process $T_i$ at RC;
            end
            else
                Offload $T_i$ to cloud;
            end
        end
        else
            Offload $T_i$ to cloud;
        end
    end
end
```

Algorithm 1. Vehicular big data offloading algorithm.

3.5 Algorithm

Algorithm 1 is developed to smartly offload VBD by evaluating delay, congestion, network utilization, and task arrival time. It operates on data collected from diverse onboard sensors, including video feeds, LiDAR, radar, SONAR, and GPS, generated by autonomous vehicles within a specific region. For each incoming task, the algorithm first calculates the total expected delay D_t, using a composite delay model that integrates transmission, propagation, processing, queuing, and handover delays. In parallel, it estimates the network's congestion probability Con_{Prob} to assess the current infrastructure load.

We used delay budget and a congestion thresholds. Consistent with vehicular V2X latency guidance, the delay budget is enforced on the *networking component* of latency, $D_{\text{net}} = D_{\text{tran}} + D_{\text{prop}} + D_{\text{hovr}}$, rather than the full end-to-end time. Unless stated otherwise, we use a single budget $\delta_{\text{net}} = 200\,\text{ms}$. For congestion, we estimate Con_{Prob} over a short window (e.g., $W = 100\,\text{ms}$) using channel-busy-ratio/utilization and set $\theta = 0.7$, a commonly adopted operating boundary that avoids the response-time knee near $\sim$ 70% utilization and aligns with cooperative ITS congestion-control practice (Balador et al., 2022; ETSI, 2011). These literature-backed settings are implemented and validated in our RegionalEdgeSimPy experiments: decreasing δ_{net} triggers premature cloud offloading, whereas increasing it admits unnecessary local delay and degrades responsiveness.

If both the delay and congestion levels remain below these thresholds, the algorithm checks the network utilization and the time of day. When network utilization is high but stable and conditions are normal, data is processed locally at the RC server to avoid the high latency associated with cloud processing. During off-peak hours, the algorithm also prefers to process data at the RC layer, assuming there is sufficient processing capacity. If regional servers are overloaded, the algorithm then offloads the task to the cloud. Additionally, when specific data is needed for training or collaborative learning in another geographic region, the algorithm routes it to the cloud. This ensures model sharing and optimization across distributed regions. Overall, this adaptive offloading strategy minimizes delay, balances workload distribution, and enables scalable, real-time processing in next-generation vehicular networks.

4 Evaluation

To assess the effectiveness of the proposed regional offloading framework, we used a custom-developed simulation environment, *RegionalEdgeSimPy*. Unlike general-purpose tools such as EdgeCloudSim (Sim, 2024), RegionalEdgeSimPy is specifically designed to model big data offloading across multi-tier computing architectures. This simulator enables dynamic evaluation of performance under varying network conditions, particularly focusing on peak and off-peak hour scenarios.

The evaluation recorded key performance metrics, including processing delay, service time, network congestion, operational cost, and server utilization. By simulating realistic vehicular workloads and data flows, the simulator provided fine-grained insights into how the proposed regional computing model performs in terms of latency reduction, resource allocation, and communication overhead.

4.1 Experimental setup

The experimental setup involved two distinct scenarios designed to compare cloud and regional computing servers. In the first scenario, we calculated the delay, cost, service time, processing time, and server utilization associated with transferring, storing, and processing data on cloud servers. In the second scenario, these parameters were evaluated on regional computing servers to provide a comparative analysis.

For cloud-based processing, a centralized Data Center (DC) was configured, simulating a realistic cloud environment where AV data is transmitted, stored, and processed. The DC was equipped with 204,800 MB RAM, 100 TB storage, 1,000 Mbps uplink bandwidth, four CPUs, and a processing speed of 10,000 MIPS. Similarly, the Regional Computing (RC) tier was provisioned with 40,960 MB RAM, 10 TB storage, 500 Mbps bandwidth, and a processing speed of 3,000 MIPS. These profiles emulate commercially deployed configurations for micro-edge data centers and core cloud backends. The hardware parameters were calibrated to align with the compute and network configurations commonly used in established simulation frameworks and prior deployment studies (Gupta et al., 2017; Sim, 2024; Calheiros et al., 2011).

To observe server performance under varying loads, two operational periods were defined: peak hours from 01:00 PM to 09:00 PM, during which 1,000 AVs sent simultaneous requests to the server, and off-peak hours from 01:00 AM to 09:00 AM, where 100 AVs requested resources concurrently. Each request was processed through the cloud and regional server setups, allowing for comparative analysis across peak and off-peak periods.

The parameters measured across both scenarios included service time, representing the total time taken to serve each data request; processing time, or the time taken by the server to process the data; network delay, indicating latency during data transfer; server utilization, reflecting the percentage of server resources used; and cost, denoting the overall expense associated with data transfer and processing.

To ensure the reliability of results, each experimental run was repeated ten times for each scenario, with average values calculated for each parameter. This approach provided a robust basis for evaluating the comparative effectiveness of cloud versus regional servers, as presented in the results section.

4.2 Service time

As shown in Figure 5, the average service time for regional computing is significantly lower than for cloud computing. In cloud computing, service times fluctuate between 8.005 and 8.283 s as task loads increase from 119,357 to 257,252 tasks. Regional computing, however, starts at just 1.7669 s for 119,357 tasks, increasing only slightly to 2.7696 s at the highest task load of 257,252 tasks. This considerable reduction in service time for regional computing highlights its ability to process tasks more

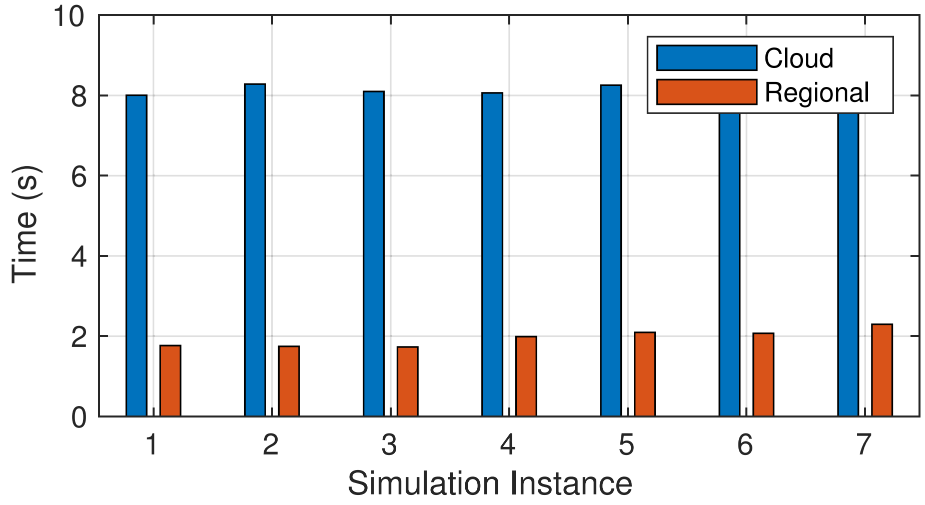


FIGURE 5
Service time comparison of cloud and regional computing.

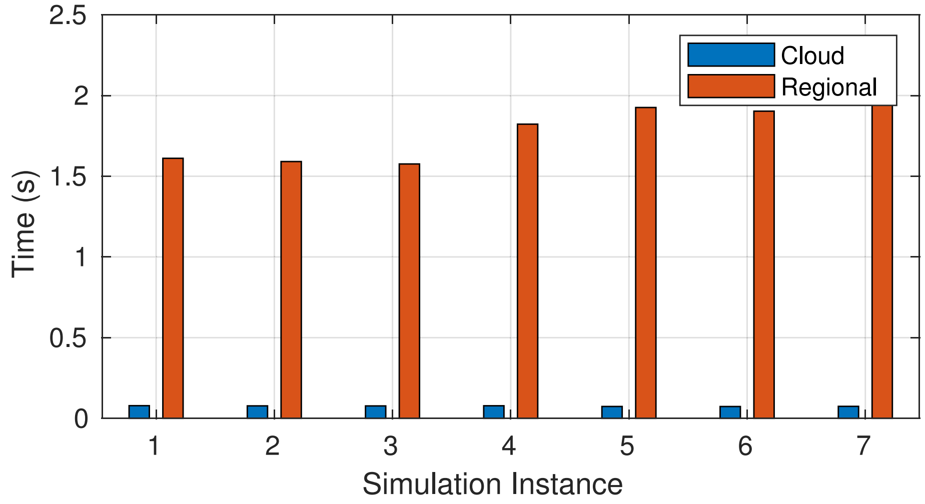


FIGURE 6
Processing time comparison of cloud and regional computing.

efficiently by minimizing delays that are typically introduced in cloud environments.

4.3 Processing time

Figure 6 illustrates the average processing time comparisons. Cloud computing exhibits significantly shorter processing times, ranging from 0.0733 to 0.0786 s, owing to its higher computational capacity. In contrast, regional computing shows longer processing times, starting at approximately 1.6110 s for 119,357 tasks and rising to 2.5716 s for 257,252 tasks. While regional computing has higher raw processing times, it compensates for this by reducing overall service times (as shown in Figure 5) through minimized network delays and queuing overhead.

4.4 Network delay

Figure 7 compares the average network delay across cloud and regional computing. Cloud network delays range from 7.927 to 8.205 s, whereas regional network delays are consistently much lower, ranging from 0.1559 to 0.1979 s as task counts increase from 119,357 to 257,252. This substantial difference underscores the reduced latency in regional computing due to its proximity to the data source, thus improving the overall speed and responsiveness.

4.5 Server utilization

Figure 8 displays the server utilization metrics for both cloud and regional computing environments. Cloud server utilization

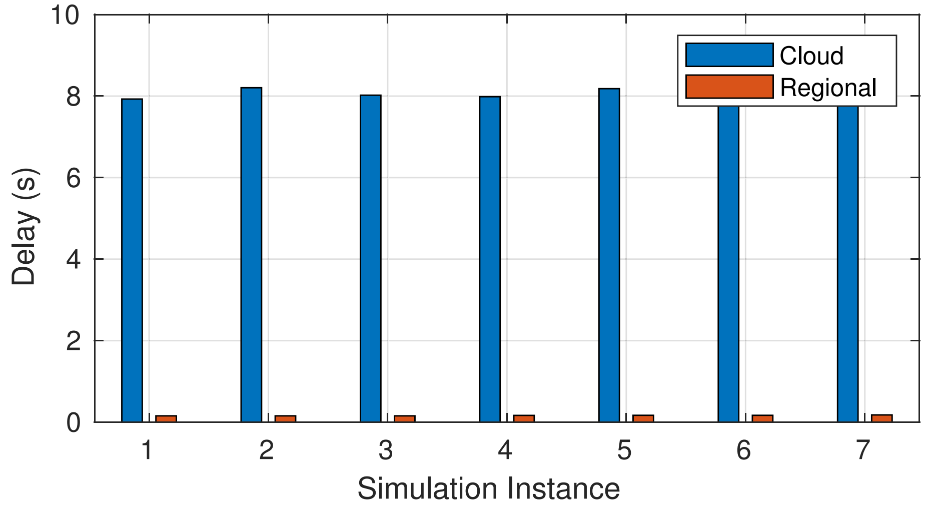


FIGURE 7
Network delay comparison of cloud and regional computing.

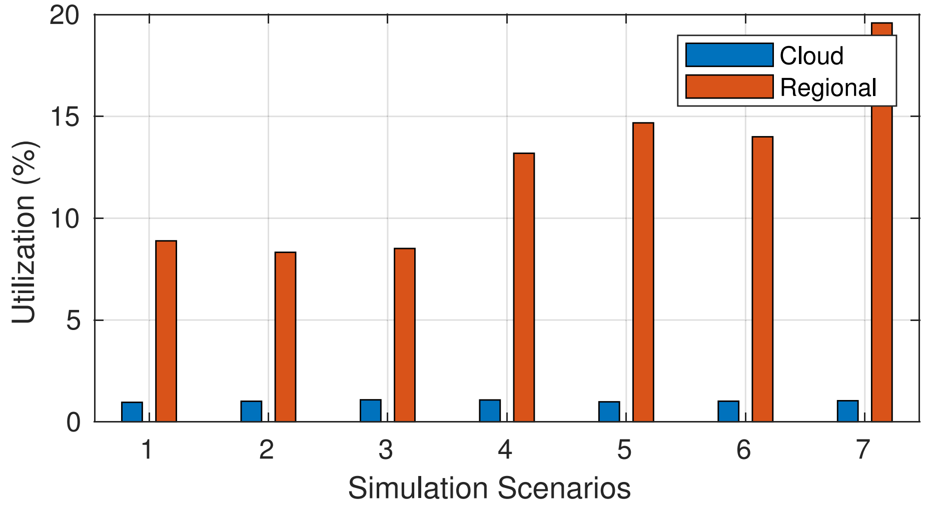


FIGURE 8
Server utilization comparison of cloud and regional computing.

shows a slight fluctuation between 0.958 and 1.081% as the task load increases. In contrast, regional server utilization begins at 8.8877% for 119,357 tasks and increases to 29.2775% as the load reaches 257,252 tasks. Despite higher utilization percentages, regional computing maintains optimal performance, showcasing its resilience and effective resource allocation.

4.6 Cost

Figure 9 illustrates the cost metrics associated with both cloud and regional computing scenarios. The cost for regional computing starts at 0.065$ and incrementally rises to 0.075$ as the number of tasks increases from 119,357 to 257,252. This gradual increase reflects a stable cost structure that aligns well with the growing task load. In contrast, the cloud computing scenario experiences a more significant rise in cost, starting at 0.065$ and escalating to 0.175$ with the increasing number of tasks. This steeper increase highlights the financial implications of utilizing cloud resources, indicating that while cloud computing can handle larger workloads, it incurs substantially higher costs in comparison to regional computing.

In conclusion, regional computing consistently outperforms cloud computing across all tested metrics, with notably lower service times, processing times, and network delays, as well as an efficient server utilization rate, even at higher task loads. These results suggest that regional computing is a more effective solution for high-performance needs in distributed systems.

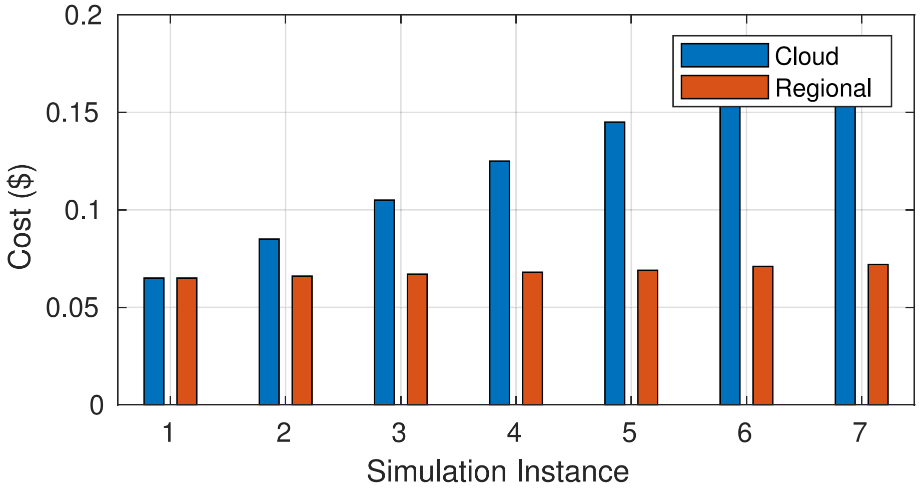


FIGURE 9
Cost comparison of cloud and regional computing.

TABLE 2 Comparison of vehicular regional computing and cloud computing.

Parameters	Regional computing	Cloud computing
Vehicles	Regional coverage	Vehicles around the world
Area	Covers a specific region or locality	Global coverage
Delay	Medium-level latency	High-level latency
Data	Aggregated data from regional vehicles	Data from vehicles worldwide
Computation Power	Moderate computational capabilities	High computational capabilities
Storage	Moderate storage capacity	High storage capacity
Server mobility	Immovable (fixed location)	Immovable (fixed location)
Resource scalability	Medium scalability	High scalability

5 Discussion

The findings from our study underscore the significant differences in performance between cloud computing and regional computing for AVs. As shown in Figure 5, the analysis reveals that cloud computing typically incurs higher levels of delay compared to regional computing. While cloud solutions can handle substantial workloads, they often introduce latency due to the distance data must travel to and from centralized servers. In contrast, regional computing minimizes these delays by processing data closer to the AVs, enhancing overall performance. This observation aligns with existing literature, which suggests that leveraging regional computing resources can significantly improve efficiency for AV applications (Badshah et al., 2022b).

Additionally, Figure 6 shows that although cloud computing demonstrates lower raw processing times due to its higher computational power, regional computing compensates by achieving better end-to-end performance because it reduces network queuing and propagation delays. Figure 7 further confirms that regional computing experiences significantly lower network delays than cloud computing, reinforcing its suitability for latency-sensitive vehicular applications.

Moreover, Figure 8 indicates that regional servers operate at higher utilization levels yet still maintain optimal performance under heavy workloads. Figure 9 highlights that regional computing also provides a more stable and lower-cost structure than cloud computing, where costs increase sharply as workloads grow.

RC can be effectively deployed in real-world ITS through a variety of use cases. For instance, RC servers can be installed at major traffic control centers or regional highway authorities to support autonomous vehicle fleets in urban and semi-urban areas. These servers can process VBD from local roads in real time, provide congestion insights, and coordinate map updates across vehicles operating in the region. In another scenario, RC can be integrated with regional logistics hubs, enabling real-time tracking and route optimization for autonomous delivery. These deployments reduce dependency on distant cloud servers, enhance responsiveness, and support location-specific learning and compliance with local traffic policies.

Overall, our results, supported by Figures 5–9, underscore the importance of considering regional computing as a viable alternative to centralized cloud computing for AVs. By utilizing RC resources, AVs can experience improved performance in terms of delay and cost, thereby enhancing their reliability and attractiveness as a computing option.

However, specific challenges must be addressed to ensure that the proposed framework positively impacts delay and cost. The leading challenges are:

- Firstly, the ownership cost presents a challenge as the Vehicular industry must invest significant capital in deploying and operating regional computing servers (Martens et al., 2012).
- Secondly, regional servers' data processing at the terminal level raises security and privacy concerns that must be addressed (Ometov et al., 2022).
- A third challenge is the variation of cyber rules across different regions, requiring the implementation of region-specific management strategies for these issues (Dayanand Lal et al., 2023).

Based on the experimentation above and the literature, we comprehensively compared regional computing and cloud computing for VBD across various parameters such as delay, area, vehicles, data, servers' mobility, computational power, and capacity. The detailed comparison is presented in Table 2.

6 Conclusion

In this article, we introduced the concept of RC as a strategic solution for managing VBD, particularly during peak hours. By shifting processing tasks from distant cloud servers to regionally deployed infrastructure, the proposed architecture effectively reduces latency, mitigates network congestion, and improves the responsiveness of time-sensitive vehicular applications. To evaluate this framework, we developed a custom Python-based simulator, RegionalEdgeSimPy, tailored to model vehicular data offloading across multi-tier architectures. The simulation outcomes confirm that RC significantly enhances processing efficiency and resource utilization, offering a scalable alternative to traditional cloud or edge paradigms. In future work, we plan to integrate AI-based decision-making mechanisms into the offloading strategy, with a particular focus on reinforcement learning techniques such as Proximal Policy Optimization (PPO). To support this, we are developing a specialized simulator named *DrivNetSim*, which models mobility-aware vehicular environments involving vehicular edge, base station edge, and cloud layers. This platform will enable experimentation with varying traffic loads, server congestion, and communication delays.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

ABa: Conceptualization, Visualization, Writing – original draft. TA: Methodology, Writing – review & editing. SA: Methodology, Writing – review & editing. AA: Investigation, Resources, Writing – review & editing. ABu: Data curation, Software, Writing – review & editing. AD: Project administration, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Alexakis, T., Peppes, N., Demestichas, K., and Adamopoulou, E. (2022). A distributed big data analytics architecture for vehicle sensor data. *Sensors* 23:357. doi: 10.3390/s23010357

Alharbey, R., Shafiq, A., Daud, A., Dawood, H., Bukhari, A., Alshemaimri, B., et al. (2024). Digital twin technology for enhanced smart grid performance: integrating sustainability, security, and efficiency. *Front. Energy Res.* 12:1397748. doi: 10.3389/fenrg.2024.1397748

Amin, F., Abbasi, R., Khan, S., Abid, M. A., Mateen, A., de la Torre, I., et al. (2024). Latest advancements and prospects in the next-generation of internet of things technologies. *PeerJ Comput. Sci.* 10:e2434. doi: 10.7717/peerj-cs.2434

Badshah, A., Daud, A., Alharbey, R., Banjar, A., Bukhari, A., Alshemaimri, B., et al. (2024). Big data applications: overview, challenges and future. *Artif. Intell. Rev.* 57:290. doi: 10.1007/s10462-024-10938-5

Badshah, A., Ghani, A., Daud, A., Chronopoulos, A. T., and Jalal, A. (2022a). Revenue maximization approaches in IAAS clouds: Research challenges and opportunities. *Trans. Emerg. Telecommun. Technol.* 33:e4492. doi: 10.1002/ett.4492

Badshah, A., Ghani, A., Irshad, A., Naqvi, H., and Kumari, S. (2021). Smart workload migration on external cloud service providers to minimize delay, running time, and transfer cost. *Int. J. Commun. Syst.* 34:e4686. doi: 10.1002/dac.4686

Badshah, A., Ghani, A., Siddiqui, I. F., Daud, A., Zubair, M., Mehmood, Z., et al. (2023). Orchestrating model to improve utilization of IAAS environment for sustainable revenue. *Sustain. Energy Technol. Assess.* 57:103228. doi: 10.1016/j.seta.2023.103228

Badshah, A., Iwendi, C., Jalal, A., Hasan, S. S. U., Said, G., Band, S. S., et al. (2022b). Use of regional computing to minimize the social big data effects. *Comput. Ind. Eng.* 171:108433. doi: 10.1016/j.cie.2022.108433

Balador, A., Cinque, E., Pratesi, M., Valentini, F., Bai, C., Gómez, A., et al. (2022). Survey on decentralized congestion control methods for vehicular communication. *Vehic. Commun.* 33:100394. doi: 10.1016/j.vehcom.2021.100394

Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., and Buyya, R. (2011). Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw. Pract. Exp.* 41, 23–50. doi: 10.1002/spe.995

Chen, C., Zhang, Y., Wang, Z., Wan, S., and Pei, Q. (2021). Distributed computation offloading method based on deep reinforcement learning in ICV. *Appl. Soft Comput.* 103:107108. doi: 10.1016/j.asoc.2021.107108

Daniel, A., Subburathinam, K., Paul, A., Rajkumar, N., and Rho, S. (2017). Big autonomous vehicular data classifications: towards procuring intelligence in its. *Veh. Commun.* 9, 306–312. doi: 10.1016/j.vehcom.2017.03.002

Dayanand Lal, N., Mahomad, R., Akram, S. V., Averineni, A., Babu, G. R., and Sai Satyadeva, C. N. (2023). "Innovative cyber security techniques based on blockchain technology for use in industrial 5.0 applications," in *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)* (Delhi), 1316–1320.

Du, S., Hou, J., Song, S., Song, Y., and Zhu, Y. (2020). A geographical hierarchy greedy routing strategy for vehicular big data communications over millimeter wave. *Phys. Commun.* 40:101065. doi: 10.1016/j.phycom.2020.101065

ETSI (2011). *Intelligent Transport Systems (Its); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems Operating in the 5 ghz Range; Access Layer Part. ETSI TS 102 687 V1.2.1*. Sophia Antipolis Cedex: ESTI.

Garg, S., Singh, A., Kaur, K., Aujla, G. S., Batra, S., Kumar, N., et al. (2019). Edge computing-based security framework for big data analytics in vanets. *IEEE Netw.* 33, 72–81. doi: 10.1109/MNET.2019.1800239

Guo, L., Dong, M., Ota, K., Li, Q., Ye, T., Wu, J., et al. (2017). A secure mechanism for big data collection in large scale internet of vehicle. *IEEE Internet Things J.* 4, 601–610. doi: 10.1109/JIOT.2017.2686451

Gupta, H., Dastjerdi, A. V., Ghosh, S. K., and Buyya, R. (2017). IFOGSIM: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments. *Softw. Pract. Exp.* 47, 1275–1296. doi: 10.1002/spe.2509

He, J., Yang, K., and Chen, H.-H. (2020). 6g cellular networks and connected autonomous vehicles. *IEEE Netw.* 35, 255–261. doi: 10.1109/MNET.011.2000541

Khan, M. A., Khan, M., Dawood, H., and Daud, A. (2024). Secure explainable-AI approach for brake faults prediction in heavy transport. *IEEE Access* 12, 114940–114950. doi: 10.1109/ACCESS.2024.3444907

Kumer, S. A., Nadipalli, L. S., Kanakaraja, P., Kumar, K. S., and Kavya, K. C. S. (2021). Controlling the autonomous vehicle using computer vision and cloud server. *Mater. Today Proc.* 37, 2982–2985. doi: 10.1016/j.matpr.2020.08.712

Lee, D., Camacho, D., and Jung, J. J. (2023a). Smart mobility with big data: approaches, applications, and challenges. *Appl. Sci.* 13:7244. doi: 10.3390/app13127244

Liang, Q., Lai, X., Li, Z., Sheng, W., Sun, L., Cai, Y., et al. (2025). Optical communications in autonomous driving vehicles: requirements, challenges, and opportunities. *J. Lightwave Technol.* 43, 1690–1699. doi: 10.1109/JLT.2025.3533911

Liu, C., Li, J., and Sun, Y. (2024). Deep learning-based privacy-preserving publishing method for location big data in vehicular networks. *J. Signal Process. Syst.* 96, 401–414. doi: 10.1007/s11265-024-01912-z

Manikandan, K., Pamisetty, V., Challa, S. R., Komaragiri, V. B., Challa, K., Chava, K., et al. (2025). Scalability and efficiency in distributed big data architectures: a comparative study. *Metall. Mater. Eng.* 31, 40–49. doi: 10.63278/1318

Martens, B., Walterbusch, M., and Teuteberg, F. (2012). "Costing of cloud computing services: a total cost of ownership approach," in *2012 45th Hawaii International Conference on System Sciences* (Maui, HI: IEEE), 1563–1572. doi: 10.1109/HICSS.2012.186

Mordorintelligence (2024). *Autonomous Vehicle Data Storage*. Available online at: https://premioinc.com/pages/autonomous-vehicle-data-storage.

Murk, Malik, A. W., Mahmood, I., Ahmed, N., and Anwar, Z. (2019). Big data in motion: a vehicle-assisted urban computing framework for smart cities. *IEEE Access* 7, 55951–55965. doi: 10.1109/ACCESS.2019.2913150

Nan, J., Qiang, D., Liangliang, W., Shaojie, G., and Shilan, H. (2023). "Cloud computing design of intelligent connected vehicles based on regional load," in *Sixth International Conference on Intelligent Computing, Communication, and Devices (ICCD 2023), Volume 12703* (Hong Kong: SPIE), 517–523.

Ometov, A., Molua, O. L., Komarov, M., and Nurmi, J. (2022). A survey of security in cloud, edge, and fog computing. *Sensors* 22:927. doi: 10.3390/s22030927

Padmaja, B., Moorthy, C. V., Venkateswarulu, N., and Bala, M. M. (2023). Exploration of issues, challenges and latest developments in autonomous cars. *J. Big Data* 10:61. doi: 10.1186/s40537-023-00701-y

Prehofer, C., and Mehmood, S. (2020). "Big data architectures for vehicle data analysis," in *2020 IEEE International Conference on Big Data (Big Data)* (Atlanta, GA: IEEE), 3404–3412. doi: 10.1109/BigData50022.2020.9378397

Rajput, N. S., Singh, U., Dua, A., Kumar, N., Rodrigues, J. J. P. C., Sisodia, S., et al. (2023). Amalgamating vehicular networks with vehicular clouds, Ai, and big data for next-generation its services. *IEEE Trans. Intell. Transp. Syst.* 25, 1–15. doi: 10.1109/TITS.2023.3234559

Ramirez-Robles, E., Starostenko, O., and Alarcon-Aquino, V. (2024). Real-time path planning for autonomous vehicle off-road driving. *PeerJ Comput. Sci.* 10:e2209. doi: 10.7717/peerj-cs.2209

Santos, D., Chi, H. R., Almeida, J., Silva, R., Perdigão, A., Corujo, D., et al. (2025). Fully-decentralized multi-MNO interoperability of Mec-enabled cooperative autonomous mobility. *IEEE Trans. Consum. Electron.* doi: 10.1109/TCE.2025.3569343

Sim, E. C. (2024). *Edgecloudsim*. Available online at: https://github.com/CagataySonmez/EdgeCloudSim (Accessed December, 2024).

Sivakumar, G., Pazhani, A. A. J., Vijayakumar, S., and Marichamy, P. (2025). "Trends in distributed computing," in *Self-Powered AIoT Systems* (Apple Academic Press), 199–217. doi: 10.1201/9781032684000-10

Statista (2024). *Penetration Rate of Light Autonomous Vehicles (l4) Worldwide in 2021, with a Forecast Through 2030*. Available online at: https://www.statista.com/statistics/875080/av-market-penetration-worldwide-forecast/ (Accessed December, 2024).

Waqas, M., Abbas, S., Farooq, U., Khan, M. A., Ahmad, M., Mahmood, N., et al. (2024). Autonomous vehicles congestion model: a transparent LSTM-based prediction model corporate with explainable artificial intelligence (EAI). *Egypt. Inform. J.* 28:100582. doi: 10.1016/j.eij.2024.100582

Xu, W., Zhou, H., Cheng, N., Lyu, F., Shi, W., Chen, J., et al. (2018). Internet of vehicles in big data era. *IEEE/CAA J. Autom. Sin.* 5, 19–35. doi: 10.1109/JAS.2017.7510736

Zhang, Y., Chen, C., Zhu, H., and Wang, J. (2024). "Task offloading for mec-v2x assisted autonomous driving," in *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)* (Singapore: IEEE), 01–05. doi: 10.1109/VTC2024-Spring62846.2024.10683168