Computer Modeling in
Engineering & Sciences

Tech Science Press

Doi:10.32604/cmes.2025.069430

ARTICLE

An Efficient Content Caching Strategy for Fog-Enabled Road Side Units in Vehicular Networks

Faareh Ahmed[1], **Babar Mansoor**[2], **Muhammad Awais Javed**[3] **and Abdul Khader Jilani Saudagar**[4,*]

[1]School of Electrical Engineering and Computer Sciences, National University of Sciences and Technology, Islamabad, 44000, Pakistan
[2]Department of Computer Engineering, COMSATS University, Islamabad, 45550, Pakistan
[3]Department of Electrical Engineering, COMSATS University, Islamabad, 45550, Pakistan
[4]Information Systems Department, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, 11432, Saudi Arabia

*Corresponding Author: Abdul Khader Jilani Saudagar. Email: aksaudagar@imamu.edu.sa

Received: 23 June 2025; Accepted: 27 August 2025; Published: 30 September 2025

ABSTRACT: Vehicular networks enable seamless connectivity for exchanging emergency and infotainment content. However, retrieving infotainment data from remote servers often introduces high delays, degrading the Quality of Service (QoS). To overcome this, caching frequently requested content at fog-enabled Road Side Units (RSUs) reduces communication latency. Yet, the limited caching capacity of RSUs makes it impractical to store all contents with varying sizes and popularity. This research proposes an efficient content caching algorithm that adapts to dynamic vehicular demands on highways to maximize request satisfaction. The scheme is evaluated against Intelligent Content Caching (ICC) and Random Caching (RC). The obtained results show that our proposed scheme entertains more content-requesting vehicles as compared to ICC and RC, with 33% and 41% more downloaded data in 28% and 35% less amount of time from ICC and RC schemes, respectively.

KEYWORDS: Vehicular networks; fog computing; content caching; infotainment services

1 Introduction

Intelligent Transportation Systems (ITS) in vehicular networks have revolutionized the transportation system by providing a transformative approach to enhance safety, efficiency, and sustainability. The applications of ITS are further enhanced by incorporating modern technological advancements such as seamlessly connected Internet of Things (IoT), and by utilizing 5G and 6G communications [1,2]. Software Defined Networks (SDN) and Network Function Virtualization (NFV) are the key technologies that are used to improve the security, scalability, and traffic management in vehicular networks. SDN uses software-based controllers and Application Programming Interfaces (APIs) for scalability and traffic management. However, NFV provides a concept of virtual roadside units by reducing the reliance on specialized hardware. The combination of SDN and NFV meets the dynamic and demanding requirements of vehicular networks.

Vehicular networks in ITS mainly provide three types of communications, i.e., Vehicle-to-Vehicle (V2V), Vehicle-to-infrastructure (V2I), and Vehicle-to-everything (V2X) [3–5]. Among these, V2V communications are primarily utilized for safety-related applications, whereas V2I and V2X communications are mostly used for traffic management and infotainment services. Vehicular networks facilitate communication between vehicles to provide warnings about potential hazards, such as accidents, road conditions, or obstacles. By sharing traffic information, vehicles can optimize their routes to avoid traffic congestion, resulting in an improved and smoother traffic flow with reduced travel time. In case of emergencies, such as accidents or medical emergencies, vehicular networks are used to quickly alert emergency services and provide them with the necessary information. Vehicular networks also offer drivers and passengers access to real-time information, such as weather updates, nearby points of interest, and entertainment options.

Infotainment services provide a comfort level to drivers and passengers and make their journeys more pleasurable. Generally, the usual requirements of infotainment applications are better network connectivity, availability, and reliability, ensuring that the required contents are delivered to the commuter instantly. Infotainment services include providing the location of nearby hotels, weather forecasts, and live streaming services. These infotainment services, along with traffic management services, are provided and controlled by the Traffic Command Center (TCC).

TCC consists of multiple servers placed at remote locations that will provide information and services to vehicles [6–8]. TCC is wirelessly connected to all the Road Side Units (RSUs) placed along the roadside. All vehicles in the vehicular network throughout their journey are wirelessly connected with one of these RSUs through their onboard units. RSUs forward the road conditions along with vehicle infotainment requests to TCC. Popular contents are stored on multiple caching servers in the TCC location. The demanded infotainment content is forwarded to vehicles by TCC, which is mostly located in metropolitan areas. Fetching the required contents from the remotely placed TCC offers a low data rate and often causes interrupted downloading, which results in irritation at the vehicle end.

To overcome these interruptions in content downloading, these contents are required to be cached in a nearby location with a higher data rate offering, such as a fog node [9]. Fog computing encompasses essential services from the network centre to the network edge, which can effectively resolve issues such as bandwidth constraints, low latency, and energy consumption instigated by the prolonged distances [10]. Since vehicles remain connected to one of the RSUs during travelling, therefore, placing the caching server at the location of the RSU enhances the data rate of the vehicles in downloading their requested data contents.

Due to the limited storage capacity of fog nodes, it is not feasible to cache all content requested by vehicles. To maximize data availability and reduce download delays, it is essential to cache only the most popular content at nearby fog-enabled Road Side Units (RSUs). However, content demand varies significantly across geographic regions depending on vehicle density and speed, making efficient content delivery a challenging task.

This work aims to enhance Quality of Service (QoS) by minimizing content access delays. To achieve this, we propose an efficient content caching scheme that stores the most frequently requested data on fog-enabled RSUs, thereby improving download performance and reducing latency. The key contributions of our approach are:

- A content placement mechanism is designed using the 0/1 knapsack algorithm to optimally cache popular contents on RSUs within their limited storage capacity, maximizing cache utility and minimizing content access delay.
- To handle dynamically changing vehicle densities and content requests across RSUs, a clustering-based mechanism is introduced. It ensures that popular contents are intelligently replicated across multiple RSUs to meet future requests, thereby improving hit ratio and reducing redundant data retrieval from remote servers.
- The proposed scheme incorporates a practical highway model with variable RSU coverage, vehicle speeds, and content request patterns, capturing real-world vehicular mobility and network constraints.

The structure of the remaining paper is as follows:

Section 2 presents recent research on content caching in vehicular networks. Section 3 outlines the system model, while the proposed scheme is detailed in Section 4. Simulation, results, and comparative analysis are covered in Section 5, and Section 6 provides the conclusion for the manuscript.

2 Related Work

In order to enhance the caching mechanism in vehicular and mobile networks, several techniques have been proposed in the existing literature. In this regard, the majority of the existing work aims at improving the caching process by effective content placement based on its popularity.

In [11], authors have proposed a content caching technique for the case of Unmanned Aerial Vehicle (UAV) networks. In the proposed technique, UAVs are employed as service providers for the ground station while caching the most popular content. A notable attribute of the proposed work is to take into account the geographic popularity of the content in a specific time duration, thus resulting in enhanced access to the content and a significant reduction of the overall delay in content retrieval.

By exploiting the transportation correlation analysis, a cooperative content caching technique has been proposed by the authors in [12] for vehicular edge networks. The problem of service caching is addressed by leveraging the surrounding function-features of vehicles and transportation correlations. For the considered scenario, authors have utilized the surrounding function-features to estimate content preferences while ensuring a high hit-ratio for cached contents. Moreover, the vehicular service caching has been framed as a constrained optimization problem and a cooperation mechanism between RSUs has been devised by exploiting the transportation correlations of the vehicle trajectories. Furthermore, in order to mitigate the negative effects of vehicle mobility and to optimize the network delay, Gibbs sampling has also been incorporated for the vehicular service caching problem.

Authors in [13] have proposed two unique data caching techniques for Internet-of-Vehicles (IoV) by grouping data into infotainment and safety categories. Based on the distinctive nature of both categories, separate caching methods are devised for each. The infotainment contents are cached by utilizing the federated learning based mobility-aware collaborative content caching, whereas spatio-temporal characteristics-aware based emergency content caching is devised for emergency content. Moreover, in order to anticipate the content popularity, authors have incorporated a long short-term memory model trained with federated learning to guarantee the user's privacy.

In order to effectively handle the problem of frequent variations in vehicular edge networks as of vehicle's mobility, causing repeated link interruptions and increased transmission delays, authors in [14] have devised an adaptive cooperative content caching strategy for vehicular edge networks. In the proposed scheme, an optimization problem has been devised aiming to minimize the transmission delay. Moreover, authors have also proposed two low-complexity based greedy algorithms to provide approximate solutions for optimal caching decisions.

Authors in [15] have employed an Artificial Intelligence (AI) based scheme for cache enhancement in vehicular networks. In order to achieve this, a deep reinforcement learning-based approach is incorporated to obtain an optimal policy function for improving the caching mechanism and reducing the transmission delay. In [16], authors have taken into consideration a scenario of satellite-aided UAVs and vehicular networks such that a geosynchronous low Earth orbit (GEO) satellite is utilized as cloud servers whereas UAVs is employed as edge-cache servers. The proposed work claims to significantly enhance the multi-casting opportunities, reduce the backhaul transmission volume, as well as energy consumption in the considered scenario by incorporating an energy-aware coded technique for content caching.

In [17], authors have devised a forecasting scheme to predict the mobility, and content caching trend of vehicles by incorporating a recurrent neural network classifier in conjunction with the elliptic curve cryptography method. In the proposed technique, vehicles are further partitioned into clusters for content caching management. Based upon the obtained simulation results, the authors have claimed significant improvements in the content retrieval time.

In [18], authors addressed the high bandwidth and low latency demands of virtual reality (VR) content, particularly 360° video, by proposing a novel edge-assisted caching and delivery framework. Leveraging the sharable nature of VR content tiles, the framework integrates transcoding, cooperative caching using multi-agent reinforcement learning, and a two-tier NOMA-based multicast strategy. Their proposed techniques aim to optimize resource utilization and reduce service delay in next-generation wireless networks. The authors claimed through experimental results that their proposed scheme outperforms existing methods in cache hit rate, latency, and overall delivery efficiency.

In order to reduce the content downloading delay and enhance the caching hit ratio, authors in [19] have incorporated the notion of clustering among the vehicles. For this purpose, K-means algorithm is exploited to group the vehicles into clusters. In each cluster, a certain vehicle is designated as a cluster head and contents with high popularity are cached in that cluster head as well as base station. Consequently, a significant reduction in the content transmission distance is achieved by the proposed scheme.

Authors in [20] have devised a mobility-aware optimal content caching technique for vehicular networks. The proposed scheme focuses on the modelling and optimization of vehicular caching for the scenario of content-centric networks. In their proposed scheme authors have incorporated a two-dimensional Markov process for modelling the interactions between mobile users & caching vehicles such that the network availability for the mobile users is characterized. Moreover, from their deduced model, the authors have come up with an online caching strategy and enhanced network energy efficiency. In [21], a popularity-incentive-based content caching strategy has been proposed for the case of vehicular named data networks. In this scheme, those vehicles that offload their cache and share content with others are rewarded by the base station (BS).

In contrast to the existing work, our research aims to devise an effective and efficient content caching strategy for Fog-enabled RSUs in vehicular networks by taking into consideration the factual scenario of dynamic content requests with dynamic vehicle density and changes in the coverage area of RSUs.

Table 1 presents a summary of the related work and the proposed scheme, highlighting their key contributions and the corresponding network scenarios.

Table 1: Summary of the related work

Paper	Year	Contributions	Network scenario
[11]	2022	Popularity based content-caching by exploiting temporal and spatial attributes of contents	Content Caching for UAVs
[12]	2024	Cooperative content caching based on transportation correlations	Vehicular Edge Networks
[13]	2024	Federated learning on long short-term memory model	Internet of Vehicle
[14]	2024	Adaptive cooperative content caching	Vehicular Edge Networks

(Continued)

Table 1 (continued)

Paper	Year	Contributions	Network scenario
[15]	2023	Deep neural network based cooperative content caching and delivery	Content Caching in Vehicular Network
[16]	2022	Energy-aware coded caching strategy with resource optimization	Satellite-UAVs Assisted Vehicular
[17]	2023	Mobility-aware content caching using neural network	Content Caching in Vehicular Network
[18]	2022	Multi-agent reinforcement learning and two tier NOMA based group matching	360° panoramic video content in wireless network
[19]	2023	Collaborative caching using reinforcement learning	Clustered Vehicular Networks
[20]	2019	Mobility-aware caching scheme in Networks: Model and Optimization	Content Centric Vehicular Network
[21]	2022	Popularity-incentive based content caching	Vehicular Named Data
Proposed work	–	Efficient content caching strategy for Fog-enabled RSUs	Content Caching in Vehicular Network

3 System Model

The proposed caching framework is designed for a smart transportation environment, specifically targeting content delivery in highway or urban vehicular networks. In this scenario, a fleet of vehicles travels along road segments equipped with multiple fog-enabled RSUs. These RSUs are responsible for caching and delivering time-sensitive multimedia or control data, such as HD maps, traffic updates, or infotainment content, to the moving vehicles. Each RSU has limited cache and computing resources and must operate autonomously or cooperatively to optimize content delivery.

Vehicles generate dynamic and delay-sensitive content requests as they move through the RSU coverage areas. Due to high mobility and rapidly changing network topology, prefetching and efficient caching become crucial to reduce latency and backhaul load. The caching decision-making is managed using a two-level strategy:

- The 0/1 knapsack algorithm helps each RSU decide which contents to cache based on size and popularity under resource constraints.
- A clustering-based reallocation strategy allows RSUs to cooperatively respond to delayed requests by distributing content replicas intelligently among neighbors.

This framework ensures low-latency service, minimizes reliance on central servers, and adapts to dynamic traffic patterns. The system also includes a central Traffic Control Center that monitors content popularity trends and assists RSUs with updated content metadata and clustering information.

In this work, a two-way highway is connected with link roads from where vehicles are entering the highway. RSUs are installed along the highway to cover the whole highway. Vehicles communicate with RSUs through their Onboard Units (OBUs). All RSUs are strategically placed, and each one has a different coverage area due to changing geographical surroundings. All RSUs are wirelessly connected to vehicles in their coverage area and backwardly connected with TCC through wired connections, as shown in Fig. 1. TCC knows the cache capacity of each connected fog node along with its coverage area and the average channel condition of all vehicles in an RSU coverage area. Moreover, the TCC has information about the exact number of content requests received by each RSU during the past *t* time and is also supposed to know traffic densities in different durations of the day and weeks. Vehicles are supposed to download content directly from their attached RSU by using V2I communication only, and there is no content access from their neighbouring vehicles.

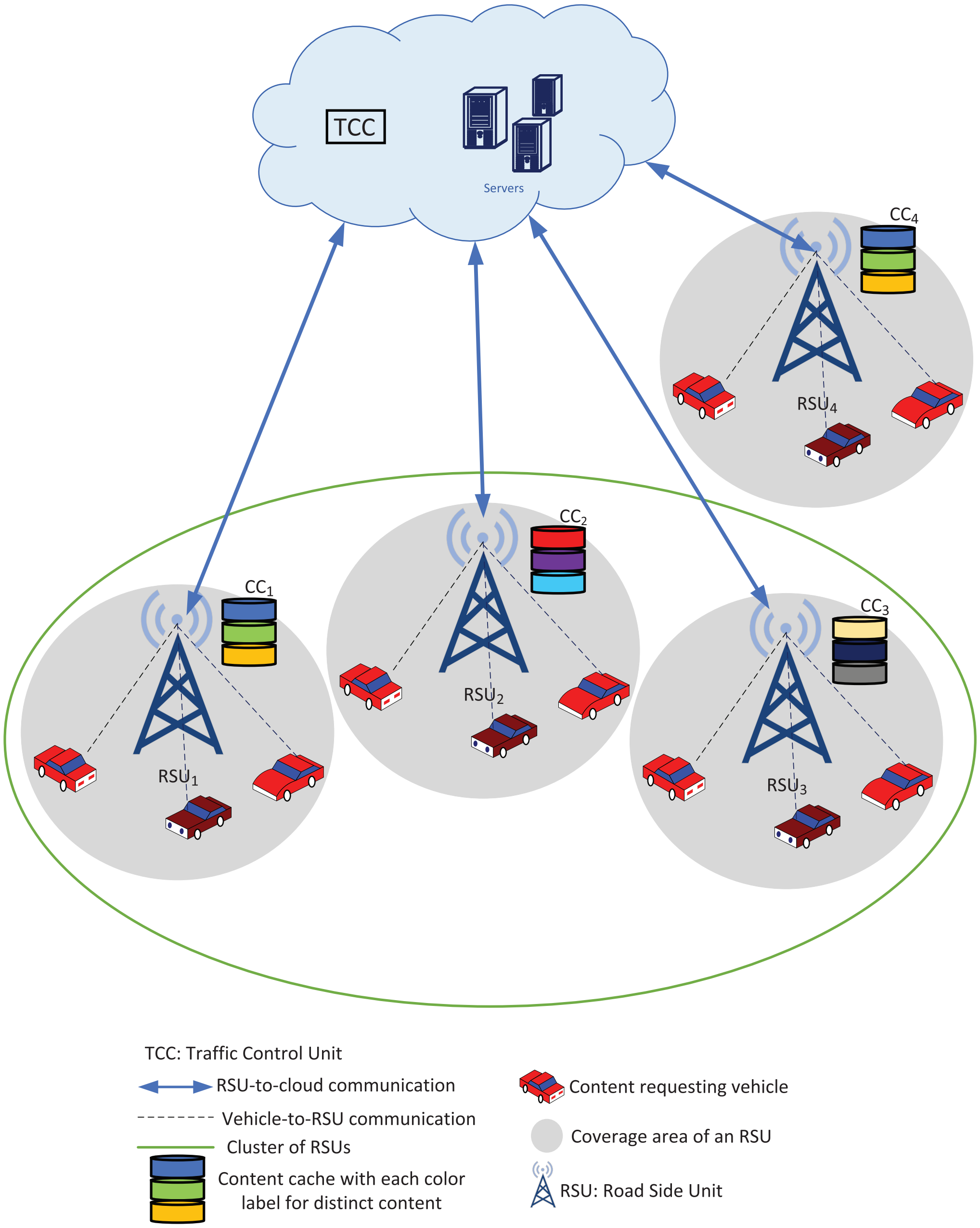

TCC
Servers
CC$_4$
RSU$_4$
CC$_2$
RSU$_2$
CC$_1$
RSU$_1$
CC$_3$
RSU$_3$
TCC: Traffic Control Unit
RSU-to-cloud communication
Vehicle-to-RSU communication
Cluster of RSUs
Content cache with each color label for distinct content
Content requesting vehicle
Coverage area of an RSU
RSU: Road Side Unit

Figure 1: System model of the proposed scheme

Suppose a total of R RSUs are placed on the highway so that the whole highway is under their coverage area. There is V number of vehicles that pass through their coverage areas in a given time t. V number of vehicles originated the contents requests (C_{req}) to each of its connected RSU with sizes CS_{req}. $C_{x,y}$ represents xth content of the yth vehicle and its size is represented by $CS_{x,y}$.

If V vehicles that requested a total of M contents then C_{req} and CS_{req} at a given time t are computed as:

$$C_{req} = \sum_{x=1}^{M}\sum_{y=1}^{V} C_{xy} \tag{1}$$

$$CS_{req} = \sum_{x=1}^{M}\sum_{y=1}^{V} CS_{xy} \tag{2}$$

It is obvious that all vehicles do not initiate their requests at the start of their journey, and some vehicles may initiate content requests at a later stage of the journey when they will be in the coverage area of the next RSU. In this work, we suppose that the major part of vehicles originate their request at the start of their journey, and the remaining initiates their content requests at a uniform rate when vehicles enter the upcoming RSUs' coverage areas. If V vehicles originate their requests for the content C and it increases at the rate of R in the next coverage area of RSUs then total content requests C_{tot} originated by vehicles after X number of RSUs is calculated as:

$$C_{tot} = V + (R \times X) \tag{3}$$

All fog-enabled RSUs are supposed to possess the same amount of content storing capacity, which is denoted by C_{Cap}. However, each fog-enabled RSU is allowed to cache those contents that fall within its storage limit. Suppose ith content (C_i) having size CS_i is required to be cached and R number of fog nodes possessing the maximum caching capability of Fog_{cache}. The total sum of contents C_{Alloc} that are allocated on R fog-enabled RSUs possessing sizes CS_{Alloc} are calculated by applying the following equations.

$$C_{Alloc} = \sum_{i=1}^{R} C_i \tag{4}$$

$$CS_{Alloc} = \sum_{i=1}^{R} CS_i \ldots | CS_{Alloc} \varepsilon CS_{req} \&\& CS_{Alloc} \leq Fog_{cache} \tag{5}$$

In vehicular networks, all vehicles are directly connected to one of the RSUs throughout. The duration of the connection of the vehicle with the RSU depends upon its velocity and the coverage area of the RSU. If V_n represents the velocity of the nth vehicle and ϕ_k represents the coverage area of the kth RSU, then the total duration t_n the vehicle remains in connection with the RSU is calculated as follows:

$$t_n = \phi_n \times v_n \tag{6}$$

Suppose there is m number of vehicles, that are within the coverage area of RSU ϕ_k simultaneously, the average duration spent by a vehicle (t_{ave}) along with the average velocity of a vehicle (V_{ave}) in the coverage area an RSU is calculated by applying Eqs. (7) and (8), respectively.

$$t_{ave} = \frac{\sum_{k=1}^{m} (\phi_k \times V_n)}{m} \tag{7}$$

$$v_{ave} = \frac{\sum_{n=1}^{m}(V_n)}{m} \tag{8}$$

The primary factors affecting the time it takes for a vehicle to download content from remote servers depend upon the content size as well as the data rate in downloading the content. Suppose DR1 represents the data rate required to download the content and is calculated as:

$$DR1 = log_2(1 + \Psi_{S,V}) \tag{9}$$

Whereas $\Psi_{S,V}$ denotes the Signal-to-Noise Ratio (SNR) between the content-providing server and the vehicle 'V' that is downloading the content. Similarly, DR2 represents the data rate to download contents from the fog-enabled RSU and it is computed as:

$$DR2 = log_2(1 + \Psi_{F,V}) \tag{10}$$

where $\Psi_{F,V}$ represents SNR between the fog-enabled RSU and vehicle V.

Similarly, downloading times for downloading the content size CS from the remote server (DT1) and the fog-enabled RSU (DT2) are calculated.

$$DT1 = CS/DR1 \tag{11}$$

$$DT2 = CS/DR2 \tag{12}$$

The list of symbols used in this manuscript, along with their descriptions, is shown in Table 2.

Table 2: List of symbols and notations

Symbol	Description
V	Total number of vehicles on the highway
R	Total number of Road Side Units (RSUs)
$C_{x,y}$	xth content requested by the yth vehicle
$CS_{x,y}$	Size of $C_{x,y}$
C_{req}	Total number of content requests made by vehicles
CS_{req}	Total size of the requested contents
C_{tot}	Cumulative content requests after passing X RSUs
$CCap$	Caching capacity of each fog-enabled RSU
C_i	ith content considered for caching
CS_i	Size of the ith content
C_{Alloc}	Number of contents allocated across RSUs
CS_{Alloc}	Total size of allocated contents
ϕ_k	Coverage area of the kth RSU
v_n	Velocity of the nth vehicle
t_n	Time duration nth vehicle stays connected with an RSU
t_{ave}	Average time a vehicle spends within RSU coverage
v_{ave}	Average velocity of all vehicles
DR_1	Data rate between vehicle and remote TCC server
DR_2	Data rate between vehicle and local fog-enabled RSU

(Continued)

Table 2 (continued)

Symbol	Description
$\Psi_{S,V}$	Signal-to-Noise Ratio (SNR) between server and vehicle
$\Psi_{F,V}$	SNR between fog-enabled RSU and vehicle
DT_1	Download time from TCC server
DT_2	Download time from fog-enabled RSU
Val_i	Popularity (value) score of ith content
K_i	Size (weight) of ith content in knapsack mapping
$X[i,j]$	Cell value in the knapsack table for ith content and jth cache slot
RSU_m	Maximum number of RSUs in the highway segment
Cn	Cache capacity of an RSU
M	Total number of available contents
$X_{i,j}$	Popularity of content i in RSU j
K	Size of a content cluster
N	Total number of clusters or available RSUs
V	New content request ratio in cluster allocation

4 Proposed Scheme

The cached contents on fog-enabled RSU offer a better downloading rate and allow vehicles to download their required content in a reduced duration. Caching the requested contents from the fog-enabled RSU is the demand of the vehicles. However, the limited caching capacity of the fog-enabled RSU does not allow all contents to be cached on these fog-enabled RSUs. To meet most of the vehicles' demanded contents from the nearby fog-enabled RSU demands, the most popular and demanded contents are to be cached on these RSUs. This helps in reducing the overall downloading time of all the vehicles on the highway. In this work, an efficient content caching scheme for the fog-enabled RSUs by considering the content requests from the other vehicles that are not cached on RSUs will increase at a uniform rate.

The proposed caching scheme applies the 0/1 knapsack algorithm and a clustering-based reallocation strategy, providing several key advantages over conventional methods. The knapsack algorithm enables each RSU to optimally select a subset of popular content based on size and access frequency, which is critical in environments with limited storage capacity. This ensures that highly demanded content is prioritized without exceeding cache constraints. Furthermore, the clustering-based mechanism allows RSUs to cooperatively reassign content in real time, addressing delayed or dynamic content requests from moving vehicles. This dual approach effectively minimizes content download latency, increases cache hit rates, and improves quality of service for vehicular users.

The proposed scheme efficiently allocates and determines the most popular contents to be cached on the RSUs by following the below-mentioned steps, and the same is represented in pictorial form as shown in Fig. 2.

- TCC based on the previous history of data knows the contents request trend as well as the density of the vehicles in different hours of the day.
- The contents from the total content requests are scrutinized by applying the 0/1 knapsack algorithm as discussed in Section 4.1 in such a way that the most popular contents are supposed to be cached on the first RSU, the next set of popular contents on the next RSU, and so on.

- All those vehicles whose contents are placed on their RSUs are supposed to download their contents before leaving the coverage area of that RSUs.
- A cluster of the most popular contents is repeated on the next set of RSUs by applying a clustering algorithm as discussed in Section 4.2.

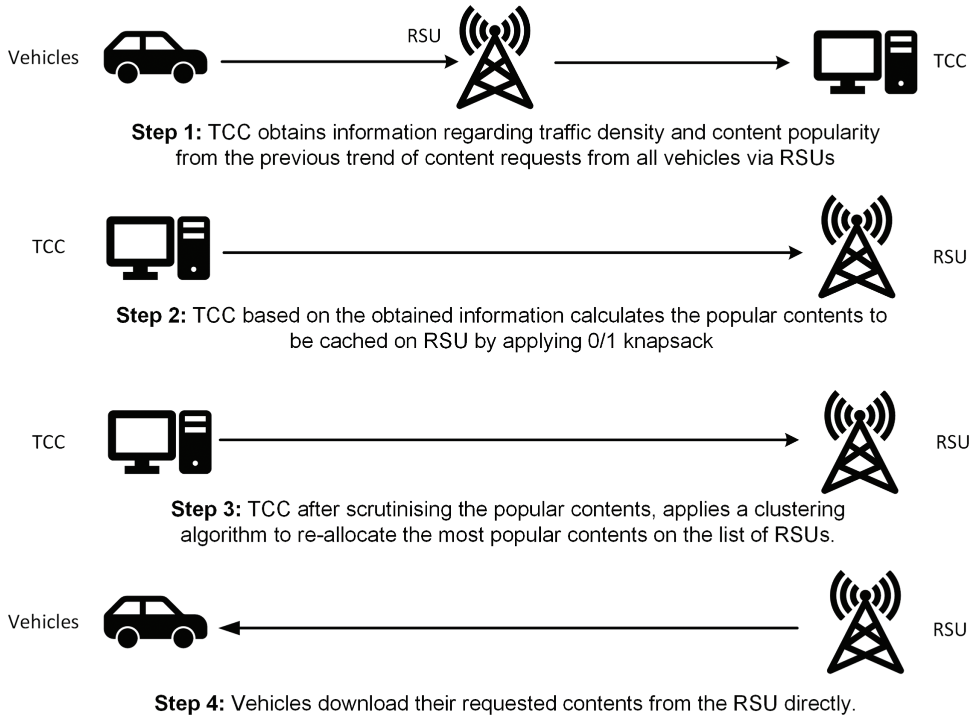

Vehicles
RSU
TCC
Step 1: TCC obtains information regarding traffic density and content popularity from the previous trend of content requests from all vehicles via RSUs
TCC
RSU
Step 2: TCC based on the obtained information calculates the popular contents to be cached on RSU by applying 0/1 knapsack
TCC
RSU
Step 3: TCC after scrutinising the popular contents, applies a clustering algorithm to re-allocate the most popular contents on the list of RSUs.
Vehicles
RSU
Step 4: Vehicles download their requested contents from the RSU directly.

Figure 2: Proposed scheme steps

4.1 0/1 Knapsack for Content Allocation

In this work, a 0/1 knapsack algorithm is applied to scrutinize the most demanding contents on the differently placed fog-enabled RSUs. The 0/1 knapsack optimally fills the sack up to its maximum capacity with the most valuable items. TCC applies the knapsack algorithm to place the most popular contents on the fog-enabled RSU within its cache capacity.

Our content caching scheme can be mapped with the 0/1 knapsack algorithm as mentioned in the Table 3.

Table 3: Knapsack mapping parameters

Contents caching problem	Knapsack problem
Available contents	Items in knapsack
Size of each content	Weight of the item
Popularity of the content	Value of the item
Caching capacity of each fog-enabled RSU	Carrying capacity of the sack

Following the placement of these items on the most valued RSU, the TCC uses the knapsack method to place the contents on the fog-enabled RSUs with the most valuable group of contents. The operation continues until all RSUs' caches are full as shown in Algorithm 1.

The algorithm populates a knapsack table with the total number of contents and RSU cache space in rows and columns, respectively. Cell value $X_{i,l}$ of the knapsack table denotes the ith content of size b and the lth cache position. The value in each table cell is filled according to one of the two conditions.

- If the size of the ith content exceeds the cache size for that column, the value in the $X_{i,l}$ cell is replaced with the value from $X_{i-1,l}$.
- If the size of the ith content exceeds the cache size for that column, the value in the $X_{i,l}$ cell is replaced with the value from $X_{i-1,l}$.
- If the size is less then we have 2 more cases given below:
    1. if cell value of $X_{i-1,l-b}$ and popularity of the content Val_i is more than the value of $X_{i-1,l}$, then $X_{i,l}$ will be filled with the accumulated values of Val_i and $X_{i-1,l-b}$.
    2. if cell value of $X_{i-1,l-b}$ and popularity of the content Val_i is less than value of $X_{i-1,l}$, then $X_{i,l}$ will be filled with $Val_i + X_{i-1,l}$

Once all the cells in the table have been filled, the process of selecting the best content begins. This selection process involves following certain parameters, starting from the last and last column and moving towards the first and first row.

1. If $X_{i,l} > X_{i-1,l}$, then the ith content is selected, and the cell value is moved to the $i-1$ row and $l-b$ column.
2. If the ith column is not chosen, the cell will be moved to the upper row with the same column $(X_{i-1,l})$ and repeat the above process.

A complete 0/1 knapsack algorithm with a content-filling table along with optimal content selection, is shown below:

Algorithm 1: Optimal content selection criteria

```
$RSU_m$ ← Maximum number of RSUs
$C_n$ ← Caching capacity
$Val_i$ ← Value of $i$th content
$M$ ← Maximum contents
$K_i$ ← Content size of $i$th content
$X[i, j]$ ← Cell value with $i$th content and $j$th cache slot
Table Filling Criteria:
while $RSUm > 0$ do
    for $k = 0$ to $Cc$ do
        Fill all cells in the first row with zeros
    end
    for $i = 1$ to $n$ do
        Fill all cells in the first row with zeros
    end
    for $i = 1$ to $M$ do
        for $l = 0$ to $Cm$ do
```

(Continued)

Algorithm 1 (continued)

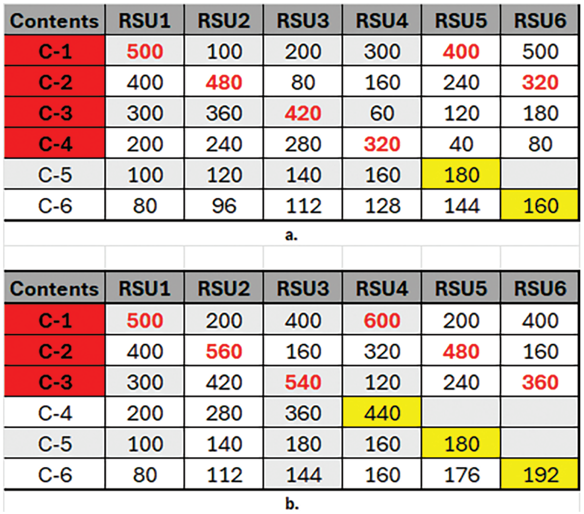
```
            if $l_i \leq H$ then
                if $Val_i + X[i-1, l-l_i] > X[i-1, l]$ then
                    $X[i, l] = Val_i + X[i-1, l-l_i]$
                end
                else
                    $X[i, l] = X[i-1, l]$
                end
            end
            else
                $X[i, l] = X[i-1, l]$
            end
        end
    end
end
Content selection from Table:
while $i > 1$ and $l > 1$ do
    // Content Selection procedure will start from the bottom right corner, i.e., last row and last column
    if $X[i, l] > X[i-1, k]$ then
        $i$th content is selected
        $i = i - 1$
        $l = l - l_i$
    end
    else
        $i = i - 1$
    end
end
```

4.2 Clustering Algorithm

In this section, the RSUs clustering algorithm is proposed to cache most of the requested contents. After successfully calculating the most demanding contents on each of the fog-enabled RSUs, the request for the allocated contents again arises in the upcoming RSUs. Suppose there is an R number of RSUs. They are filled in such a way that the most popular contents are placed at RSU1. The remaining popular contents are placed at RSU2, and the less popular contents are cached on RSU3, and so on. All the vehicles entering the highway do not originate their content requests at the start of their journey. However, it is assumed that the majority of the vehicles initiate their requests at the start of their journey, and the rest of the vehicles in the vehicular network originate their content requests at a later stage of their journey. Similarly, vehicles entering from different link roads join the vehicular network, and their content requests may be requested later. Caching the contents on the fog-enabled RSUs without considering the content requests at a later stage and by allocating contents on the RSU only once, does not satisfy the content requests of many of the vehicles. The proposed scheme is designed to cater to the majority of the content requests of the vehicles.

The proposed clustering algorithm allows TCC to reallocate the list of contents that are cached on RSUs on the upcoming RSUs by considering their traffic densities and content requests. The proposed scheme is after tentatively allocating the cache on RSUs by considering their demand. The list of those contents, that are cached on fog-enabled RSUs is considered to be downloaded by all the requested vehicles that are within their communication range. However, those vehicles that requested the same contents in the next RSUs' coverage area are not entertained, and these content requests increase as vehicles enter in the next coverage area. The proposed clustering algorithm selects the number of RSUs to cache similar contents on the upcoming list of RSUs. The proposed scheme compares the popularity of the next list of contents in the coverage of the RSU with the list of contents to be cached on the first RSU. If the popularity of the next contents is more than the first list of contents, then it allocates the contents on that RSU and moves towards the next list of popular contents to be cached on the next RSU, and compares it with the first list of contents again. If the first list of contents is more than the next popular contents, then the cluster will be finalized, and the same list of contents on all the previously cached RSUs will be repeated.

Suppose 1580 vehicles initiate their content requests at the start of their journey while entering the first RSU's communication range. However, the same set of contents is requested at a uniform rate, such as a 20% increase and 40% of the initial request at the later stage of their journey as shown in Fig. 3a and 3b, respectively. Fig. 3a shows that the requests of the list of contents C-5 are less than the content requests C-1. The proposed scheme allows contents C-1, C-2, C3, and C4 to be repeated in the next RSUs to entertain maximum requests when the number of content requests increases at the rate of 20% and C-5 and C-6 will not be cached on the next list of RSU. However, when the content requests rate increases then this allows a smaller list of contents to be cached on the next RSUs. Fig. 3b shows that contents C-1, C-2, and C-3 will be repeated on the next list of RSUs when content requests increase at a uniform rate of 40%. However, content sets C-4, C-5, and C-6 will not be cached on the upcoming RSUs.



Contents	RSU1	RSU2	RSU3	RSU4	RSU5	RSU6
C-1	500	100	200	300	400	500
C-2	400	480	80	160	240	320
C-3	300	360	420	60	120	180
C-4	200	240	280	320	40	80
C-5	100	120	140	160	180	
C-6	80	96	112	128	144	160

a.

Contents	RSU1	RSU2	RSU3	RSU4	RSU5	RSU6
C-1	500	200	400	600	200	400
C-2	400	560	160	320	480	160
C-3	300	420	540	120	240	360
C-4	200	280	360	440		
C-5	100	140	180	160	180	
C-6	80	112	144	160	176	192

b.

Figure 3: Clustering allocation example

The proposed cluster allocation algorithm is shown in Algorithm 2.

Algorithm 2: Task allocation algorithm for RSUs

```
Initialize the variables as:
    $i$ ← Current set of requested contents scrutinized by 0/1 knapsack
    $j$ ← Current RSU
    $N$ ← Maximum number of available RSUs
    $K$ ← Cluster size
```

(Continued)

Algorithm 2 (continued)

```
  $V \leftarrow$ New content requests ratio
  $X \leftarrow$ Content popularity of an RSU
for $i = 1$ to $N$ do
    Fill $N$ rows with requested calculated contents for the $j$th RSU in ascending order
    if $X_{i,j} \geq X_{1,j}$ then
        $X_{i,j}$ is allocated to the $j$th RSU
        $X_{i,j+1} = 0$
        $X_{1\text{ to }N,j+1} = V \times X_{1\text{ to }N,j+1}$
        $K = K + 1$
        else
            Cluster size $\leftarrow K$
        end
    end
end
```

5 Results and Analysis

In this section, we present a comprehensive comparative analysis of our proposed scheme with the existing schemes. For comparison, we have chosen the downloading time in downloading data contents as well as downloaded data for varying numbers of fog-enabled RSUs, vehicle requests, and caching capacity of vehicles. The results are compared with the intelligent content caching (ICC) scheme [8] and random caching scheme by creating a simulation environment in MATLAB is created. In the ICC scheme, content caching is performed using only the 0/1 knapsack algorithm, and no mechanism is implemented to reallocate or replicate the cached contents across fog nodes. The knapsack algorithm is applied once at the start, without taking vehicular density into account. Based on content popularity, the most frequently requested contents are cached on the first RSU, the next most popular contents on the second RSU, and so on in descending order of popularity. The random caching scheme does not use any optimization algorithm. Instead, requested contents are cached randomly on each RSU within its capacity limits. Importantly, there is no duplication of content across RSUs—each RSU caches a unique set of contents, meaning that the contents stored on one RSU are different from those on the others. The simulation configuration includes a variety of fog-enabled RSUs with a coverage area of 1 km with 8 traffic lanes. Vehicles are travelling at an average speed of 100 km/h and the minimum distance between two adjacent vehicles travelling in similar lanes is 50 m. Each fog-enabled RSU holds a fixed amount of cache capacity.

The contents which not stored on fog nodes are retrieved from servers located at TCC, whereas the data rate for the direct vehicle-to-TCC communications is set to 4 MBytes/s. Moreover, a data rate of 20 MBytes/s is set for vehicle-to-RSU communications [22]. Table 4 presents a list of parameters used for the simulation purpose.

Table 4: Configuration settings for simulation environment

Simulation parameter	Assigned value
Count of Road Side Units (RSUs)	2 to 12
Vehicle density per RSU coverage area	10 to 160
Effective RSU communication range	1 km
Range of content sizes (MB)	200 to 500

(Continued)

Table 4 (continued)

Simulation parameter	Assigned value
RSU cache storage capacity (GB)	25 to 50
Number of requested contents	60 to 160
Growth rate in content requests	10% increment
Average speed of vehicles (km/h)	80 to 140
Data throughput: Vehicle—TCC link (MB/s)	4
Data throughput: Vehicle—RSU link (MB/s)	20

5.1 Content Retrieval Duration

The time required to fetch the contents requested by a vehicle is modelled as the content retrieval time. If the content is stored on a nearby fog-enabled RSU, then it is downloaded at a faster rate. However, if the vehicle needs to download the requested data from a remote server, then its downloading time is increased due to reduced downloading speed.

Results shown in Figs. 4–6 are obtained by measuring the total time required to download the total number of contents requested by all vehicles and by varying the number of RSUs, vehicles and caching capacity of fog enabled RSUs, respectively.

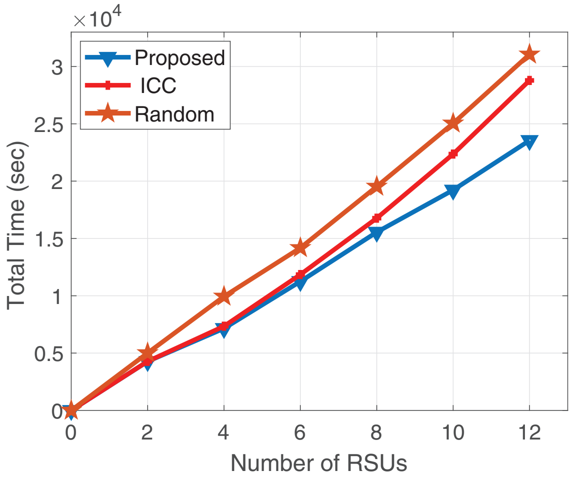


Figure 4: Accumulated duration in receiving contents for varying number of RSUs

From Fig. 4, it is evident that the total time required in downloading all the contents requested by vehicles against varying numbers of fog-enabled RSUs (when the vehicles are entering in the coverage area of each RSU) increases with an average increase of 15%. It is also evident form the obtained results by increasing the number of RSUs, the total coverage area is enhanced, which leads to an increase in vehicular count in that area. Also, it is depicted that the more vehicles request for the contents, the more time it requires to download their contents. The results show that the time calculated to download their required contents in the proposed scheme is significantly less than the other two schemes when the number of RSUs is 12. However, when the number of RSUs is 12, then the time in downloading the contents in the proposed scheme as well as in TCC is almost the same because TCC also allocates the most demanding contents cache on the fog-enabled RSUs. Moreover, when the number of RSUs increases, the time to download the requested contents in TCC increases because it does not reuse the already cached contents on the RSUs, and consequently, less number of contents are retrieved by the vehicles from RSUs, and more contents are downloaded from the servers, resulting in more downloading time. The time to download the requested contents in a random cache is much longer than the other two schemes because it does not cache contents on RSUs intelligently. This is because most of the vehicles download their required content from their attached RSUs, and a smaller number of vehicles download their content from remote servers with a lower data rate.

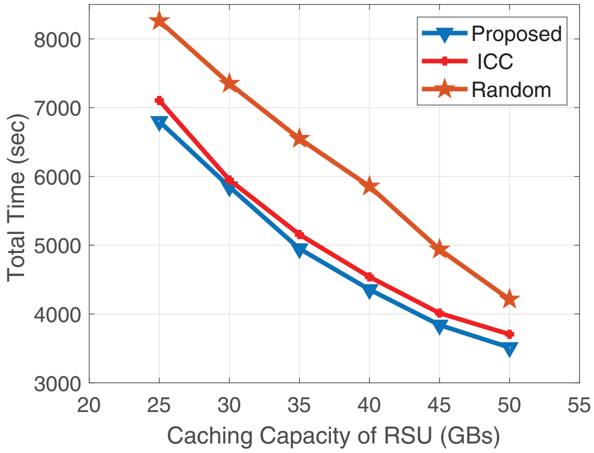


Figure 5: Accumulated duration in receiving contents for varying caching capacity of fog nodes

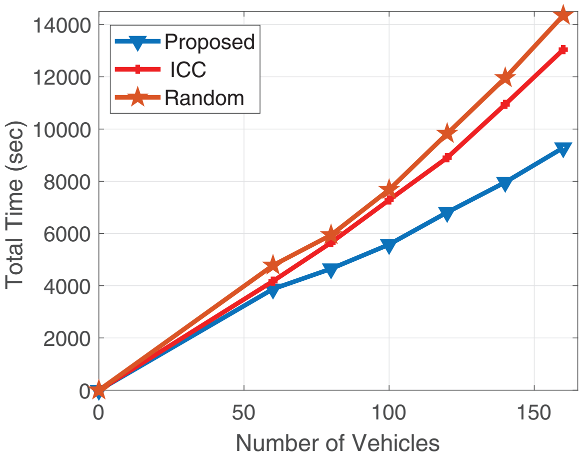


Figure 6: Accumulated duration in receiving contents for varying number of vehicles

Fig. 5 depicts that the time in downloading the requested contents in the proposed scheme is less than the other counterparts for varying cache limits of the fog-enabled RSUs. It is also evident from Fig. 5 that with the increase in the caching capacity of the fog node, the time to download the requested contents by all vehicles decreases because the increased cache capacity of the fog node allows more contents to be cached on the fog-enabled RSU, and consequently more vehicles will be entertained by downloading their requested contents from the nearby RSU. The results show that the content downloading duration of all vehicles in the proposed scheme decreases from 6750 to 3490 s when the cache capacity of the fog node increases from 25 to 50 GB, respectively. However, the contents downloading duration in ICC and Random allocation decreases from 7100 to 3765 s and 8280 to 4190 s for the same increase in cache limit, respectively.

Results in Fig. 6 depict the content downloading time for varying requests of vehicles ranging from 60 to 160. It is also illustrated in Fig. 6 that the downloading time increases with the increase in vehicles' requests. However, the total content downloading time for all vehicles in the proposed scheme is less than its counterparts. This is due to the reason that contents cached on the fog-enabled RSUs are efficiently cached by fulfilling the content requests of most of the vehicles by applying a knapsack with clustering algorithms. Based on the obtained results, it is evident that the proposed scheme has a noticeable increase in downloading time difference with an increase in vehicle density. The findings demonstrate that the proposed scheme enables vehicles to download their required content in 28% and 35% less time compared to TCC and random caching scheme, respectively.

5.2 Cached Contents

In this section, the comparative analysis focuses on the proposed scheme in terms of famous contents cached on the fog-enabled RSUs, compared to the other two schemes. The percentages of requested contents cached on fog nodes are calculated for varying numbers of RSUs, varying caching capacities of RSUs, and for varying numbers of contents requested vehicles as shown in Figs. 7–9, respectively.

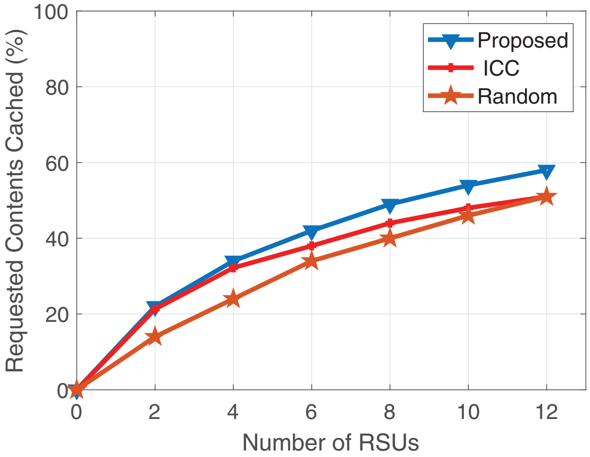


Figure 7: Requested contents cached for varying number of RSU

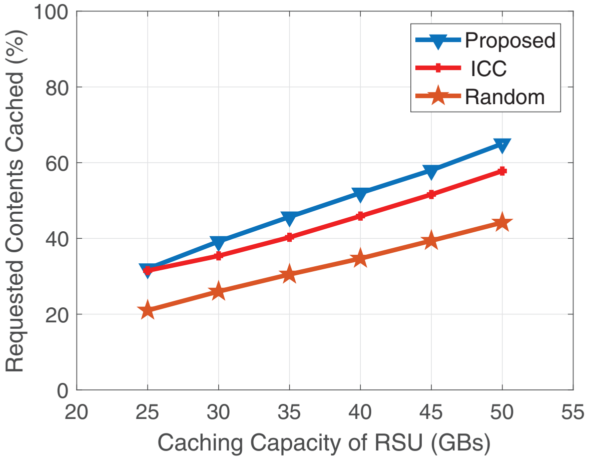


Figure 8: Requested contents cached for varying caching capacity of RSU

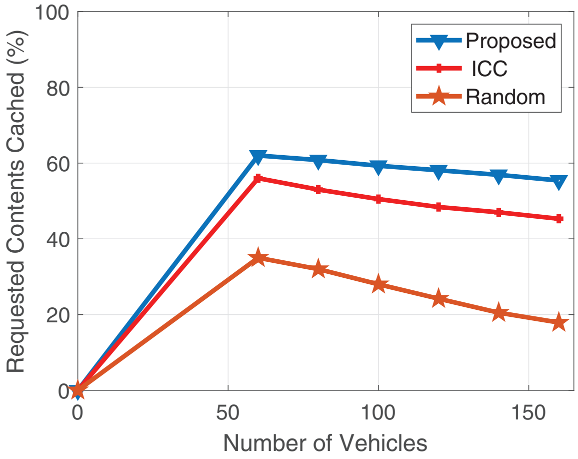


Figure 9: Requested contents cached for varying number of vehicles

Results in Fig. 7 show that the percentage of cached contents on RSUs increases with the number of RSUs. It is evident from the results that the cached contents in the proposed scheme are more than the ICC and Random cache schemes for all varying numbers of fog-enabled RSUs. The results show that the percentage of the cached content in the proposed scheme reaches 58.7% as compared to 50.7% for the other two schemes when the number of fog-enabled RSUs reaches 12.

Fig 8 represents the percentage of contents that are cached on fog-enabled RSU for varying caching capacity of fog node. The results show that with an increase in the caching capacity of the fog node, the requested contents cached on the RSUs also increase. The results show that the popular contents cached on fog-enabled RSU in the proposed scheme are 14% more than the ICC and 64% more than the random caching scheme when fog caching capacity rise to 50 GBs.

The results in Fig. 9 are calculated for varying number of vehicles' requests. The results show that the requested contents cached in the proposed scheme are more than the other two schemes. The results further show that with an increase in number of vehicles' requests, the percentage of the contents cached on fog nodes reduces for the same number of RSUs with fixed caching capacity. The results show that contents cached in the proposed scheme are up to 56% more than the ICC and up to 217% more than the random cache scheme.

5.3 Downloading Data

This section illustrates the accumulated download data from all vehicles that download their requested contents from their nearby fog-enabled RSUs only. Here, we investigate the performance capability of the proposed technique with TCC and random cache scheme by varying the number of RSUs, their caching capacity, and vehicle density as depicted in Figs. 10–12, respectively.

The obtained results in Fig. 10 illustrate that the total downloaded data of the content requesting nodes from the fog nodes for a varying number of RSUs. It is clearly evident from the obtained results that the total amount of data against the downloaded contents from fog nodes in the proposed scheme is the highest among all, and the amount of downloaded data in the random caching scheme is the least among all. The results further show that with the increase in the number of RSUs, the total data downloaded from the fog nodes also increased. This is due to the large amount of content allocated on the fog-enabled RSUs, and consequently, more data is downloaded from the fog nodes. The results further show that the amount of data downloaded from fog nodes in the proposed scheme and the TCC is almost the same as both schemes allocate popular contents on the initial RSUs. However, the popularity of the caching contents decreases with the increase in the number of RSUs because it does not consider the reallocation of contents, and the ratio of downloaded data reduces with the increase in the number of RSUs and approaches to the same amount of data as downloaded by the random caching scheme. However, in the proposed scheme, the most demanding contents are cached again to increase the content accessibility from the nearby fog node by introducing a clustering algorithm.

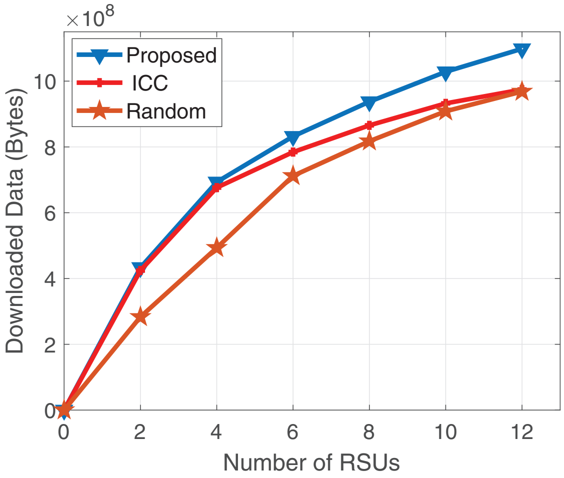


Figure 10: Accumulated contents data received against varying number of RSUs

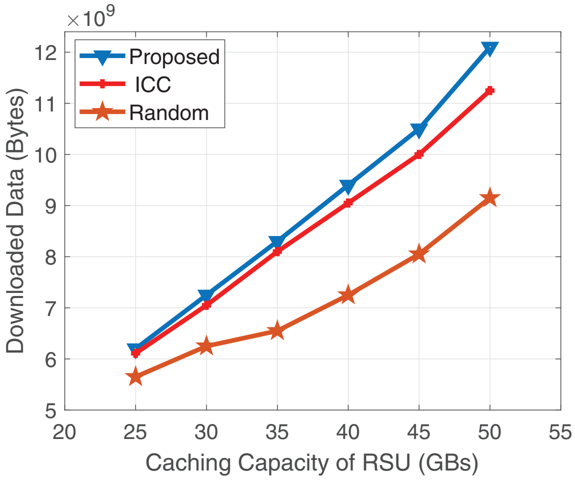


Figure 11: Accumulated contents data received against varying caching capacity of fog nodes

Results shown in Fig. 11 verify that the amount of data downloaded in the proposed scheme is more than the other two schemes for varying content caching capacity of fog-enabled RSUs. It is further demonstrated that by increasing the content caching capacity of the fog node, the downloaded data amount also increases because an increase in caching capacity allows more content to be cached on fog-enabled RSUs. The results show that the amount of data downloaded in the proposed scheme extends up to 12.05 GB of data as compared to 11.23 and 9.18 GB in TCC and random caching algorithms, respectively.

The obtained results in Fig. 12 verify that the amount of data downloaded from the fog node for varying numbers of requested vehicles in the proposed scheme is more than its counterparts. It is also demonstrated that the amount of data increases with the increase in the number of content requests. It is evident from the results that the amount of data downloaded in the proposed scheme is 33% more than the TCC and 41% more than the random caching scheme.

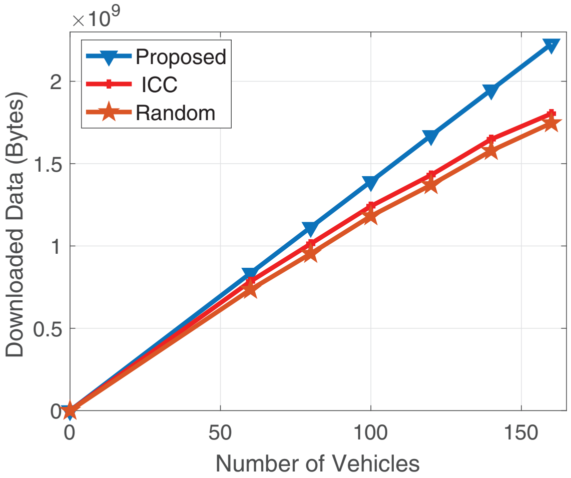


Figure 12: Accumulated contents data received against varying number of vehicles

Increased traffic density allows diverse content requests and an increased number of popular contents requests. The proposed scheme caches popular contents on the set of RSUs, and more vehicles will be entertained to fetch their contents from their attached fog-enabled RSU. The proposed caching scheme can be useful in real settings such as edge video streaming, vehicular data sharing, and IoT services where low latency matters.

6 Conclusion

In this work, an efficient content caching protocol is presented that efficiently caches the content on the fog-enabled RSUs to enhance the total downloading data with improved data rate. The proposed scheme considers the adaptive traffic density while caching the contents on the respective fog-enabled RSU first by applying the 0/1 knapsack algorithm and then applying a clustering mechanism to meet the adaptive contents requirements of vehicles. For validation, the proposed scheme was thoroughly compared with the existing random and intelligent content caching schemes. Simulation results show that the proposed scheme allows vehicles to download their requested contents in 28% and 35% reduced time compared to intelligent content caching and random cache schemes. The proposed scheme entertains up to 14% and 64% more content requesting vehicles from ICC and random caching, respectively, and overall content data downloaded from the fog-enabled RSUs is 33% more than intelligent caching and 41% more than the random caching schemes. However, this study is limited to simulations with uniform traffic density and does not fully capture real traffic variations, changing RSU coverage, or network issues like frequent handovers and link failures. The caching model also assumes smooth communication without considering security risks. In future work, we plan to test the approach under different traffic conditions, content popularity trends, and realistic mobility patterns. We also aim to apply AI-based methods to improve caching and validate the scheme in real-world settings to check its scalability and reliability.

Acknowledgement: The authors extend their appreciation to the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) for funding this work through (grant number IMSIU-DDRSP2504).

Funding Statement: This work was supported and funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) (grant number IMSIU-DDRSP2504).

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Faareh Ahmed, Babar Mansoor, Muhammad Awais Javed and Abdul Khader Jilani Saudagar; Writing—original draft, Faareh Ahmed and Babar Mansoor; Writing—review & editing, Muhammad Awais Javed and Abdul Khader Jilani Saudagar. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available within the article.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Giordani M, Polese M, Mezzavilla M, Rangan S, Zorzi M. Toward 6G networks: use cases and technologies. IEEE Commun Mag. 2020;58(3):55–61. doi:10.1109/mcom.001.1900411.
2. Saad W, Bennis M, Chen M. A vision of 6G wireless systems: applications, trends, technologies, and open research problems. IEEE Netw. 2020;34(3):134–42. doi:10.1109/mnet.001.1900287.
3. Liu T, Zhou H, Li J, Shu F, Han Z. Uplink and downlink decoupled 5G/B5G vehicular networks: a federated learning assisted client selection method. IEEE Trans Veh Technol. 2023;72(2):2280–92. doi:10.1109/tvt.2022.3207916.
4. Mohammed BA, Al-Shareeda MA, Manickam S, Al-Mekhlafi ZG, Alreshidi A, Alazmi M, et al. FC-PA: fog computing-based pseudonym authentication scheme in 5G-enabled vehicular networks. IEEE Access. 2023;11:18571–81. doi:10.1109/access.2023.3247222.
5. Sun YX. Large frequency ratio antennas based on dual-function periodic slotted patch and its quasi-complementary structure for vehicular 5G communications. IEEE Trans Veh Technol. 2023;72(7):8303–12. doi:10.1109/tvt.2023.3247022.
6. Alvi AN, Javed MA, Hasanat MHA, Khan MB, Saudagar AKJ, Alkhathami M, et al. Intelligent task offloading in fog computing based vehicular networks. Appl Sci. 2022;12(9):4521. doi:10.3390/app12094521.
7. Rahim M, Javed MA, Alvi AN, Imran M. An efficient caching policy for content retrieval in autonomous connected vehicles. Transp Res Part A Policy Pract. 2020;140(3):142–52. doi:10.1016/j.tra.2020.08.005.
8. Rahim M, Ali S, Alvi AN, Javed MA, Imran M, Azad MA, et al. An intelligent content caching protocol for connected vehicles. Emerg Telecommun Technol. 2021;32(4):1–14. doi:10.1002/ett.4231.
9. Mohammed BA, Al-Shareeda MA, Manickam S, Al-Mekhlafi ZG, Alayba AM, Sallam AA. ANAA-fog: a Novel anonymous authentication scheme for 5G-enabled vehicular fog computing. Mathematics. 2023;11(6):1446. doi:10.3390/math11061446.
10. Almazroi AA, Aldhahri EA, Al-Shareeda MA, Manickam S. ECA-VFog: an efficient certificateless authentication scheme for 5G-assisted vehicular fog computing. PLoS One. 2023;18(6):e0287291. doi:10.1371/journal.pone.0287291.
11. Wang E, Dong Q, Li Y, Zhang Y. Content placement considering the temporal and spatial attributes of content popularity in cache-enabled UAV networks. IEEE Wirel Commun Lett. 2022;11(2):250–3. doi:10.1109/lwc.2021.3124943.
12. Ling C, Zhang W, Fan Q, Feng Z, Wang J, Yadav R, et al. Cooperative service caching in vehicular edge computing networks based on transportation correlation analysis. IEEE Internet Things J. 2024;11(12):22754–67. doi:10.1109/jiot.2024.3382723.
13. Khodaparas S, Benslimane A, Yousefi S. An intelligent caching scheme considering the spatio-temporal characteristics of data in internet of vehicles. IEEE Trans Veh Technol. 2024;73(5):7019–33. doi:10.1109/tvt.2023.3337051.

14. Jin Z, Song T, Jia WK. An adaptive cooperative caching strategy for vehicular networks. IEEE Trans Mob Comput. 2024;23(10):9502–17. doi:10.1109/tmc.2024.3367543.
15. Cai X, Zheng J, Fu Y, Zhang Y, Wu W. Cooperative content caching and delivery in vehicular networks: a deep neural network approach. China Commun. 2023;20(3):43–54. doi:10.23919/jcc.2023.03.004.
16. Gu S, Sun X, Yang Z, Huang T, Xiang W, Yu K. Energy-aware coded caching strategy design with resource optimization for satellite-UAV-vehicle-integrated networks. IEEE Internet Things J. 2022;9(8):5799–811. doi:10.1109/jiot.2021.3065664.
17. Safavat S, Rawat DB. Improved multiresolution neural network for mobility-aware security and content caching for internet of vehicles. IEEE Internet Things J. 2023;10(20):17813–23. doi:10.1109/jiot.2023.3279048.
18. Xiao H, Xu C, Feng Z, Ding R, Yang S, Zhong L, et al. A transcoding-enabled 360° VR video caching and delivery framework for edge-enhanced next-generation wireless networks. IEEE J Sel Areas in Commun. 2022;40(5):1615–31. doi:10.1109/jsac.2022.3145813.
19. Bi X, Zhao L. Collaborative caching strategy for RL-based content downloading algorithm in clustered vehicular networks. IEEE Internet Things J. 2023;10(11):9585–96. doi:10.1109/jiot.2023.3235661.
20. Zhang Y, Li C, Luan TH, Fu Y, Shi W, Zhu L. A mobility-aware vehicular caching scheme in content centric networks: model and optimization. IEEE Trans Veh Technol. 2019;68(4):3100–12. doi:10.1109/tvt.2019.2899923.
21. Wang C, Chen C, Pei Q, Lv N, Song H. Popularity incentive caching for vehicular named data networking. IEEE Trans Intell Transp Syst. 2022;23(4):3640–53. doi:10.1109/tits.2020.3038924.
22. Hu Z, Zheng Z, Wang T, Song L, Li X. Roadside unit caching: auction-based storage allocation for multiple content providers. IEEE Trans Wirel Commun. 2017;16(10):6321–34. doi:10.1109/twc.2017.2721938.