# Multi-Service Caching-Based Load Balancing in Multi-RAT Vehicle-to-Infrastructure Communication

Mohammed Mudhafar Shakir [1], Lukman Audah [1], Roshayati Yahya [1,*], Mohammed A. Altahrawi [2], Abdinasir Hirsi [1, 3], and Abdullahi Farah [4]

[1] Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia
[2] Department of Computer Engineering and ELECTRONICS, Faculty of Engineering and Smart Systems, University College of Applied Science (UCAS), Gaza, Palestine
[3] Faculty of Engineering, Jamhuriya University of Science and Technology, Mogadishu 2602, Somalia
[4] Engineering Department, Somtel Telecommunication Company, Bosaso 25290, Somalia
Email: mohammedm_1990@yahoo.com (M.M.S.); hanif@uthm.edu.my (L.A.); rhayati@uthm.edu.my (R.Y.); mtahrawi@ucas.edu.ps (M.A.A.); abdinasirhirsi@just.edu.so (A.H.); abdalla.gaash@somtelnetwork.net (A.F.)
*Corresponding author

***Abstract*—Vehicular-to-Everything (V2X) communication enables data exchange between vehicles and infrastructure, playing a crucial role in autonomous and connected driving systems. However, growing vehicular density often leads to Radio Access Network (RAN) congestion. To mitigate this, Multi-Radio Access Technology (Multi-RAT) networks offer diverse services and data rates, enhancing throughput and reliability, but they also introduce the challenge of effective Load Balancing (LB) among heterogeneous RATs. Conventional LB approaches, such as those based on Received Signal Strength Indicator (RSSI) or Time-to-Leave (TTL), struggle under dynamic mobility and service variability. This paper proposes two novel LB frameworks: the Multi-Service Caching-Based Load Balancing (MSCLB) and the Intelligent Multi-Service Caching-Based Load Balancing (IMSCLB) schemes. The IMSCLB integrates service-aware caching with a CNN-LSTM predictive model to forecast RAT load and optimize resource allocation adaptively. Simulations conducted across three RATs (IEEE 802.11n, 802.11ac, and 802.11ad) and three service categories (Safety, Traffic, and Information) demonstrate substantial improvements over traditional methods. Specifically, MSCLB improves throughput by 10 Mbps, boosts packet delivery ratio by 10%, and reduces latency by up to 6%, while IMSCLB further enhances performance consistency, achieving satisfaction rates of 99.89%, 99.99%, and 99.99% for 802.11n, 802.11ac, and 802.11ad, respectively. These results validate the effectiveness of integrating service-aware caching and predictive intelligence for efficient load balancing in heterogeneous vehicular networks.***

***Keywords*—caching, load balancing, multi-RAT, Vehicular-to-Everything (V2X), multi-service**

Manuscript received September 7, 2025; revised October 27, 2025; accepted November 4, 2025; published May 25, 2026.

## I. Introduction

The Fifth Generation (5G) network employs several methods to manage network load efficiently, notably through small-cell deployment [1]. This structure combines a high-power macro cell for wide coverage with multiple low-power small cells, such as pico and femto cells, to offload traffic from congested macro cells. As demand for high-speed wireless services increases, Load Balancing (LB) becomes essential for maintaining performance [2], and the same applies to vehicular communication systems facing similar challenges.

In 5G, macro and small cells differ in coverage and power. Macro cells connect users through cloud servers, while small cells handle localized low-rate transmissions [3]. However, because macro cells have stronger Reference Signal Received Power (RSRP), they attract more users, leading to overload, whereas small cells remain underutilized. This imbalance requires efficient LB techniques in heterogeneous networks [4].

Conventional interference management directs users toward less congested cells to reduce inter-cell interference but often lowers spectral efficiency by reserving part of the spectrum for interference control [5]. Adaptive cell selection algorithms also face a trade-off between interference mitigation and throughput.

To address these limitations, Cell Range Extension (CRE) has been introduced to enhance load distribution [6]. Yet, existing CRE schemes often lack accuracy, particularly for edge users, resulting in suboptimal LB and reduced throughput [7].

In Vehicular-to-Everything (V2X) communication, which includes Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Pedestrian (V2P), and

doi:10.12720/jcm.21.3.431-447

Vehicle-to-Network (V2N) links (Fig. 1), reliable low-latency connections are crucial, especially for safety-critical signals with modest data rates [8], as shown in Table I. Mobile Edge Computing (MEC) supports these requirements by bringing computation and storage closer to users, reducing latency and improving reliability [9]. Integrating MEC with 5G enables intelligent transport systems to overcome computing limitations and support real-time decision-making [10]. Vehicles gather road data through onboard sensors, processed by MEC nodes and cloud servers for timely responses [11]

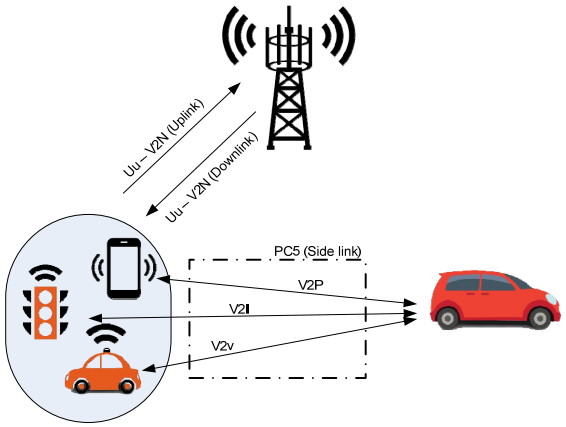



Fig. 1. V2X technology and interface.

TABLE I. REQUIREMENTS FOR V2X APPLICATIONS [12]

| Application | Latency (ms) | Reliability | Data rate (Mbps) |
|---|---|---|---|
| Safety aware and collision avoidance | 10 | 10−5 | < 5 |
| Traffic aware | 10 | 10−2 | 100 |
| Infotainment and augmented reality | 500 | None | 40 (per video) |

Recent studies have explored the use of Multi-Radio Access Technology (Multi-RAT) in V2X networks to improve reliability, increase throughput, and minimize latency [13]. Efficiently balancing the load between these diverse RATs in a heterogeneous network is crucial to meet the varying demands of V2X throughout a vehicle's journey [14], as shown in Fig. 2.

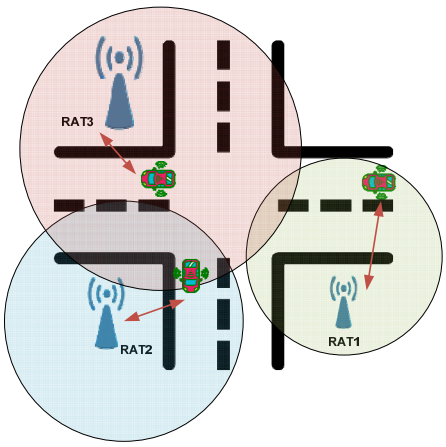



Fig. 2. Multi-RAT communication throughout a vehicle's journey.

In vehicular LB for multi-RAT V2X networks, two common metrics are the Received Signal Strength Indicator (RSSI) and Time to Leave (TTL). RSSI measures signal strength to estimate link quality, while TTL predicts how long a vehicle will remain within a RAT's coverage, helping with handover decisions. However, both metrics have notable limitations. RSSI only reflects signal power without considering congestion or interference, and TTL does not account for real-time traffic or routing dynamics. In high-mobility scenarios, these metrics quickly become outdated, leading to suboptimal balancing and delayed handovers. Furthermore, the diversity of RAT protocols complicates their integration into a unified and scalable LB strategy.

To address these issues, more context-aware methods are required—ones that integrate real-time analytics, service demands, and predictive modeling. RAT caching presents a promising solution by using historical data and mobility patterns to anticipate network conditions. This approach reduces dependence on instantaneous RSSI measurements and complements TTL with trajectory and traffic history, resulting in proactive and smoother handovers.

Different vehicular services also have distinct Quality of Service (QoS) requirements: safety messages demand low latency and high reliability, while infotainment services require high throughput and minimal jitter. Applying differentiated caching strategies tailored to each service type can improve QoS and enhance overall load balancing efficiency.

This paper introduces a Multi-Service Caching-Based Load Balancing (MSCLB) approach to distribute network load across available RATs in Vehicle-to-Infrastructure (V2I) communication. An enhanced version, the Intelligent MSCLB (IMSCLB), incorporates Long Short-Term Memory (LSTM) deep learning to predict traffic patterns and dynamically optimize QoS. Both models are evaluated in terms of throughput, packet delivery ratio (PDR), and latency under different vehicle speeds and service types. The goal is to achieve balanced network utilization, maximize throughput and PDR, and minimize latency in dense multi-RAT vehicular environments.

The rest of the paper is organized as follows: Section II gives a literature review, and Section III describes the proposed methodology. The simulation results are discussed in Section IV. The conclusion of this study and the illustration of future works are given in Section V.

## II. LITERATURE REVIEW

Load balancing in vehicular and multi-RAT communication systems has received substantial attention due to the increasing need for reliable, low-latency connectivity in highly dynamic vehicular environments. Existing studies have explored several strategies, which can be broadly categorized into RSSI/TTL-based, offloading-based, optimization-based, ML-driven, and caching-aware approaches, as summarized in Table II.

RSSI- and TTL-based Load Balancing: Early studies such as Refs. [16–18] primarily depend on the Received Signal Strength Indicator (RSSI) and Time-To-Leave

(TTL) metrics for RAT or base station selection. These methods aim to balance the network load and reduce congestion by connecting vehicles to the strongest signal source. Although simple and computationally efficient, these schemes lack responsiveness to real-time congestion, interference, and service differentiation, often leading to suboptimal handovers and performance degradation in dense traffic scenarios.

Edge Computing and Offloading-based Approaches: Subsequent works [17], Refs. [19–23] leveraged Mobile Edge Computing (MEC), fog computing, and cloud offloading to redistribute computation and communication loads. For instance, Ref. [17] and Refs. [19–21] offload intensive tasks to edge or cloud nodes to improve processing efficiency, Zhang *et al.* [22] and Sondur *et al.* [23] integrated Software-Defined Networking (SDN) for centralized resource orchestration. Despite their effectiveness in localized environments, these approaches introduce additional latency and signaling overhead due to task transfers, and they do not address inter-RAT load balancing challenges.

Storage and Resource Reallocation Mechanisms: Studies in Refs. [24–27] examined dynamic resource and storage allocation to enhance MEC performance. In Ref. [24], a collaborative storage mechanism was proposed to reduce latency and cost, whereas Refs. [25–27] introduced mechanisms such as weighted Voronoi-based region partitioning and buffer-based load redistribution. Although these models mitigate storage imbalance within MEC nodes, they overlook cross-RAT coordination and generate additional signaling overhead and processing delays during resource reallocation.

Optimization and Game-Theoretic Techniques: Several works applied optimization frameworks to jointly allocate resources and schedule offloading tasks. For instance, Hsu *et al.* [28] formulated the LB problem as a Mixed-Integer Nonlinear Programming (MINLP) model, solved using deep learning-assisted Particle Swarm Optimization (PSO), Zhang and Wang [29] adopted a game-theoretic approach to minimize execution time through decentralized competition among users. These methods deliver efficient resource utilization but are constrained by high computational cost, slow convergence, and limited scalability, making them impractical for fast-changing vehicular environments.

Machine Learning and Deep Learning-based LB: Recent advancements employ machine learning for adaptive LB and RAT selection. Dai *et al.* [16] utilized reinforcement learning for dynamic resource distribution, Altahrawi *et al.* [12] proposed a joint LSTM Multi-Criteria (SOLMC) model that predicts connectivity and selects optimal RATs based on multiple parameters, including capacity, delay, and queue length. Although these ML-based methods enhance adaptability, they demand large training datasets, centralized coordination, and significant processing power, limiting their real-time feasibility in V2X networks.

Caching-aware Load Balancing: Caching mechanisms have been introduced in Ref. [30] and Ref. [31] to improve data accessibility and reduce latency. For instance, Varanasi and Chilukuri [30] integrated caching with kernel ridge regression, and Zhang *et al.* [31] employed mobility-aware caching to enhance QoS. However, these works primarily use caching to optimize RAT-level performance, without leveraging it as a cross-RAT LB tool. Furthermore, multi-service differentiation remains unaddressed.

Our proposed MSCLB and IMSCLB schemes differ from prior works, such as service-oriented LSTM-based RAT selection Ref. [12] and caching-aware ML approaches Refs. [30, 31], in several important ways. Unlike [12], which focuses mainly on multi-criteria LSTM-based RAT selection without proactive service-specific caching, our approach integrates caching at each RAT to distribute network load based on both service requirements and predicted traffic demand. Varanasi and Chilukuri [30] and Zhang *et al.* [31] utilized caching to improve QoS within a single RAT, our method explicitly employs caching as a mechanism for load balancing across heterogeneous RATs, enabling coordinated multi-RAT and multi-service management. The CNN-LSTM model further enhances performance by capturing spatial correlations in the service load distribution across RATs and vehicles, which pure LSTM models cannot efficiently extract, and then leveraging LSTM for temporal predictions. This combination improves throughput, reduces latency, and maintains high packet delivery ratios under high mobility and multi-service scenarios, demonstrating the practical novelty and effectiveness of our proposed schemes compared to existing methods.

As summarized in Table II, most existing approaches suffer from limited cross-RAT coordination, high computational complexity, and insufficient multi-service differentiation. To overcome these challenges, this study proposes two complementary solutions: 1) MSCLB employs service-specific caching within each RAT to proactively balance traffic according to service type and cache availability, reducing congestion and latency, and 2) IMSCLB extends MSCLB by integrating LSTM-based traffic prediction, enabling real-time cross-RAT coordination and adaptive LB under high mobility.

Together, the proposed models address the shortcomings of previous RSSI-, MEC-, optimization-, and caching-based solutions by achieving low latency, high throughput, and efficient multi-service QoS provisioning across heterogeneous vehicular networks.

TABLE II. SUMMARY OF RELATED WORKS ON MULTI-RATS LB

| Ref. | Contribution | LB technique | Caching | ML used | Limitation | Multi-RAT | Multi-services |
|---|---|---|---|---|---|---|---|
| [16] | Distribute the network load evenly and reduce congestion | RSSI | × | Reinforcement learning | Limited application | × | × |
| [17] | Seamless handoff between BSs. | RSSI and offloading | × | × | Latency increases due to the required offloading time. | × | × |
| [18] | Optimize data transmission paths | RSSI with routing protocol optimization | × | × | Complexity increases when density increases. | × | × |
| [19−21] | Improving transmission efficiency in high-density vehicle networking | RSSI and offloading | × | Deep learning | Latency increases due to the required offloading time. | × | × |
| [22] | Optimize data transmission paths | RSSI, with the centralized decision at SDN | × | × | Less performance for edge users. | × | × |
| [23] | Enhance the performance of edge users. | RSSI at MEC | × | × | Limited application | × | × |
| [24] | Allowing dynamic resource allocation to reduce storage latency and cost | --- | × | × | No LB attention. | × | × |
| [25] | Enhancing LB among MEC | RSSI at MEC | × | × | Complexity increases by non-linear complex optimization problem. | × | × |
| [26] | Enhancing LB among MEC | RSSI at MEC | × | × | Increase waiting time until accept/reject the offloading between BS. | × | Task request |
| [27] | Enhancing LB among MEC | RSSI at MEC | × | × | More signaling required | × | Task request |
| [28] | Enhancing LB among MEC | RSSI at MEC | × | Deep learning and PSO | Complexity increases when density increases. | × | × |
| [29] | To minimize task execution time | RSSI at MEC | × | × | Complexity increases when density increases. | × | × |
| [12] | Joint LSTM multi-criteria RAT selection approach in V2I communication | RSSI, TTL | × | LSTM | No caching consideration. | DSRC WIFI C-V2x | Safety, traffic, information |
| [30] | Improving QoS of the network using caching-aware. | --- | √ | kernel ridge regression | No LB attention. | × | × |
| [31] | Solving non-linear optimization problem comes from the mobility-aware to enhance the network's QoS | --- | √ | × | No LB attention. | × | × |
| Proposed work | Multi-Service Caching-aware LB and Intelligent MSCLB in V2I communication | RSSI, TTL, MSCLB | √ | CNN-LSTM | --- | IEEE802.11n IEEE802.11ac IEEE802.11ad | Safety, traffic, information |

## III. METHODOLOGY

The approach detailed in this study involves a three-phase process. Initially, the framework for the analysis is set up by positioning vehicles and linking each one to one of three RATs, determined by the signal strength detected. The second phase entails implementing LB across three scenarios: RSSI, TTL, and MSCLB. At the simulation's onset, vehicles establish connections based on the strongest RSSI. Once movement commences, each vehicle selects the most appropriate RAT: the closest vehicle connects via RSSI, while a vehicle farther from a RAT but still within its range and with a longer duration before exiting the range uses TTL for connection. The proposed MSCLB allows vehicles to connect to the suitable RAT

based on the service required by each vehicle, enabling the distribution of load along all RATs. The last part of this simulation compares MSCLB and IMSCLB by proposing a Convolutional Neural Network (CNN) LSTM model (CNN-LSTM) to predict the throughput of each RAT, which means predicting the load on each RAT. The integration with LSTM was selected due to its strong capability in modeling sequential and time-dependent data, which is essential for capturing the dynamic behavior of vehicular networks. Unlike CNN alone, integration with LSTM effectively retains long-term dependencies, allowing it to learn patterns in traffic variation, mobility, and network load over time.

### *A. Vehicle Distribution and Car-Following Model*

This study introduces an innovative connection strategy reliant on the caching capabilities of the RAT, tailored to the specific service required by the vehicle. The three distinct services considered are safety (*Sf*), traffic (*Tr*), and information (*IN*). The *Sf* service is crucial at intersections with high vehicle densities, the *Tr* service is vital for vehicles intending to overtake and needing to assess the road and its surroundings, and the Information Service (*IN*), though less critical, is necessary for vehicles in less crowded conditions that require substantial data.

Each RAT houses a cache profile, and high-speed backhauling facilitates data exchange between RATs, enabling collaborative relaying when necessary. Information on all RATs' cache profiles is uploaded to the cloud, making it accessible for all network RATs. This cloud-stored data aids RATs in understanding network conditions and selecting the most appropriate RAT for connection.

All vehicles $k$, in the simulation scenario, are randomly distributed and updated in their position ($x_i$, $y_i$) using Poisson distribution where ($x_i$, $y_i$) is the position coordinate of the $i^{th}$ vehicle. The Poisson distribution is employed in V2I network modeling for several compelling reasons. Firstly, it effectively represents traffic arrival patterns. In V2I communication, data packets are exchanged between vehicles and RATs, and the timing of these packet arrivals can be described by a Poisson process. This allows the average arrival rate to be captured and used to predict and analyze traffic behavior within the network. Secondly, the Poisson distribution is well-suited for capturing the inherent randomness of events in vehicular environments, such as vehicle arrivals in a given area, the initiation of data transmissions, or the occurrence of channel errors. Its assumption of event independence and memorylessness aligns with the unpredictable nature of these phenomena. Lastly, the Poisson distribution simplifies the analytical process. Its mathematical tractability makes it easier to design and evaluate V2I systems, allowing researchers to derive closed-form expressions for key performance metrics like throughput and PDR, as [34]:

$$Pd(k;\mu) = \frac{e^{-\mu}\mu^{k}}{k!} \tag{1}$$

where $\mu$ is the average rate of events occurring in the interval, greater than 0, $Pd$ is the Poisson distribution, and $k$ is the total number of vehicles.

The simulation employs the car-following model, a theoretical framework designed to describe vehicle behavior in traffic, particularly under congested conditions where interactions between vehicles are prominent Ref. [35] and Ref. [36]. This model aims to predict how drivers adjust their speed and position in response to the actions of the vehicle directly ahead. Each vehicle in the model is characterized by its dynamic properties, such as maximum speed, acceleration, and deceleration limits, which govern its movement over time. Additionally, vehicles strive to maintain a safe following distance, which in this study is quantified using the distance headway—the physical gap between successive vehicles.

The response of a following vehicle is primarily influenced by the speed and acceleration or deceleration of the lead vehicle. As the lead vehicle alters its speed, the trailing vehicle adapts its motion to preserve a safe and consistent spacing. This paper adopts the linear follow-the-leader model to implement the car-following behavior, as illustrated in Fig. 3. In this approach, a vehicle's acceleration is modeled as directly proportional to the speed difference between itself and the vehicle ahead, and inversely proportional to the square of their headway distance. The acceleration of each vehicle, $a_i$, is calculated accordingly using [34]:

$$a_i = g\left(\frac{v_{i-1}-v_i}{x_{i-1}-x_i}\right) \tag{2}$$

where $v_i$ and $v_{i-1}$ are the speeds of the following vehicle $k_i$ and the lead vehicle $k_{i-1}$, respectively, $x_i$ and $x_{i-1}$ are the positions of the $k_i$ and $k_{i-1}$, respectively, and g is a sensitivity factor that represents how aggressively a driver reacts to the difference in speeds, considering the headway. In this paper, g is equal to 0.5 to simulate a more realistic real scenario.

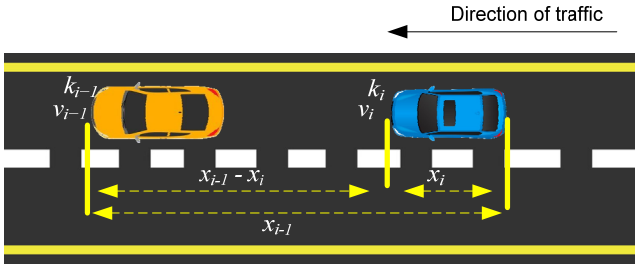



Fig. 3. Linear follow-the-leader car-following model.

### *B. The Proposed Scenario*

The simulated scenario illustrated in Fig. 4 involves N different types of RATs, where $N = 3$ in this study. The selected RATs include IEEE 802.11n, IEEE 802.11ad, and IEEE 802.11ac, all of which are widely adopted in WiFi-based vehicular networks. Each RAT provides a specific coverage area denoted by $D_N$ and is positioned at a distance $d_i$ from vehicle $k_i$, representing the physical separation between the $i^{th}$ vehicle and the corresponding RAT. The technical specifications and parameters for each RAT used in the simulation are detailed in Table III [32].

TABLE III. IEEE 802.11 RAT SPECIFICATIONS

| RAT standard | Channels used | Carrier frequency (GHz) | Bandwidth (MHz) |
|---|---|---|---|
| IEEE 802.11n | 6 | 40 | 1 |
| IEEE 802.11ac | 2.437 | 5.2 | 60.48 |
| IEEE 802.11ad | 20 | 40 | 2160 |

The simulation scenario begins with the random distribution of vehicles across the environment. Each vehicle requests one of three service types (*SE*): *Sf*, *Tr*, or *IN*. The required $R_{SE}$ for these services are 0.5 Mbits/s for *Sf*, 2 Mbits/s for *Tr*, and 10 Mbits/s for *IN*. The distribution of these services is as follows:

1) At intersections, such as with vehicle 1 in Fig. 4, the vehicle sends both *Sf* and *Tr* services to the associated RAT. This aligns with the importance of safety and traffic information at intersections, where vehicle coordination and awareness are critical.
2) Outside intersections, as seen with vehicles 2 and 3 in Fig. 4, service requests depend on the vehicle's surroundings:

Case (a): If the vehicle is distant from any intersection and has no other vehicles within a 6-meter radius in all directions, it sends the *IN* service to the RAT. This scenario is demonstrated by vehicle 2.

Case (b): If the vehicle is also away from an intersection but detects other vehicles within 6 meters in any direction, it prioritizes safety and transmits an *Sf* service to the RAT. This is illustrated by vehicle 3 in the Fig. 4.

After sending its designated service, each vehicle connects to one of the RATs based on one of four selection methods: RSSI, TTL, the proposed MSCLB, or the IMSCLB. These selection approaches are discussed in detail in the following sub-sections.

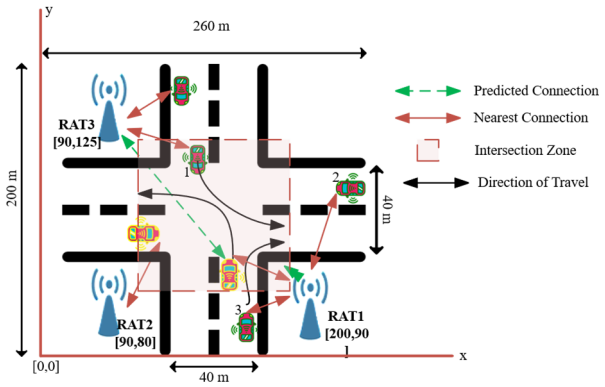



Fig. 4. Simulation scenario.

*1) The first scenario: RSSI load balancing*

To begin with, a scenario is constructed where all vehicles have varying speeds $v_i$ ranging from 1 m/s to 50 m/s. The number of vehicles $k$ is 50 at each iteration, distributed randomly using a Poisson distribution. This randomness in vehicle positions ensures a realistic representation of a dynamic vehicular network. The vehicles' movements are also simulated based on their assigned speeds and car-following model, allowing for realistic mobility patterns. This part provides insights into the network's performance under the worst-case distance-based scenario.

To evaluate different V2I use cases fairly, each vehicle randomly selects a use case to simulate. This random selection ensures equal consideration and representation of all available use cases, avoiding bias in the evaluation process. The simulation time (T) is set to 100 seconds to capture an adequate time frame for observing and analyzing the network's performance under different scenarios.

For the RSSI scenario, at every ($t$) equals one-second interval, the positions of the vehicles ($x_i$, $y_i$) are updated based on the Poisson distribution. Subsequently, each vehicle determines the nearest RAT based on the RSSI. The RSSI between $i^{th}$ vehicle and $N$ RATs, $£_i^N$, depends on the minimum $d_i$ between $k_i$ and $N$. The decision to connect to the RAT depends on the highest $£_i^N$. The number of vehicles selecting each RAT is counted to assess the distribution of RAT selection. After all vehicles are assigned to one of the RATs available, the network performance is evaluated by measuring throughput, PDR, and latency. This evaluation considers the calculated path loss ($PL$) values considering the vehicle specifications such as ($x_i$, $y_i$), $d_i$, and $f$ for the RAT as:

$$PL(dB) = 20\log_{10}(di) + 20\log_{10}(f) + 20\log_{10}\left(\frac{4\pi}{\lambda}\right) \quad (2)$$

where $PL$ is the free-space path loss, $d_i$ is the Euclidian distance between the vehicle and the RAT, $f$ is the carrier frequency of the RAT used, and λ is the wavelength and equals to $\lambda = c/f$ where $c = 3\times10^8$ is the speed of light. The $£_i^N$ calculation is given by [37]:

$$£_i^N(dBm) = P_r(dBm) + G_r(dBi) - PL(dB) \quad (3)$$

where $G_r$ is the received antenna gain and $P_r$ is the received power calculated by [37]:

$$P_r = P_t\left(\frac{G_t G_r \lambda^2}{(4\pi)^2 d^2}\right) \quad (4)$$

where $P_t$ is the transmitted power.

The utility function $u$ for the RSSI scenario follows the following criteria [12]:

$$u(RSSI)_j^i = \begin{cases} 0, & £_j^i < £_j^{min} \\ \alpha, & £_j^{min} < £_j^i < £_j^{max} \\ 1, & £_j^i \geq £_j^{max} \end{cases} \quad (5)$$

where $i$=1:N and $j$=1:k. $£_j^{max}$ is the maximum RSSI for each $k$, and $£_j^{min}$ is the minimum RSSI for each $k$ at which the receiver can no longer detect the signal energy according to sensitivity levels of IEEE 802.11 standards [12], and finally,

$$\alpha = \frac{£_j^i - £_j^{min}}{£_j^{max} - £_j^{min}} \quad (6)$$

*2) The second scenario: TTL Load balancing*

Estimating the TTL value for each vehicle is crucial for determining the duration of connectivity within the radio coverage of each RAT. We initially determine the maximum TTL ($TTL_{max}$) for each N to compute the TTL.

This is achieved by dividing the total communication range, represented as 2Ds, by the speed of the vehicle, denoted as $v_i$, in the following manner [12]:

$$TTL_{max} = \frac{2D_S}{v_i} \tag{7}$$

Secondly, the instantaneous $TTL_{ins}$ at the current vehicle position can be calculated as [12]:

$$TTL_{ins} = \frac{D}{v_i} \tag{8}$$

where D is the distance between the $k^{th}$ vehicle location $\varphi(k_i) \triangleq \{\varphi_x^k, \varphi_y^k, \varphi_z^k\}$ and the predefined position of each RAT $N_x$ at the *x-axis*, and it is calculated as [12]:

$$D = \begin{cases} D_s + D_x & \varphi_x^k < k_x \\ D_s - D_x & \varphi_x^k > k_x \end{cases} \tag{9}$$

and

$$D_x = |\varphi_x^k - k_x| \tag{10}$$

$$D_s = \sqrt{r^2 - (h_z^k - h_z^N)^2} \tag{11}$$

where $D_x$ is the distance between the N RAT and $k$ vehicle on the horizontal axis, $x$, $r$ is the coverage radius of each RAT, $h_z^k$ and $h_z^N$ are the height of the vehicle and RAT, respectively, and $D_s$ is the diameter of the coverage area. Finally, the TTL is calculated based on the direction of the movement, $dr_m$, as follows [12]:

$$TTL = \begin{cases} TTL_{ins} & dr_m > 0 \\ TTl_{max} - TTL_{ins} & dr_m < 0 \end{cases} \tag{12}$$

The utility function of $TTL$ is proposed as in [12], where it is twice differentiable, monotonic, concavity-convex, and guarantees the following conditions [12]:

$$u(TTL) = \begin{cases} 0, & TTL < TTl_{min} \\ \frac{\left(\frac{TTL}{TTl_{mid}}\right)^{\beta^2}}{1+\left(\frac{TTL}{TTl_{mid}}\right)^{\beta^2}} & TTl_{min} \le TTL \le TTl_{mid} \\ 1 - \frac{\left(\frac{TTl_{max}-TTL}{TTl_{max}-TTl_{mid}}\right)^{\beta^2}}{1+\left(\frac{TTl_{max}-TTL}{TTl_{max}-TTl_{mid}}\right)^{\beta^2}} & TTl_{mid} \le TTL \le TTl_{max} \\ 1, & TTL \ge TTl_{max} \end{cases} \tag{13}$$

where $TTL_{mid} = \frac{TTL_{max}+TTL_{min}}{2}$ $TTL_{min}$ is the minimum time required to deliver all V2x service packets from the source to the destination, computed by dividing the maximum number of service *SE* packets by the $R_{SE}$ specified for this service. $\beta$ is the constant parameter that determines the sharpness of the utility function, $\beta > 0$.

*3) The proposed MSCLB*

Caching in vehicular networks is a strategic approach to enhance data accessibility and reduce latency in communication between vehicles and RATs, which is pivotal for safety-critical applications and infotainment services. By storing data locally in RATs, caching minimizes the dependence on continuous network connectivity, which is crucial in high-speed mobility environments where network conditions can fluctuate rapidly.

This technique leverages vehicles' mobility patterns and content's popularity to decide what data to cache and when to update or replace it. As a result, it significantly improves data dissemination efficiency, ensuring timely access to vital information for driving safety and enriching the in-vehicle experience with seamless access to entertainment content. This section introduces the operation of caching, building on Service-based Networking (SbN) to exploit its native support for in-network caching.

Each RAT maintains a cache sized at *ch* data chunks. This cache is divided into three parts based on a parameter δ, where δ ranges between 0 and 1. The first part, C1, can hold up to (δ*c) chunks, where δ here equals 0.5, dedicated to *Sf* data. The second part, C2, can hold up to (0.3*c) chunks reserved for Tr content, while the remaining C3 can hold up to (0.2*c) chunks reserved for the *IN* content [12].

Distinct eviction strategies are applied to these sections: First-In-First-Out (FIFO) for all Cs. Table IV shows the possible cases of LB based on caching.

TABLE IV. POSSIBLE CASES OF LB BASED ON CACHING

| Case | Service needed | Vehicle's Location | RAT to connect to | Description |
|---|---|---|---|---|
| 1 | *Sf* | Intersection | IEEE 802.11n | n RAT gives high coverage and less data rate, which is suitable for *Sf* services. |
| 2 | *Sf* and *Tr* | Near the intersection | IEEE 802.11n | Due to the wide coverage that n RAT offers. |
| 3 | *Tr* | Overtake another vehicle | IEEE 802.11ac | To offload the load on n RAT, ac RAT can serve vehicles that need *Tr* or *IN* data. |
| 4 | *IN* | Away from intersection | IEEE 802.11ad | ad RAT gives a very high data rate suitable for the required *IN* data. |
| 5 | *Sf* but n RAT is overloaded | Intersection | IEEE 802.11ac | If n RAT is overloaded, ac RAT offers a small portion of its cash to help in *Sf* data when needed. |
| 6 | *Tr* but ac RAT is overloaded | Near the intersection | IEEE 802.11ad | If ac RAT is overloaded, ad RAT offers a small portion of its cash to help in *Tr* data when needed. |

Fig. 5 depicts the simulation process flowchart, beginning with the initialization of the simulation environment by randomly distributing 50 vehicles that adhere to the car-following model. Initially, each RAT is assigned a connection based on the RSSI for the first cycle. As vehicles commence movement, their locations, RSSI, TTL, and required services are updated. RSSI determines connections to the nearest RAT. Conversely, connections

based on TTL prioritize vehicles with longer TTL to RATs servicing their area. For connections determined by caching, they adhere to the scenarios outlined in Table IV. Subsequently, the simulation involves tallying the vehicles linked to each RAT and computing the throughput, PDR, and latency for each RAT before the cycle repeats.

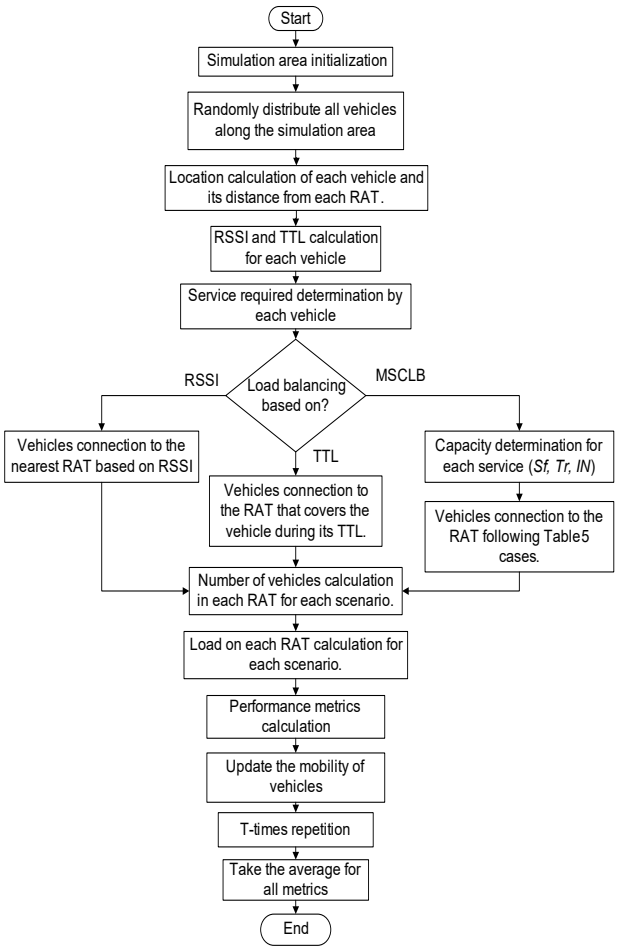



Fig. 5. Flowchart for the simulation.

Each $k$ has its utility function as $u_k = [u(RSSI), u(TTL)]_k$. In this case, the maximization problem for the MSCLB becomes:

$$NV_n = max \prod_i^k u_i \times U_{MSCLB} \quad (14)$$

where

$$U^i_{MSCLB} = \begin{cases} C^i + 1, & C^i > NV^i \\ C^i - 1, & C^i \leq NV^i \end{cases} \quad (15)$$

s.t.:

$$C1: £_k < £_{min}$$
$$C2: C_{sf} > C_{Tr}$$
$$C3: C_{Tr} > C_{IN}$$
$$C4: TTL_k > TTL_{max}$$
$$C5: \sum_{i=1}^{NV_n} P_i = 1$$

The problem in Eq. (15) is nonlinear; to facilitate the solution of it, we proposed the IMSCLB as described in 4.

*4) The proposed IMSCLB*

The methodology proposed in this model focuses on predicting the availability of free *ch* in *N* using a combination of LSTM and CNN deep learning models as shown in Fig. 6. The goal is to enable efficient LB by allowing RATs to make informed decisions regarding connection requests based on caching availability predictions and service required before the connection request is initiated. This methodology utilizes historical channel usage and available data stored in the cloud for online prediction based on the offline training of the IMSCLB according to the dataset generated from the simulation scenario.

The proposed block diagram shown in Fig. 6 starts by entering the dataset gathered from the simulation scenario to the CNN convolutional layer, activated by relu, and has 128 filter orders. The flatten layer converts data after CNN from matrix form to vector form suitable for LSTM sequence input. The time step layer follows the flatten layer to select one element by one to enter the LSTM, which has 40 units as the total simulation time is 100 s. A fully connected layer as an output of the LSTM is activated by relu followed by the dropout layer with a rate of 0.5 because the classification layer is a two-label classifier. The classification labels are (connection) or (no connection) to the chosen RAT.

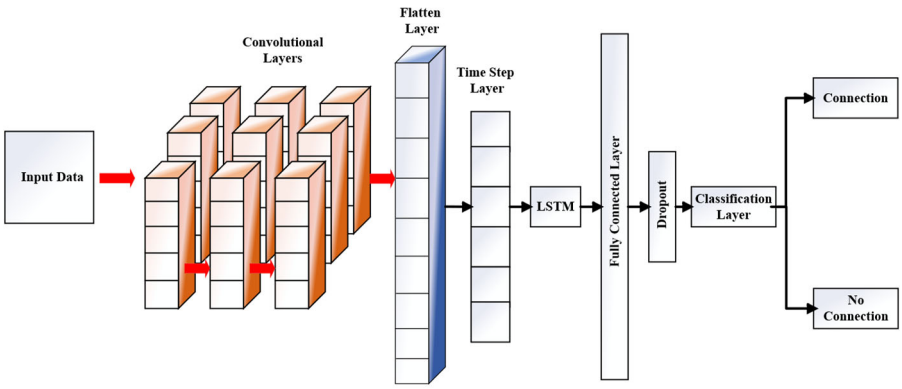



Fig. 6. LSTM-CNN block diagram proposed.

The IMSCLB algorithm offers a data-driven approach that leverages historical data and patterns to make accurate and personalized RAT selection predictions. The purpose of using LSTM is to perform sequence learning. LSTM networks excel in learning from sequential data, making them well-suited for RAT selection prediction. By capturing dependencies and long-term patterns in network and vehicle behavior, LSTM enables accurate predictions based on temporal information. However, the purpose of using CNN is spatial feature extraction. CNNs are renowned for their ability to extract spatial features from input data. In the context of RAT selection, CNNs can process information from the generated dataset, such as RSSI, the data rate required, the type of RAT, the vehicle's position, and the necessary services. The model can make informed decisions regarding RAT selection by learning relevant spatial features.

The IMSCLB dataset was synthetically generated with 50 vehicles operating over n, ac, and ad RATs, covering speeds from 1 to 50 m/s and service requests for safety, traffic, and information. A total of 100,000 samples were created, each capturing instantaneous vehicle positions, RAT connectivity, service demand, and queue backlog. The dataset size ensures robust CNN-LSTM training while keeping computational overhead manageable, with dataset generation and preprocessing completed within 2 hours. This process provides sufficient diversity to capture typical urban and highway scenarios, allowing the predictive model to accurately anticipate RAT load and support real-time load balancing decisions.

The first step of using the proposed IMSCLB is to generate the dataset. The dataset used in this model was gathered by simulating the three simulation scenarios in this work. The dataset generated consists of 17 features: $[\varphi(k_i)$, $v_i$, $\pounds_i^N$, $TTL_N$, $d_i$, and NV for each service], where NV is the number of vehicles connected to each RAT. The correlation coefficient of two random variables measures their linear dependence. If each variable has M scalar observations, then the Pearson correlation coefficient, $\rho$, is defined as:

$$\rho(A,B) = \frac{1}{M-1}\sum_{i=1}^{M}\left(\frac{A_i-\mu_A}{\sigma_A}\right)\left(\frac{B_i-\mu_B}{\sigma_B}\right) \quad (17)$$

where $\mu_A$ and $\sigma_A$ are the mean and standard deviation of A, respectively, and $\mu_B$ and $\sigma_B$ are the mean and standard deviation of B. The correlation coefficient matrix, $R$, of two random variables is the matrix of correlation coefficients for each pairwise variable combination as:

$$R = \begin{pmatrix} \rho(A,A) & \rho(A,B) \\ \rho(B,A) & \rho(B,B) \end{pmatrix} \quad (18)$$

Since A and B are always directly correlated to themselves, the diagonal entries are just 1. For our dataset, the matrix R is given as:

$$R = \begin{pmatrix} 1 & \cdots & \rho(A_1,A_{17}) \\ \vdots & \ddots & \vdots \\ \rho(A_{17},A_1) & \cdots & 1 \end{pmatrix} \quad (19)$$

Fig. 7 displays the correlation matrix among all features utilized in the dataset. There's a substantial correlation, approximately 0.55, between TTL outcomes and the position of each vehicle, as well as a strong correlation between TTL and the velocity of each vehicle. These correlations are expected since TTL relies on vehicles' speed and position. Notably, a correlation reaches 0.8 for TTL results between departing ad RAT and neighboring n RAT. This correlation is sensible because vehicles departing from ad RAT would naturally prefer to connect with n RAT due to its wider coverage.

Additionally, there's a correlation between RSSI values and distance, which is logical as the distance between vehicles and RATs influences RSSI. Furthermore, a correlation of 0.6 exists between velocity and being "*IN*" service, which is reasonable since a vehicle requesting an *IN* service typically indicates higher speed. Lastly, there's a correlation between *Tr* and *Sf* services, which makes sense in an intersection scenario where each vehicle may require both services from the RAT to navigate through traffic efficiently.

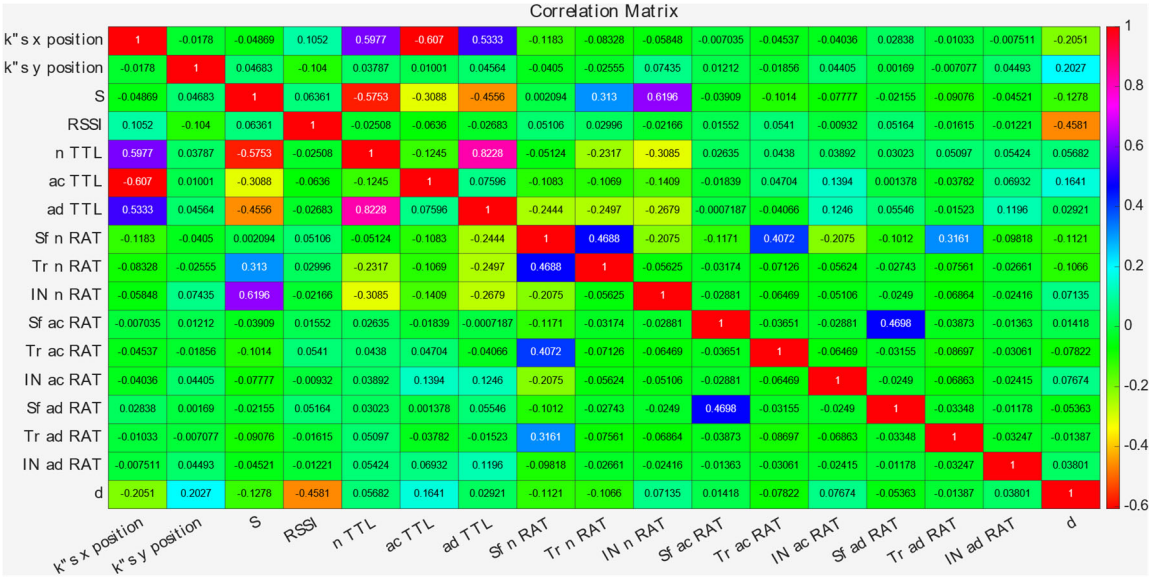



Fig. 7. Correlation matrix for the dataset features.

The model's fusion of LSTM and CNN enables it to simultaneously utilize temporal and spatial information. This integration allows for a comprehensive understanding of the network dynamics and user requirements, resulting in more accurate predictions of the optimal RAT connection. Besides, LSTM-CNN models can adapt to changing network conditions, vehicle demands, and channel characteristics. The models can continuously learn from real-time data, updating their predictions accordingly. This adaptability ensures the model's effectiveness in diverse, dynamic V2X scenarios.

The LSTM-CNN model can learn from individual user behavior and preferences, making it capable of providing personalized RAT selection recommendations. By considering factors such as service requirements, location, device capabilities, and historical usage patterns, the model can optimize the vehicle's connection experience.

The collected data undergoes preprocessing to ensure compatibility with the LSTM-CNN architecture. This preprocessing stage involves data normalization and splitting the dataset into training and testing sets for evaluating model performance. The splitting rate used is 70/30. Once the model is trained, it undergoes evaluation using the testing dataset to assess its predictive capabilities. The fit curves between the predicted and actual channels are used to evaluate the model capability.

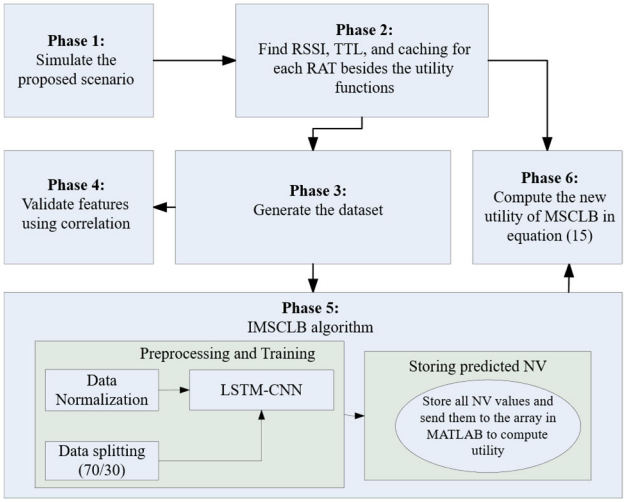



Fig. 8. The IMSCLB algorithm workflow.

With a trained and evaluated LSTM-CNN model, the methodology progresses to the actual prediction phase. The model is deployed and integrated into the RATs within the network. Each RAT continuously monitors the current state of the network and utilizes the trained model to predict the availability of free channels based on real-time data. This enables RATs to make informed decisions when handling connection requests, effectively balancing the load across the network.

The methodology emphasizes continuous learning to ensure the accuracy and adaptability of the predictions over time. The LSTM-CNN model is periodically updated and retrained using newly collected data. This enables the model to adapt to changing network conditions and improve its prediction accuracy by incorporating more recent information.

By employing the proposed methodology as shown in Fig. 8 and Algorithm 1, RATs in the network can leverage the power of deep learning to predict the availability of free channels. This enables efficient LB, optimizes resource utilization, and enhances overall network performance. With access to cloud-stored historical data and the capabilities of LSTM-CNN models, the methodology provides a robust and data-driven approach to address channel availability prediction challenges in LB scenarios. Also, it enhances the overall performance of the network based on throughput.

**Algorithm 1: MSCLB procedure**

1. Set T=100.
2. Generate the first random distribution of $k$ with $S_i$ for each $k$.
3. $k = 50$.
4. Determine *SE* the vehicle sends to the RAT based on location.
5. Allow connection based on *SE*.
6. Offload load from IEEE 802.11n RAT to IEEE 802.11ac, and from IEEE 802.11ac to IEEE 802.11ad RAT if needed.
7. Count the number of vehicles that select the same RAT.
8. Repeat steps 2 to 10 for each $t$.
9. Calculate Throughput, PDR, and latency for each iteration.

TABLE V. SIMULATION PARAMETERS

| Simulation parameter | Value |
|---|---|
| Simulation area ($m^2$) | 260 × 200 |
| No. of RATs | 3 |
| No. of vehicles | 50 |
| Vehicle speed (m/s) | 1 to 50 |
| IEEE 802.11n position | [90,125] |
| IEEE 802.11ad position | [90,80] |
| IEEE 802.11ac position | [200,90] |
| Road width and height (m) | 40 × 40 |
| Vehicles separation (m) | 6 |
| Vehicle height (m) | 1.5 |
| Simulation time (s) | 100 |
| Services | Safety, Traffic, Information |
| The data rate for services (Mbits/s) | 0.5, 2, 10 |
| No. of lanes | 4 (up, down, left, right) |
| **RATs specifications** | |
| Modulation schemes | 64-QAM |
| Coding rate | ¾ |
| Path loss model | Friis propagation |
| Transmit power (dBm) | 30 |
| Rx noise figure (dBm) | -90 |
| IEEE 802.11n height (m) | 15 |
| IEEE 802.11ac height (m) | 20 |
| IEEE 802.11ad height (m) | 25 |
| **CNN-LSTM model parameters** | |
| Optimizer | Adam |
| No. of epochs | 5000 |
| CNN activation | Relu |
| CNN filter order | 128 |
| LSTM units | 100 |
| Prediction output | 9 |
| Drop rate | 0.5 |
| Evaluation | RMSE |

The parameters utilized in the simulation for this study are presented in Table V. The objective of the simulation is to enhance the network's LB and calculate the

throughput, PDR, and latency of a network containing three different IEEE 802.11 RATs and three services required. The number of vehicles involved in the simulation is 50, and their speeds are constrained between 1 m/s and 50 m/s. The selection of different data rates allows for the representation of diverse network configurations. Their distribution follows a Poisson distribution pattern to account for the random distribution of vehicles along the road.

This paper considers three QoS performance metrics: Root Mean Square Error (RMSE) and satisfaction. The QoS metrics are throughput, PDR, and latency. Throughput measures the amount of data that can be transmitted through the network. It's important to note that throughput is distinct from bandwidth, as the latter refers to the maximum capacity of the network channel. Actual throughput, denoted as $B_{rx}$, may differ from the available bandwidth due to congestion and traffic within the network. Also, propagation losses significantly influence throughput value, which can significantly impact its magnitude. Throughput is calculated by [10]:

$$B_{rx} = \frac{8 \times Byte_{rx}}{T2_{packet} - T1_{Packet}} \quad (20)$$

where $T2_{packet}$ means the time last Rx packet and $T1_{packet}$ means the time the first packet

The second QoS performance metric is the PDR. PDR measures the percentage of packets successfully received at the destination compared to the total number of packets sent from the source, which could be either the vehicle or the road sink. The equation below illustrates the calculation of PDR [8]:

$$PDR = \frac{Packets\ successfully\ delivered\ to\ a\ destination}{Packets\ sent\ out\ by\ the\ sender} \quad (21)$$

The third QoS performance metric is latency. Increased latency implies more losses and reduced efficiency from the sender to the receiver. Latency is the time elapsed from when the first bit is sent from the source of the entire message until it reaches the destination and completes the retrieval process. The latency is computed as [11]:

$$\text{Latency} = T_p + T_t \quad (22)$$

where $T_p$ is the propagation latency, and $T_t$ is the transmission latency. Equation 11 assumes that queuing time during transmission and processing delay in the entire process is negligible.

Root Mean Square Error (RMSE) is also measured in this paper for the regression model deep learning training process [33]. It measures the average difference between the predicted values and the actual values, and it is calculated as 15]:

$$RMSE = \sqrt{\frac{1}{N} \times \sum \left(Y_{pred} - Y_{actual}\right)^2} \quad (23)$$

where $N$ is the number of the total data points, $Y_{pred}$ is the predicted value of a point from the deep learning model, and $Y_{actual}$ is the actual value of the same point.

Satisfaction measures the transmission efficiency of each $N$. In general, the satisfaction percentage measured the throughput achieved compared to theoretical throughput for each $N$ as [12]:

$$Satisfaction\ (\%) = \frac{measured\ Brx}{Theoritical\ Brx} \times 100\% \quad (24)$$

## IV. Simulation Results and Discussion

### A. Network Load Balancing

This section details the discussion of simulation results derived from a scenario simulation. As previously mentioned, the simulation involves 50 vehicles scattered across the simulation area, as depicted in Fig. 9. The distribution of vehicles is achieved randomly, following a Poisson distribution commonly utilized in vehicular simulations. Furthermore, the model extends this distribution to simulate a more realistic scenario by incorporating a car-following model, which considers the actual dynamics of moving vehicles on the road.

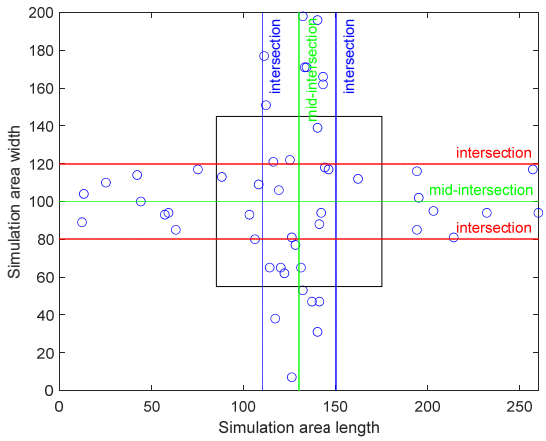



Fig. 9. Vehicles distribution based on the poisson distribution and the Car-following model.

For the two primary connectivity scenarios utilized in Multi-RAT networks, RSSI-based connectivity, TTL-based connectivity, and the proposed MSCLB, the LB results obtained from the network are illustrated in Figure 10 within the simulation area.

In Fig. 10, the load on IEEE 802.11n RAT, stemming from TTL-based connectivity, is notably higher than RSSI-based connectivity. This discrepancy is attributed to the extensive coverage of IEEE 802.11n RAT, enabling vehicles to maintain a connection even as they move away from the network, thus providing more time for connection to IEEE 802.11n RAT for each vehicle. Conversely, in the case of TTL-based connectivity, RSSI diminishes as the distance between the RAT and the vehicle increases. Vehicles with ample time to exit the RAT coverage remain connected to the same RAT based on TTL while becoming disconnected based on RSSI.

This contrast is evident in the load distribution on ad and IEEE 802.11ac RATs. The number of vehicles connected to IEEE 802.11ac RAT based on RSSI increases from 2 at IEEE 802.11n RAT to 32 at IEEE 802.11ac RAT due to its intermediate coverage between IEEE 802.11n RAT and ad

RAT, along with enhanced directive beamforming capabilities.

Consequently, despite the reduced coverage of ac RAT compared to IEEE 802.11n RAT, the number of vehicles connected to it based on TTL decreases, while the number of vehicles connected based on RSSI increases due to improved beamforming.

Similar trends are observed for IEEE 802.11ad RAT, characterized by superior directive beamforming compared to IEEE 802.11ac RAT, and it has the lowest coverage within the network. Consequently, the number of vehicles connected to the ad RAT decreases from 32 to 16 based on RSSI and from 6 to 4 based on TTL. The slight decrease observed in the TTL scenario is attributed to the relatively similar coverage between IEEE 802.11ac and IEEE 802.11ad RAT.

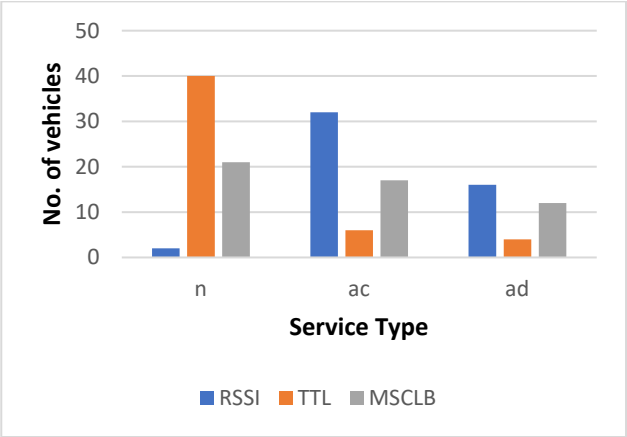



Fig. 10. No. of vehicles connected to each RAT using RSSI, TTL, and MSCLB.

This paper's main contribution is to propose an LB scenario based on MSCLB for each RAT. As previously outlined, caching uses three proposed services: safety, traffic, and information. This suggested scenario enables vehicle connectivity to each available RAT in the network based on the specific requirements of each vehicle. As shown in Figure 10, the LB from applying the proposed MSCLB is distributed along all RATs.

For instance, a vehicle at an intersection prioritizes safety service from the nearest RAT over information service, whereas a vehicle traveling on an open road with no imminent danger prioritizes information service over safety. Additionally, a vehicle passing another vehicle necessitates traffic service to stay informed about road conditions. Fig. 11 illustrates the LB scenario based on caching among the three available RATs. It is evident that n RAT, owing to its extensive coverage, accommodates the highest number of connected vehicles needing safety services. This is logical as safety services require broad coverage for simple connectivity, a feature provided by n RAT despite not requiring high data rates.

The ac network lies midway between n RAT and ad RAT, accommodating traffic services that some vehicles require. In this scenario, instead of n RAT bearing the burden of vehicles requiring traffic services exclusively, the load is distributed to the IEEE 802.11ac RAT thanks to its higher data rate capabilities. Information services are predominantly hosted by ad RAT, with an increasing demand for traffic services.

This alignment is highly advantageous since information services necessitate high data rates, which ad RAT can provide. LB in the caching scenario facilitates 1) distributing the load across all networks based on the required service, 2) allowing space for more vehicles to connect to each RAT based on service, and 3) offloading connections from RATs once the service requirement is fulfilled, thereby minimizing load on each RAT in a shorter duration.

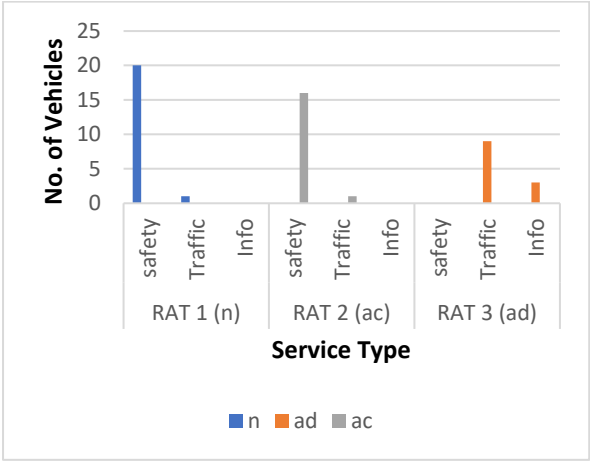



Fig. 11. LB by service between vehicles based on MSCLB.

### B. LB Throughput

Fig. 12 depicts the network throughput for each LB scenario concerning all RATs. Fig. 12(a) shows the throughput for the n RAT. The optimal throughput, at 88 Mbps, is achieved in the MSCLB. Across all scenarios, throughput diminishes as vehicle speeds increase, thereby elevating the connection and disconnection rates within the network.

A significant observation from Fig. 12(a) is the considerable degradation in throughput in the TTL scenario, dropping to 46 Mbps, notably lower than the throughput in the RSSI scenario. This decline is logical as the increased vehicle speed results in more vehicles leaving the RAT coverage area, leading to a degradation in throughput. It's important to note that the majority of vehicles connected to n RAT are primarily for safety services, which do not consume substantial throughput available.

Fig. 12(b) illustrates the throughput of the ac RAT, revealing an uptick in throughput from the TTL scenario and a corresponding decrease in throughput from the RSSI scenario. The optimal throughput, reaching 498 Mbps, is observed in the caching scenario. This enhancement stems from the LB strategy implemented for the ac RAT across each scenario. In the RSSI scenario, there's an increase in the number of vehicles connected to the ac RAT, as depicted in Fig. 10, resulting in reduced required throughput.

Conversely, in the TTL scenario, there's a decrease in the number of vehicles connected, leading to an increase in throughput provided by the RAT. In the caching scenario, the number of connected vehicles increases, as shown in Fig. 11, primarily for traffic services, thus yielding the highest throughput. Notably, the disparity between the TTL and caching scenarios is merely 10 Mbps, attributable to the demand for traffic services and some safety services from this RAT.

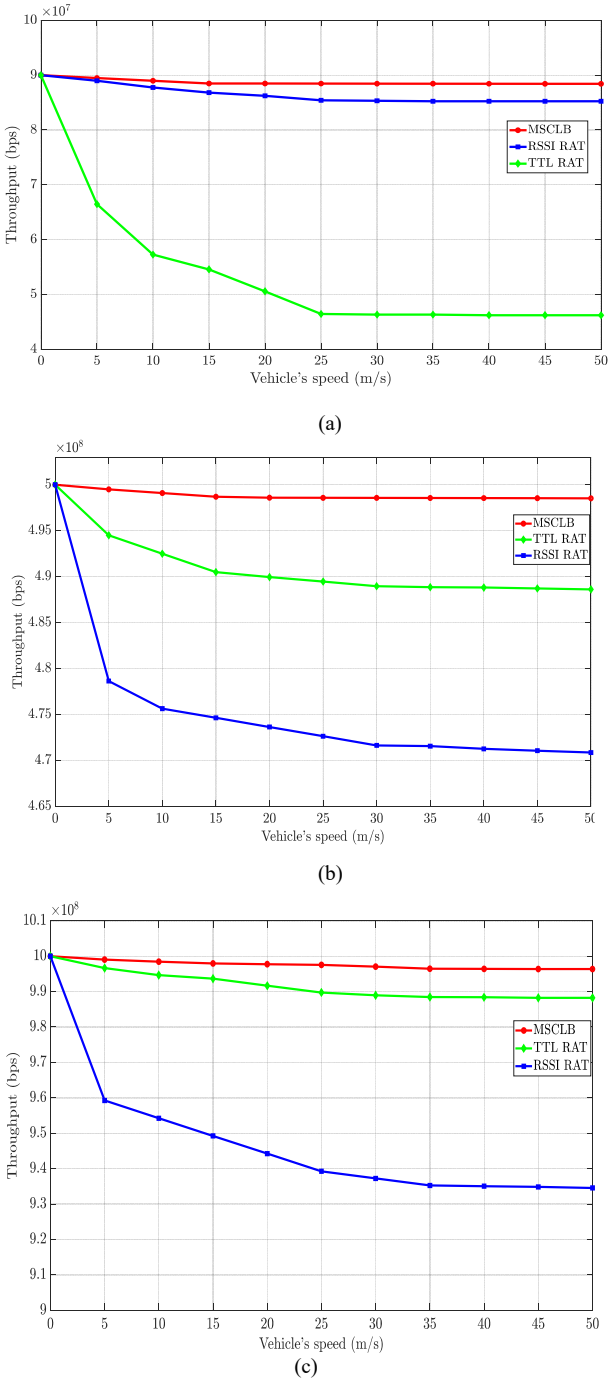



Fig. 12. Total throughout for (a) IEEE 802.11n: (b) IEEE 802.11ac, (c) IEEE 802.11ad.

Fig. 12 (c) illustrates the throughput for the ad network. A significant observation is the minimal disparity of 10 Mbps between the throughput from the TTL and caching scenarios. This can be attributed to this RAT type's high directivity and data rate. As vehicle speeds increase, the time available for connectivity diminishes, prompting new vehicles to connect to RATs offering the required services. Vehicles traveling at high speeds prioritize information services over traffic or safety services. Consequently, they tend to connect to ad networks primarily for information services. As a result, the number of vehicles connected to RAT n decreases, reducing throughput, as depicted in Fig. 12(c).

### *C. LB PDR*

Fig. 13 (a) illustrates the resultant PDR stemming from communication utilizing n RAT. It's important to note that the bandwidth of n RAT is limited and experiences degradation as the number of vehicles connected to it rises. Consequently, the connection scenario based on RSSI falls between the TTL and caching scenarios. This is because the wide coverage of this RAT leads to a higher number of connected vehicles than those solely based on RSSI.

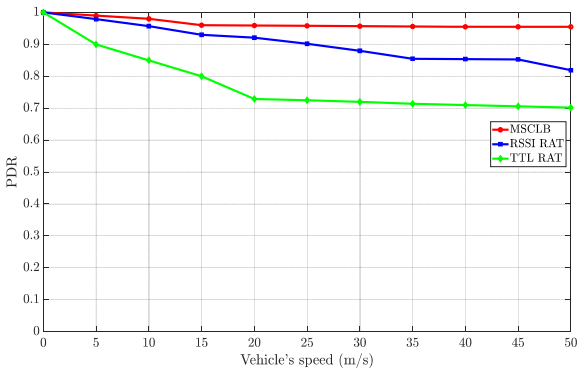
The fewer connected vehicles in this RAT scenario facilitate data exchange with fewer errors, enhancing the PDR. The caching scenario demonstrates the highest PDR, approximately 0.99, as vehicles are evenly distributed across the service, meeting each vehicle's demand.

Fig. 13 (b) depicts the PDR originating from the ac RAT. The PDR drops to 0.69 in the caching scenario compared to the value obtained from the n RAT. This decline can be attributed to the increased demand for traffic and informational data in this RAT, facilitated by its broader bandwidth. Consequently, the rise in vehicles accessing this RAT leads to higher data delivery errors.

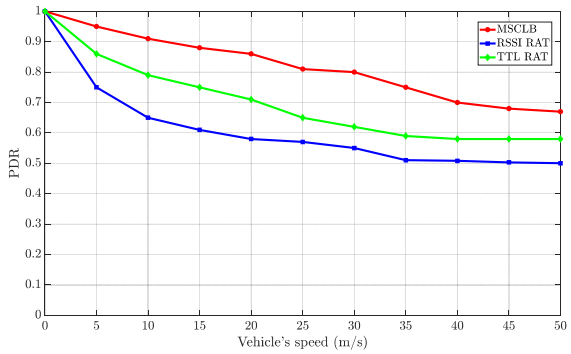
Another noteworthy observation from this RAT is that the RSSI scenario yields the lowest PDR value, approximately 0.5, as the number of vehicles accessing this RAT increases, and each vehicle's needs transition from safety services to traffic and informational data.

The same conclusion drawn from the PDR analysis of the IEEE 802.11ac RAT also applies to the IEEE 802.11ad RAT, as depicted in Fig. 13(c). This similarity arises from both RATs offering ample bandwidth for fulfilling traffic and informational needs. However, the distinction lies in the higher PDR ratio observed in the caching scenario due to the broader bandwidth provided by the IEEE 802.11ad RAT compared to the IEEE 802.11ac RAT. In this scenario, the PDR ratio increases to 0.88, representing a 19% increase over the PDR ratio of the IEEE 802.11ac RAT connection. Another contributing factor to this outcome is the high directivity of the ad RAT beam, resulting in reduced interference and transmission errors.

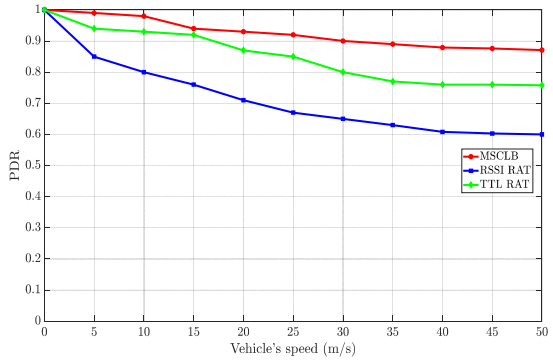
It is clear that the PDR follows a consistent trend across RATs: the RSSI-based method exhibits the lowest PDR, TTL shows moderate performance, and the proposed model attains the highest PDR. Notably, IEEE 802.11ad achieves a slightly higher PDR than IEEE802.11ac but remains below IEEE802.11n, reflecting its directional nature and increased susceptibility to noise and interference.




(a)




(b)




(c)

Fig. 13. Total PDR for (a) IEEE 802.11n, (b) IEEE 802.11ac, (c) IEEE 802.11ad.

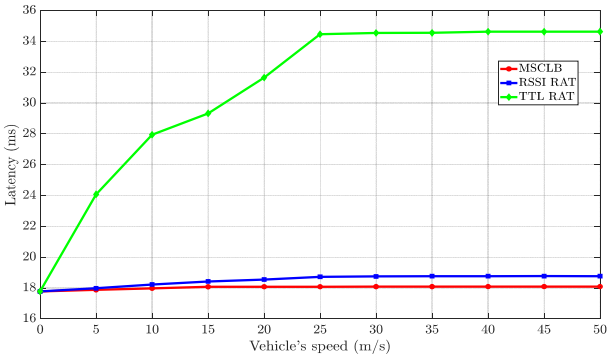
### *D. LB Latency*

Regarding latency, Fig. 14(a) illustrates the overall latency experienced by the n RAT in managing vehicle connections. As vehicles move farther from the RAT, propagation delay increases, resulting in higher latency. This explains why the TTL scenario exhibits the highest latency compared to other scenarios [40].

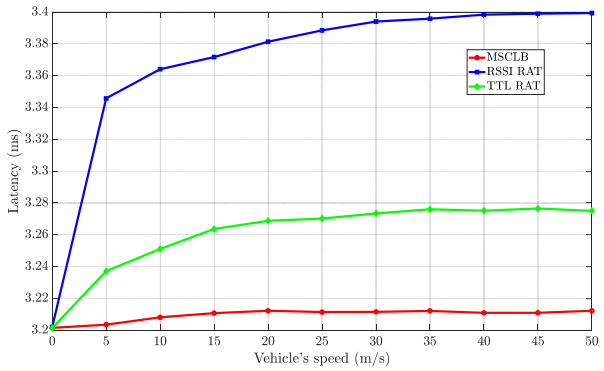
Conversely, the caching scenario demonstrates the lowest latency due to minimal propagation and transmission delays. These reduced latency values stem from the RAT connection being determined by service requirements rather than RSSI or distance from the RAT, as in the TTL scenario. The latency associated with the caching scenario is 18 milliseconds.

In Fig. 14(b), enhancing the beamwidth directivity, communication speed, and coverage area of the ac RAT results in decreased latency. Despite these improvements, latency remains lowest in the caching scenario at 3.21 milliseconds compared to the n RAT. Additionally, the TTL latency decreases to 3.26 milliseconds due to the reduced coverage of the ac RAT.

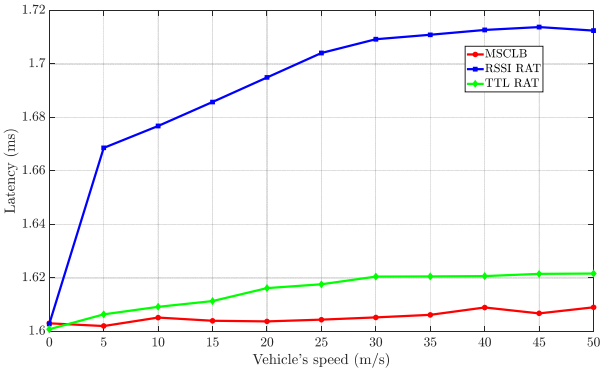
Fig. 14© illustrates the overall latency analysis for the IEEE 802.11ad RAT. Similar conclusions can be drawn from the latency findings of the ac RAT, albeit with reduced latency values, particularly evident in the caching scenario where latency decreases to 1.61 compared to that of the ac RAT in Fig. 14(b). This reduction can be attributed to the decrease in the coverage area of the IEEE 802.11ad RAT and the simultaneous increase in data rate and directivity.




(a)




(b)




(c)

Fig. 14. Total latency for (a) IEEE 802.11n (b) IEEE 802.11ac (c) IEEE 802.11ad.

### *E. IMSCLB Evaluation*

Fig. 15 shows the satisfaction rate between IMSCLB and MSCLB. The IMSCLB outperforms the MSCLB because of the prediction capability of the IMSCLB to

predict the throughput required for each RAT at each new time. This enhances the overall throughput value of each RAT, which means enhancing the LB in the network and enhancing the ability to serve more vehicles simultaneously, especially in a highly dense network.

Fig. 16 depicts the IMSCLB scheme employed for LB. The figure indicates that the CNN-LSTM model effectively predicted the load on each RAT, achieving an RMSE value of 0.213. This value falls below 4, a threshold commonly acknowledged as indicative of predictive solid performance in related literature. With a RMSE of less than 1, the prediction accuracy notably improves.

The proposed MSCLB and IMSCLB schemes demonstrate superior performance compared to the SOLMC RAT selection approach [12]. While SOLMC achieves up to 47.5% improvement in network throughput and 20.42% enhancement in packet delivery ratio over nearest-RAT selection, MSCLB provides higher throughput across all RATs, reaching 88 Mbps for IEEE 802.11n RAT and 498 Mbps for IEEE 802.11ac RAT, with a PDR up to 0.99. The IMSCLB scheme further improves these metrics by incorporating predictive load balancing using a CNN-LSTM model, achieving an 8% throughput increase over MSCLB, lower latency, and more balanced service-specific load distribution. These results indicate that IMSCLB efficiently manages multi-RAT, multi-service vehicular traffic, offering more reliable and high-performance V2I communication under high-mobility conditions.

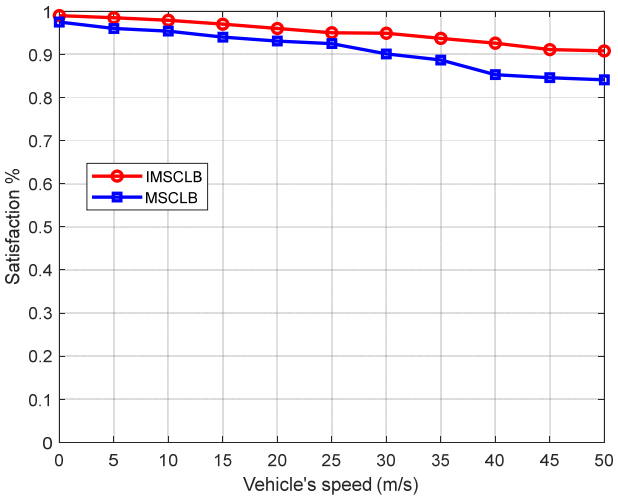



Fig. 15. Satisfaction rate between IMSCLB and MSCLB.

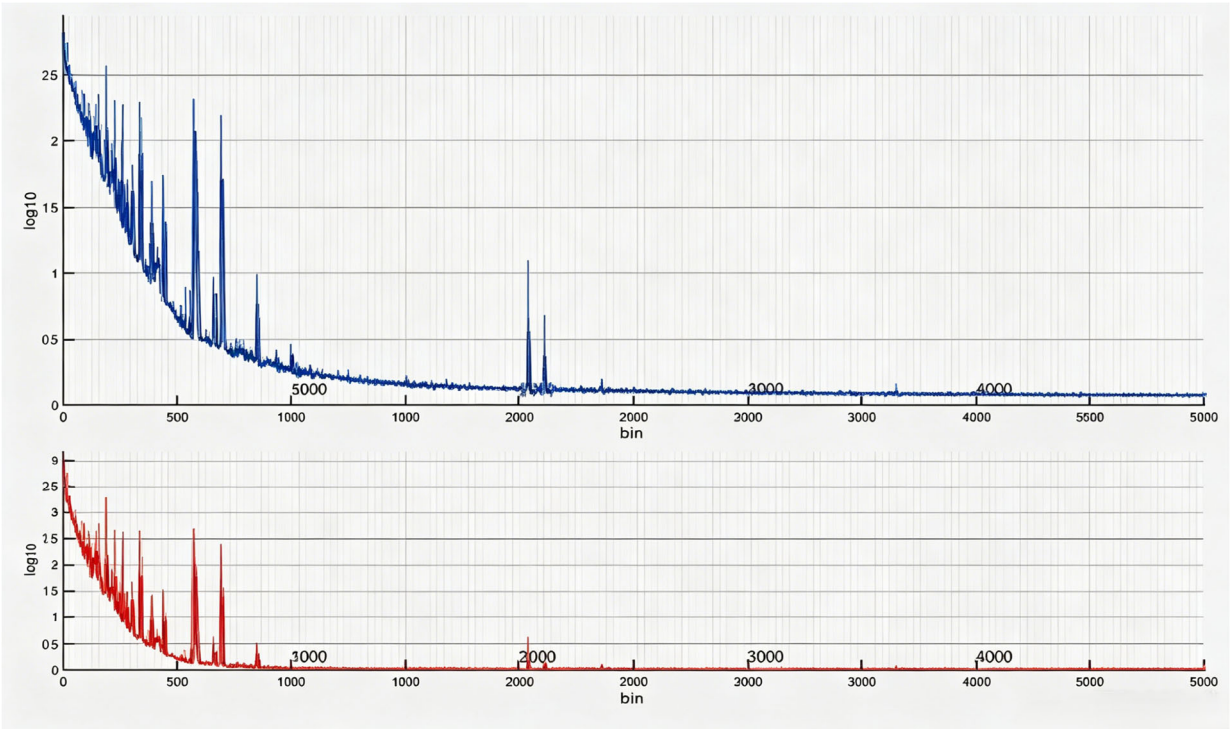



Fig. 16. RMSE for the training data.

The computational complexity of the proposed IMSCLB scheme arises from two main components: the CNN-LSTM predictor and the caching-based RAT selection. For the CNN, with $L_c$ convolutional layers each having $F_l$ filters of size $K_l \times K_l$ and input channels $C_{l-1}$, the FLOPs are calculated as:

$$\text{FLOPs}_{\text{CNN}} = \sum_{l=1}^{L_c} H_l W_l F_l K_l^2 C_{l-1} \quad (25)$$

where $H_l$ and $W_l$ are the height and width of the feature map at layer $l$. For the LSTM with $T$ time steps, $n_{\text{in}}$ input features, and $n_h$ hidden units, the FLOPs per step are:

$$\text{FLOPs}_{\text{LSTM}} = 4 \times n_h \times (n_{\text{in}} + n_h + 1) \quad (26)$$

yielding a total for all time steps:

$$\text{FLOPs}_{\text{LSTM-total}} = T \times 4 \times n_h \times (n_{\text{in}} + n_h + 1) \quad (27)$$

Thus, the total FLOPs for the predictor are:

$$\text{FLOPs}_{\text{predictor}} = \text{FLOPs}_{\text{CNN}} + \text{FLOPs}_{\text{LSTM-total}} \quad (28)$$

For the caching-based RAT selection, assuming $K$ vehicles, $N$ RATs, and $SE$ services, the complexity scales linearly as:

$$\text{FLOPs}_{\text{RAT-selection}} = O(K.N.SE). \quad (29)$$

Consequently, the overall IMSCLB complexity is:

$$\text{FLOPs}_{\text{IMSCLB}} = \text{FLOPs}_{\text{predictor}} + \text{FLOPs}_{\text{RAT-selection}} \quad (30)$$

This is significantly more efficient than conventional optimization-based LB schemes, which can have exponential complexity $O(N^K \times I)$ for $I$ iterations. Therefore, IMSCLB offers scalable real-time performance while maintaining near-optimal load balancing across multi-RAT vehicular networks.

## V. Conclusion

This paper presents and validates the IMSCLB scheme to address the simulated congestion and load balancing challenges in multi-RAT V2X communication. By dynamically selecting the optimal RAT based on the selected service needs and integrating service-aware caching with a CNN-LSTM model, IMSCLB improves network QoS and achieves an 8% throughput gain over MSCLB. The study faced limitations due to the lack of real-world datasets, requiring extensive synthetic data generation and validation, the need to balance accuracy with real-time processing, the scalability to larger vehicular networks, simplifying assumptions in mobility modeling, and the practical feasibility. Future work will focus on overcoming these limitations, exploring non-line-of-sight communication for greater robustness, and extending the model to support cooperative V2V load balancing for enhanced network resilience.

## Conflict of Interest

The authors declare no conflict of interest.

## Author Contributions

Mohammed Mudhafar Shakir conceived and designed the research, developed the simulation methodology, performed the experiments, and analyzed the results; Lukman Audah and Roshayati Yahya supervised the work, guided the research direction, validated the results, and administered the project; Mohammed A. Altahrawi contributed to manuscript drafting and figure preparation. Abdinasir Hirsi assisted with literature review and data analysis; Abdullahi Farah supported simulation implementation and performance evaluation; all authors reviewed and approved the final version of the manuscript.

## Funding

Communication of this research is made possible through monetary assistance by Universiti Tun Hussein Onn Malaysia and the UTHM Publisher's Office via Publication Fund E15216.

## References

[1] U. Ahmed, J. C.-W. Lin, G. Srivastava, U. Yun, and A. K. Singh, "Deep active learning intrusion detection and load balancing in software-defined vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 953−961 2022.

[2] K. Hejja, S. Berri, and H. Labiod, "Network slicing with load-balancing for task offloading in vehicular edge computing," *Vehicular Communications*, vol. 34, 100419, 2022.

[3] S. Moon and Y. Lim, "Task migration with partitioning for load balancing in collaborative edge computing," *Applied Sciences*, vol. 12, no. 3, 1168, 2022.

[4] U. Ahmed, J. C. W. Lin, and G. Srivastava, "A resource allocation deep active learning based on load balancer for network intrusion detection in SDN sensors," *Computer Communications*, vol. 184, pp. 56−63, 2022.

[5] M. H. Kashani and E. Mahdipoussr, "Load balancing algorithms in fog computing: A systematic review," *IEEE Transactions on Services Computing*, no. 1, p. 1, 2022.

[6] V. Balakrishnan, Z. Shi, C. L. Law, R. Lim, L. L. Teh, and Y. Fan, "A deep learning approach in predicting products' sentiment ratings: a comparative analysis," *The Journal of Supercomputing*, vol. 78, no. 5, pp. 7206−7226, 2022.

[7] E. Gures, I. Shayea, M. Ergen, M. H. Azmi, and A. A. E. Saleh, "Machine learning based load balancing algorithms in future heterogeneous networks: A survey," *IEEE Access*, vol. 10, pp. 37689−37717, 2022.

[8] D. Zhai, H. Li, X. Tang, R. Zhang, and H. Cao, "Joint position optimization, user association, and resource allocation for load balancing in UAV-assisted wireless networks," *Digital Communications and Networks*, vol. 10, no. 1, pp. 25−37. 2022.

[9] R. G. Lazar, A. V. Militaru, C. F. Caruntu, C. Pascal, and C. P. Sultanoiu, "Real-time data measurement methodology to evaluate the 5G network performance indicators," *IEEE Access*, vol. 11, pp. 43909−43924. 2023.

[10] K. Tan, D. Bremner, J. L. Kernec, L. Zhang, and M. Imran, "Machine learning in vehicular networking: An overview," *Digital Communications and Networks*, vol. 8, no. 1, pp. 18−24, 2022.

[11] H. Guo, X. Zhou, Y. Wang, and J. Liu, "Achieve load balancing in multi-UAV edge computing IoT networks: A dynamic entry and exit mechanism," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18725−18736, 2022.

[12] M. Altahrawi, N. F. Abdullah, and R. Nordin, "Service-oriented lstm multi-criteria rat selection scheme for vehicle-to-infrastructure communication," *IEEE Access*, vol. 10, pp. 110261−110284, 2022.

[13] X. Deng *et al*., "An ant colony optimization-based routing algorithm for load balancing in leo satellite networks," *Wireless Communications and Mobile Computing*, vol. 11, pp. 43909−43924, 2022.

[14] G. Husnain and S. Anwar, "An intelligent probabilistic Whale Optimization Algorithm (i-WOA) for clustering in vehicular ad hoc networks," *International Journal of Wireless Information Networks*, vol. 29, no. 2, pp. 143−156, 2022.

[15] H. C. Fu and J. J. Hao, "Efficient RSU selection approaches for load balancing in vehicular ad hoc networks," *Advances in Technology Innovation*, vol. 5, no. 1, 2020.

[16] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4377−4387, 2018.

[17] C. M. Huang, M. S. Chiang, D. T. Dao, W. L. Su, S. Xu, and H. Zhou, "V2V data offloading for cellular network based on the Software Defined Network (SDN) inside Mobile Edge Computing (MEC) architecture," *IEEE Access*, vol. 6, pp. 17741−17755, 2018.

[18] H. Ye and G. Y. Li, "Deep reinforcement learning for resource allocation in V2V communications," in *Proc. 2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1−6.

[19] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 3, pp. 1657−1681, 2017.

[20] B. Toghi *et al*., "A maneuver-based urban driving dataset and model for cooperative vehicle applications," in *Proc. 2020 IEEE 3rd Connected and Automated Vehicles Symposium (CAVS)*, 2020, pp. 1−6.
[21] C. M. Huang and C. F. Lai, "The Mobile Edge Computing (MEC)-based Vehicle to Infrastructure (V2I) data offloading from cellular network to vanet using the delay-constrained computing scheme," in *Proc. 2020 International Computer Symposium (ICS)*, 2020, pp. 1−6.
[22] H. Zhang, Z. Wang, and K. Liu, "V2X offloading and resource allocation in SDN-assisted MEC-based vehicular networks," *China Communications*, vol. 17, no. 5, pp. 266−283, 2020.
[23] S. Sondur, K. Kant, S. Vucetic, and B. Byers, "Storage on the edge: Evaluating cloud backed edge storage in cyberphysical systems," in *Proc. 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2019, pp. 362−370.
[24] G. Wu *et al*., "Meccas: Collaborative storage algorithm based on alternating direction method of multipliers on mobile edge cloud," in *Proc. 2017 IEEE International Conference on Edge Computing* (EDGE), 2017, pp. 40−46.
[25] S. Chen, Z. Chen, S. Gu, B. Chen, J. Xie, and D. Guo, "Load balance awared data sharing systems in heterogeneous edge environment," in *Proc. 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, 2020, pp. 132−139.
[26] A. Khan, A. Muhammad, Y. Kim, S. Park, and B. Tak, "EDGESTORE: A single namespace and resource-aware federation file system for edge servers," in *Proc. 2018 IEEE International Conference on Edge Computing (EDGE)*, 2018, pp. 101−108.
[27] R. Beraldi, A. Mtibaa, and H. Alnuweiri, "Cooperative load balancing scheme for edge computing resources," in *Proc. 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, 2017, pp. 94−100.
[28] C. H. Hsu, Y. Chiang, Y. Zhang, and H. Y. Wei, "Mobility-aware QoS promotion and load balancing in MEC-based vehicular networks: A deep learning approach," in *Proc. 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, 2021, pp. 1−6.
[29] F. Zhang and M. M. Wang, "Stochastic congestion game for load balancing in mobile-edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 778−790, 2020.
[30] V. S. Varanasi and S. Chilukuri, "Adaptive differentiated edge caching with machine learning for V2X communication," in *Proc. 2019 11th International Conference on Communication Systems and Networks (COMSNETS)*, 2019, pp. 481−484.
[31] Y. Zhang, C. Li, T. H. Luan, Y. Fu, W. Shi, and L. Zhu, "A mobility-aware vehicular caching scheme in content centric networks: Model and optimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3100−3112, 2019.
[32] R. Meireles, A. Rodrigues, A. Stanciu, A. Aguiar, and P. Steenkiste, "Exploring Wi-Fi network diversity for vehicle-to-infrastructure communication," in *Proc. 2020 IEEE Vehicular Networking Conference (VNC)*, 2020, pp. 1−8.
[33] P. Sharma, P. Dadheech, N. Aneja, and S. Aneja, "Predicting agriculture yields based on machine learning using regression and deep learning," *IEEE Access*, vol. 11, pp. 111255−111264, 2023.
[34] A. O. Almagrabi *et al.*, "A poisson process-based random-access channel for 5G and beyond networks," *Mathematics*, vol. 9, no. 5, p. 508, 2021.
[35] J. Han, X. Wang, and G. Wang, "Modeling the car-following behavior with consideration of driver, vehicle, and environment factors: A historical review," *Sustainability*, vol. 14, no. 13, 8179, 2022.
[36] K. Zhu *et al.*, "A heterogeneity-aware car-following model: Based on the XGBoost method," *Algorithms*, vol. 17, no. 2, p. 68, 2024.
[37] J. Guo, Y. Zhang, X. Chen, S. Yousefi, C. Guo and Y. Wang, "Spatial stochastic Vehicle traffic modeling for VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 416−425, Feb. 2018.