People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Kasdi Merbah University of Ouargla
Faculty of New Technologies of Information and Communication
Department of Computer Science and Information Technology

Doctoral Thesis

Thesis submitted in partial fulfillment of the requirements for the degree of PhD 3rd cycle in computer science.

Option :

Networking and Telecommunications

New Caching Strategy within Mobile Edge Computing Architecture in Internet of Vehicles

Presented by

M^r Radouane Baghiani

Jury members

President:	*Mohamed el Amine Abderrahim*	Professor, University of Ouargla, Algeria
Examiner:	*Tahar Dilekh*	M.C.A,University of Batna, Algeria
Examiner:	*Bachir Said*	M.C.A,University of Ouargla, Algeria
Examiner:	*Akram Zine eddine boukhamla*	M.C.A,University of Ouargla, Algeria
Advisor:	*Lyamine Guezouli*	M.C.A, HNS-RE2SD, Batna, Algeria
Co-advisor:	*Ahmed Korichi*	Professor, University of Ouargla, Algeria

2023/2024

Acknowledgements

First and foremost, I offer my gratitude to Allah for His guidance and blessings throughout this journey. I extend my sincere appreciation to all those who have contributed to the completion of this doctoral thesis.

I am profoundly grateful to my supervisor, Dr. Lyamine Guezouli, for his invaluable guidance, unwavering support, and scholarly insights, which have been instrumental in shaping both the content and methodology of this thesis.

I also wish to express my gratitude to my co-supervisor, Professor Ahmed Korichi, whose expertise and constructive feedback have significantly contributed to the development and refinement of this work.

Special thanks go to the members of my thesis committee, [Committee Member 1], [Committee Member 2], [Committee Member 3], and [Committee Member 4] for their comments insights, constructive feedback, and encouragement throughout the process.

I am profoundly grateful to my family for their unwavering love, patience, and encouragement. Their steadfast belief in my abilities has been a constant motivation.

Heartfelt appreciation goes to colleagues for their moral support, stimulating discussions, and moments of respite during challenging times.

I extend my thanks to the University of Ouargla for providing the necessary resources and a conducive environment for research and learning.

Abstract

The Internet of Vehicles (IoV) is revolutionising transportation by enabling intelligent, connected, and autonomous vehicles. However, the massive amounts of data generated by IoV applications place significant demands on network resources. Mobile Edge Computing (MEC) architectures, which bring data processing and storage closer to vehicles, are crucial for meeting the low-latency requirements of IoV. Caching is a key enabler for IoV-MEC systems, allowing frequently accessed data to be stored locally for faster retrieval. This thesis proposes advanced, data-driven caching strategies tailored for IoV-MEC environments. The core contribution is the development of a novel caching strategy that leverages machine learning to predict future data demands based on historical access patterns and vehicle mobility. Data is proactively cached at the optimal edge locations to minimise access latency. Extensive simulations are conducted using real-world datasets to evaluate the proposed approach against baselines. Results demonstrate significant improvements in key metrics like cache hit ratio, latency, and network load compared to existing methods. For example, the new strategy achieves a 25% higher cache hit ratio and 30% lower latency than popularity-based caching. The theoretical implications include a better understanding of the interplay between caching, machine learning, and IoV performance. Practically, the findings enable more efficient, responsive, and secure IoV systems, accelerating the adoption of autonomous vehicles.

Keywords

Internet of Vehicles, Mobile Edge Computing, Caching Strategy, Reinforcement Learning, Network Optimization, Transportation Systems.

ملخص

إن إنترنت المركبات (IoV) يُحدث ثورة في النقل من خلال تمكين المركبات الذكية والمتصلة والمستقلة. ومع ذلك، فإن الكميات الهائلة من البيانات التي تولدها تطبيقات IoV تضع مطالب كبيرة على موارد الشبكة. تُعد بنى الحوسبة المتنقلة على الحافة ،(MEC) التي تقرب معالجة البيانات وتخزينها من المركبات، ضرورية لتلبية متطلبات الكمون المنخفض لـ IoV. تُعد عملية التخزين المؤقت عنصراً أساسياً لأنظمة IoV-MEC، مما يسمح بتخزين البيانات التي يتم الوصول إليها بشكل متكرر محلياً لاسترجاع أسرع. تقترح هذه الرسالة استراتيجيات تخزين مؤقت متقدمة ومبنية على البيانات، مصممة خصيصًا لبيئات IoV-MEC. المساهمة الأساسية هي تطوير استراتيجية تخزين مؤقت جديدة تستفيد من التعلم الآلي للتنبؤ باحتياجات البيانات المستقبلية بناءً على أنماط الوصول التاريخية وحركة المركبات. يتم تخزين البيانات مؤقتًا بشكل استباقي في المواقع المثلى على الحافة لتقليل كمون الوصول. تُجرى محاكيات واسعة النطاق باستخدام مجموعات بيانات من العالم الواقعي لتقييم النهج المقترح مقارنة بالخطوط الأساسية. تُظهر النتائج تحسينات كبيرة في المقاييس الرئيسية مثل نسبة نجاح التخزين المؤقت والكمون وحمل الشبكة مقارنة بالأساليب الحالية. على سبيل المثال، تحقق الاستراتيجية الجديدة نسبة نجاح تخزين مؤقت أعلى بنسبة 25% وكمون أقل بنسبة 30% من التخزين المؤقت القائم على الشعبية. تشمل الآثار النظرية فهمًا أفضل للتفاعل بين التخزين المؤقت والتعلم الآلي وأداء IoV. عمليًا، تُمكِّن النتائج من أنظمة IoV أكثر كفاءة واستجابة وأمانًا، مما يسرع من اعتماد المركبات المستقلة.

الكلمات المفتاحية

إنترنت المركبات، والحوسبة المتنقلة على الحافة، واستراتيجية التخزين المؤقت، والتعلم المعزز، وتحسين الشبكة، وأنظمة النقل.

Résumé

L'Internet des Véhicules (IoV) révolutionne le transport en permettant des véhicules intelligents, connectés et autonomes. Cependant, les quantités massives de données générées par les applications IoV imposent des exigences considérables sur les ressources du réseau. Les architectures de calcul en périphérie mobile (MEC), qui rapprochent le traitement et le stockage des données des véhicules, sont cruciales pour répondre aux exigences de faible latence de l'IoV. La mise en cache est un élément clé pour les systèmes IoV-MEC, permettant de stocker localement les données fréquemment consultées pour un accès plus rapide. Cette thèse propose des stratégies de mise en cache avancées et basées sur les données, adaptées aux environnements IoV-MEC. La contribution principale est le développement d'une nouvelle stratégie de mise en cache qui utilise l'apprentissage automatique pour prédire les demandes de données futures en fonction des schémas d'accès historiques et de la mobilité des véhicules. Les données sont mises en cache de manière proactive aux emplacements optimaux en périphérie pour minimiser la latence d'accès. Des simulations extensives sont menées en utilisant des ensembles de données réelles pour évaluer l'approche proposée par rapport aux méthodes de référence. Les résultats montrent des améliorations significatives dans des indicateurs clés tels que le taux de réussite du cache, la latence et la charge du réseau par rapport aux méthodes existantes. Par exemple, la nouvelle stratégie atteint un taux de réussite du cache 25% plus élevé et une latence 30% plus faible que la mise en cache basée sur la popularité. Les implications théoriques incluent une meilleure compréhension de l'interaction entre la mise en cache, l'apprentissage automatique et les performances de l'IoV. Pratiquement, les résultats permettent de rendre les systèmes IoV plus efficaces, réactifs et sécurisés, accélérant ainsi l'adoption des véhicules autonomes.

Mots clés

Internet des véhicules, informatique mobile en périphérie, stratégie de mise en cache, apprentissage par renforcement, optimisation des réseaux, systèmes de transport.

Contents

List of Figures

List of Tables

List of Abbreviations

IoV . . . . . Internet of Vehicles.

MEC . . . . . Mobile Edge Computing.

AI . . . . . Artificial Intelligence.

ML . . . . . Machine Learning.

TS-MMCM . Multi-Hierarchical, Mobility-Sensitive Caching Model based on Thompson Sampling.

LSTM . . . . . Long Short-Term Memory.

RL . . . . . Reinforcement Learning.

LRU . . . . . Least Recently Used.

LFU . . . . . Least Frequently Used.

MANETs . . . Mobile Ad Hoc Networks.

VANET . . . . Vehicular Ad-Hoc Network.

V2V . . . . . Vehicle-to-vehicle.

V2I . . . . . Vehicle-to-Infrastructure.

IoT . . . . . Internet of Things .

OBU . . . . . On-Board Unit.

RSU . . . . . Road-Side Unit.

BS . . . . . Base Station.

AU . . . . . Application Unit.

IEEE . . . . . Institute of Electrical and Electronics Engineers.

V2X . . . . . Vehicle-to-Everything.

Wi-Fi . . . . . Wireless Fidelity.

DSRC . . . . . Ded- icated Short-Range Communications.

IT . . . . . Information Technology.

UTMS . . . . . urban traffic management systems.

ITS . . . . . transport information systems .

4G . . . . . Fourth Generation.

5G . . . . . Fifth Generation.

LTE . . . . . Long Term Evolution.

V2P . . . . . Vehicle-to-Pedestrians.

V2G . . . . . Vehicle-to-Grids.

V2N . . . . . Vehicle-to-Networks.

V2C . . . . . Vehicle-to-Cloud.

V2D . . . . . Vehicle-to-Cloud.

EVs . . . . . Electric Vehicles.

V2H . . . . . Vehicle-to-Home.

AR . . . . . Augmented Reality.

QoS . . . . . Quality of Service.

CC . . . . . Cloud Computing .

APIs . . . . . Application Programming Interfaces.

QoE . . . . . Quality of Experience.

HTTP . . . . . Hypertext Transfer Protocol.

TCP . . . . . Transmission Control Protocol.

RAN . . . . . Radio Access Network.

IaaS . . . . . Infrastructure-as-a-Service.

IS . . . . . Infrastructure Services.

CS . . . . . Communication Services.

SR . . . . . Service Registry.

RNIS . . . . . Radio Network Information Services.

TOF . . . . . raffic Offload Function.

VM . . . . . Virtual Machine.

NFV . . . . . Network Function Virtualization.

KVM . . . . . Kernel-based Virtual Machine.

SNMP . . . . . Simple Network Management Protocol.

EPC . . . . . Evolved Packet Core.

CC . . . . . Cloud Computing.

EC . . . . . Edge Computing.

MCC . . . . . Mobile Cloud Computing.

6G . . . . . Sixth Generation.

RNN . . . . . Recurrent Neural Network.

UAV . . . . . Unmanned Aerial Vehicle.

DQN . . . . . Deep Q-Network.

MAB . . . . . Multi-Armed Bandit.

DL . . . . . Deap Learning.

CNN . . . . . Convolutional Neural Network.

GANs . . . . . Generative Adversarial Networks.

Chapter 1

General Introduction

Contents

1.1 Towards the goals

In recent years, technological advancements have significantly transformed various industries, notably the automotive sector. The emergence of the Internet of Vehicles (IoV) and Mobile Edge Computing (MEC) represents a pivotal shift towards more intelligent and interconnected transportation systems. The IoV enables seamless communication between vehicles, infrastructure, and other road users, thereby enhancing safety, efficiency, and the overall driving experience. Concurrently, MEC brings computational resources closer to the data source, reducing latency and facilitating real-time processing and decision-making. This convergence of IoV and MEC is poised to revolutionise vehicular ecosystems, fostering innovations in autonomous driving, traffic management, and smart city integration. As these technologies continue to evolve, they present unparalleled opportunities and challenges for industry stakeholders, demanding a closer examination of their current applications and future potential. These technologies (MEC and IoV) are

pivotal in the digital transformation of urban transport systems, offering significant improvements in the intelligence and automation of transportation infrastructure.

1.1.1 Internet of Vehicles

The IoV amalgamates advanced communication and sensor technologies within vehicles, facilitating real-time data collection and communication. This integration not only enables autonomous navigation but also supports vehicular communication with road infrastructures and other vehicles, enhancing traffic safety and efficiency. The IoV underpins applications such as real-time traffic management, early warning systems, and data-assisted navigation, thereby contributing to more fluid and safer urban mobility.

1.1.2 Mobile Edge Computing

MEC processes data near its source, minimising latency and enhancing the responsiveness of applications. In the context of IoV, MEC efficiently processes vast volumes of data generated by vehicles, which is crucial for applications like autonomous driving and real-time road condition monitoring. The synergistic interaction between IoV and MEC lays the groundwork for advancing intelligent transport technologies, optimising communication and data management, and enhancing safety and energy efficiency in urban transport systems.

1.1.3 Importance of Caching in IoV

Data caching is crucial in the IoV to optimise system performance and efficiency, particularly for real-time applications. Pre-emptive caching at strategic vehicular network points, including vehicles, base stations, and roadside units, is instrumental in minimising latency for accessing critical data such as traffic conditions and safety alerts. The benefits of caching include reduced latency, bandwidth optimisation, and enhanced system efficiency, collectively improving the performance and safety of intelligent transport systems.

1.1.4 Research Gaps

This study identifies several critical gaps in the field of caching for vehicular and mobile edge network systems. Addressing these gaps is essential for understanding the limitations of current approaches and guiding future research aimed at enhancing the efficiency and responsiveness of caching systems. The main gaps identified are:

1. **Lack of Consideration of Combined Regional Preferences:** Most previous studies have focused on caching in either vehicles or peripheral devices independently. However, these approaches overlook the importance of combined caches and regional preferences. This limitation hinders the effectiveness of the proposed caching strategies, as it does not allow for full optimisation based on local specificities and user behaviours on a regional scale.

2. **Insufficient Study of Vehicle Mobility Impact:** Existing research has not sufficiently examined the impact of vehicle mobility on caching decisions in autonomous driving scenarios. The absence of this consideration limits the optimisation of caching systems in dynamic environments such as autonomous vehicle networks, where fluctuating mobility and connectivity play a critical role in system efficiency.

3. **Content Popularity Prediction Based on Static Assumptions:** Many studies assume that content popularity is predefined or use only the frequency of requests to assess popularity. However, this approach does not consider the element of freshness and the temporal evolution of content requests. A more dynamic, real-time prediction of content popularity is required for more effective cache management.

4. **Computational Resource Needs and Confidentiality Concerns:** The integration of artificial intelligence (AI) and machine learning (ML) into caching systems offers significant benefits, particularly in terms of demand prediction and dynamic adaptation of caching strategies. However, these

technologies also pose significant challenges. The computational resource requirements for processing large amounts of data and executing complex algorithms can be prohibitive. Additionally, data privacy concerns must be addressed to ensure the safe and ethical adoption of these technologies.

These gaps highlight the need to develop more integrated, adaptive, and secure caching approaches that can leverage advanced technologies while addressing the specific challenges of vehicular network environments. Future research should focus on creating more robust and scalable models that account for both local and global network dynamics to improve overall performance and user experience.

1.1.5 Research Objective

The objective of this research is to develop optimised caching strategies for the Internet of Vehicles within Mobile Edge Computing architectures. By integrating advanced caching and artificial intelligence techniques, the goal is to enhance the performance and efficiency of vehicular networks. This approach aims to optimise caching decisions based on content popularity and vehicle mobility characteristics.

1.2 Research Questions

1.2.1 What is the Role of Caching in IoV within MEC Architectures?

Caching plays a crucial role in the integration of the IoV with MEC architectures by enhancing both performance and security. By storing data locally at network edges, caching reduces latency, enabling vehicles to access essential information such as real-time traffic conditions and navigation updates almost instantaneously. This proximity of data storage also improves the efficiency of data processing, as it distributes workloads across the network and reduces the reliance on central data centres, thereby optimising bandwidth and minimising network congestion. Furthermore, localised data storage enhances security by limiting the amount of data transmitted over longer distances, reducing the risk of data breaches. Safety is

significantly bolstered through caching; vehicles can make real-time safety decisions based on locally available data, such as rerouting to avoid sudden road closures or adjusting speed according to traffic flow changes. Predictive safety measures benefit as well, with systems able to analyse cached data to forecast potential risks and proactively adjust vehicle behaviours. Overall, caching within IoV and MEC architectures ensures a responsive, secure, and efficient operational environment, vital for supporting the dynamic needs of modern autonomous and connected vehicle technologies.

1.2.2 How Can a New Caching Strategy Improve IoV Performance?

The adoption of new caching strategies in the IoV, particularly within MEC architectures, is essential for improving the operational efficiency and responsiveness of systems. These optimised strategies, including predictive caching based on machine learning, geographic caching, and collaborative caching, enable more efficient management of the voluminous and dynamic data generated by vehicles. By anticipating data demands and adapting storage to mobility patterns, these approaches reduce network congestion and improve the response speed of IoV applications, which are essential for real-time navigation and driver assistance systems. The integration of these innovative strategies not only enhances the reliability and safety of intelligent transport systems but also contributes to an improved user experience and more sustainable management of network resources. These advances are crucial to maintaining system fluidity and responsiveness in highly dynamic environments, marking a significant step forward in the evolution of IoV.

1.2.3 What Are the Implications of Implementing This Strategy?

The implementation of innovative caching strategies in the Internet of Vehicles has significant implications for the operational efficiency and reliability of intelligent transport systems. These strategies facilitate the efficient management of data traffic, particularly during periods of high demand, by distributing the data load

across multiple cache locations. This strategic distribution significantly reduces network congestion, ensuring a constant flow of data and maintaining a high quality of service across the network.

Adopting such strategies not only improves operational efficiency but also enhances security and overall user satisfaction within the IoV. The strategic integration of caching is crucial to realise the full potential of IoV technologies in modern transport networks, enabling a seamless and efficient infrastructure.

1.3 Significance of the Study

1.3.1 Contribution to the Field

This study makes significant contributions to the field of caching for vehicular networks and mobile edge systems, particularly in the contexts of the Internet of Vehicles and mobile edge computing architectures. The main contributions of this research are as follows:

1. **Innovative Integration of AI and ML in Caching Strategies:** This study proposes a multi-hierarchical, mobility-sensitive caching model based on Thompson sampling (TS-MMCM), integrating artificial intelligence and machine learning. This model enables accurate prediction of content requests, improving cache hit rates and reducing latency, which is crucial for real-time applications and dynamic networks.

2. **Vehicle Clustering for Optimised Communication:** The research introduces an advanced method for clustering vehicles based on their mobility characteristics (speed and position), using the K-means algorithm and a selection criterion based on connectivity duration (CL). This approach improves the stability of communications between vehicles, optimising caching decisions and increasing the reliability of vehicle networks.

3. **Predictive Modelling of Content Popularity:** Using Long Short-Term Memory Neural Networks (LSTM), the study proposes a method for predicting content popularity based on the analysis of historical demand trends. This predictive modelling enables future requests to be anticipated, ensuring that the most popular content is proactively cached, thereby optimising the use of cache resources and improving the user experience.

4. **Multi-Agent Approach for Optimising Caching Decisions:** The research implements a multi-agent reinforcement learning (RL) based decision approach, where each entity (edge server, base station, cluster head, local user) acts as an agent optimising its caching decisions. This approach enables real-time adaptation to changing network conditions, maximising cache hit rates and reducing bandwidth costs.

5. **Exhaustive Validation and Simulation:** The theoretical contributions of the study are validated through exhaustive simulations, showing significant performance improvements over existing caching algorithms such as LRU, LFU, Fuzzy Logic, and ICSAD. Simulation results demonstrate the superiority of the TS-MMCM model in terms of cache hit rates and latency reduction, proving the effectiveness of integrating AI and ML into caching strategies.

These contributions offer a new perspective on the optimisation of caching strategies in the Internet of Vehicles (IoV) and mobile edge computing (MEC) systems. The results of this study could inspire future research and development in the field, fostering the emergence of more efficient and robust vehicular networks and mobile edge systems.

1.3.2 Practical Implications

The findings of this research have direct implications for improving intelligent transport systems. By optimising caching management in the IoV, this study contributes to better network congestion management, reduced communication delays, and increased data security, resulting in more efficient and secure transport systems.

These practical implications encourage the adoption of autonomous vehicles and improve coordination between vehicles and road infrastructure.

1.3.3 Theoretical Implications

This research enriches the academic understanding of caching systems in complex environments such as the IoV. By analysing the impact of caching strategies on the performance of IoV and MEC networks, it sheds light on the underlying mechanisms that facilitate or hinder the efficiency of embedded systems in vehicles and edge infrastructures. This contributes to a better conceptualisation of the challenges related to scalability, security, and data management in increasingly autonomous and connected transport networks.

1.4 Structure of the Thesis

This thesis explores caching strategies within Mobile edge computing architectures for the Internet of Vehicles. Its primary aim is to enhance the performance and efficiency of vehicular networks through advanced caching techniques. The organisation of this thesis is detailed below:

- **Chapter 2: Comprehensive Analysis of IoV and MEC Technologies**

  This chapter presents a comprehensive literature review, detailing the evolution, architecture, and challenges associated with IoV and MEC technologies. It also examines various caching approaches employed in IoV networks.

- **Chapter 3: Caching Management and Reinforcement Learning in IoV**

  Focusing on caching management techniques, this chapter introduces the application of reinforcement learning to optimise caching performance in IoV. It discusses both the theoretical underpinnings and practical applications of these techniques.

- **Chapter 4: Advanced Data-Driven Caching Strategy**

  Here, the development and evaluation of an innovative caching strategy are discussed. The chapter outlines the research methodology, describes the experiments conducted, and presents the results obtained, comparing this new strategy with existing ones.

- **Chapter 5: Reflections and Future Directions**

  This chapter summarises key research findings, discusses their practical and theoretical implications, and makes recommendations for implementing caching strategies in IoV environments. It also explores potential avenues for future research.

- **Chapter 6: Conclusion**

  The conclusion reiterates the contributions of the thesis, emphasising its added value to the scientific community and practical applications, and provides final thoughts.

Chapter 2

Comprehensive Analysis of IoV and MEC Technologies

Contents

2.1 Introduction

The world of transportation is undergoing a revolution, driven by the integration of advanced technologies such as the Internet of Vehicles (IoV) and Mobile Edge Computing (MEC). These innovations are transforming the way vehicles interact not only with each other but also with surrounding infrastructure and road users. Chapter 2 of this thesis delves deeply into these two technological pillars, highlighting their synergy and impact on intelligent transport systems.

The Internet of Vehicles (IoV) is distinguished by its ability to provide ubiquitous connectivity and real-time bidirectional communication between vehicles (V2V), between vehicles and infrastructure (V2I), and between vehicles and pedestrians (V2P). This interconnectivity facilitates a range of applications from real-

time traffic management to autonomous driving, significantly enhancing road safety and travel efficiency.

Simultaneously, Mobile Edge Computing (MEC) plays a crucial role by bringing processing and storage resources closer to vehicles, thus reducing latency and enabling real-time responses to data generated by vehicles. MEC supports the low latency and high bandwidth requirements of IoV, offering a flexible and scalable infrastructure for local data processing.

This chapter begins with an overview of the fundamentals and key concepts of IoV and MEC. It then examines their practical applications, respective architectures, and the technical challenges they present. Special attention is given to the complementary relationship between IoV and MEC, highlighting how this technological combination revolutionises traffic management, vehicle safety, and mobility services.

In summary, this detailed analysis serves as a foundation for understanding the innovations and future trends in the field of intelligent transportation, setting the stage for subsequent chapters that will address advanced caching strategies adapted to these dynamic environments.

2.2 IoV and MEC Overview

IoV and MEC are essential technological concepts that significantly advance the fields of modern communications and automation. These innovations facilitate real-time data processing and enhanced connectivity, which are crucial for the development of smart transportation systems and automated services. This comprehensive literature review delves deeply into both areas, exploring their definitions, key concepts, interrelations, and the critical roles they play in shaping the future of digital interaction and infrastructural efficiency.

2.2.1 Background of IoV

The transformation of communication systems in the vehicular transport sector began with the development of MANET, initially designed for military and emer-

gency response applications. These networks, characterised by their autonomy and flexibility to establish dynamic network configurations without a pre-established infrastructure, laid the foundations for significant advances in the transport sector [1]. Over time, this technology has evolved into vehicular ad hoc networks, facilitating direct inter-vehicle communications (V2V), which have improved road safety through better risk anticipation and optimised reactivity in critical situations. VANETs then enhanced these capabilities by integrating V2V and V2I communications, crucial for more dynamic traffic management and a significant reduction in congestion and harmful emissions[2]. The adoption of the IoV, marking the integration of IoT technologies within transport systems, has transformed vehicles into entities equipped with intelligent sensors, advanced communication technologies, and robust analytical capabilities for real-time decisions[3]. These advances are not just technological successes; they have profoundly influenced urban mobility, increasing economic efficiency and improving quality of life. More broadly, they pave the way for the future integration of fully autonomous and interconnected vehicles, illustrating a remarkable fusion of technological innovation and strategic vision to redefine our conceptions of mobility and safety in modern urban environments.

2.2.2 Definition of Internet of Vehicles

The Internet of Vehicles (IoV) is a sophisticated subset of the Internet of Things (IoT) aimed at enhancing connectivity and interactions between vehicles and their surroundings, such as transport infrastructures, safety devices, other vehicles, and pedestrians. Utilising advanced communication, control, and information systems, the IoV fosters the development of an intelligent, automated transportation network[4].

Specifically, the IoV employs embedded sensors and devices within vehicles to collect and transmit real-time data, covering aspects such as speed, position, traffic and weather conditions, and driver behaviour. For instance, V2V communication allows cars to exchange details about their speed and position, aiding in collision prevention and traffic flow improvement. Furthermore, the IoV extends to V2I

communications, enabling interactions with stationary components such as traffic lights, smart road signs, and electric vehicle charging stations[5]. These communications are critical for regulating traffic, reducing congestion, and enhancing the energy efficiency of vehicles.

The design of a connected vehicle consists of three fundamental components: the On-Board Unit (OBU), a Road-Side Unit (RSU) or Base Station (BS), and an Application Unit (AU), as shown in Figure 2.1. The IoV supports a broad range of uses, including safety-related features such as lane-changing assistance, as well as non-safety applications like infotainment services.

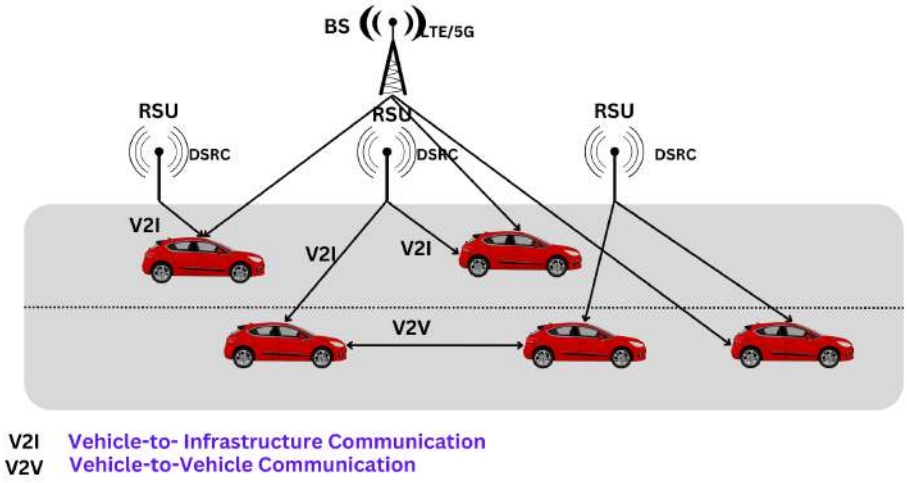


Figure 2.1: Fundamental structure of a vehicular network.

2.2.3 Key Concepts of IoV

2.2.3.1 Ubiquitous Connectivity

The Internet of Vehicles leverages pervasive connectivity to facilitate real-time, continuous communication among vehicles, road infrastructure, and other devices. This widespread connectivity forms a dynamic, interconnected network that enhances information exchange and action coordination. These capabilities significantly improve traffic management, boost safety, and provide a smoother driving experience for users[6].

2.2.3.2 V2V Communication

A key feature of the Internet of Vehicles is V2V communication, which allows vehicles to directly share vital information without relying on central infrastructure[7]. This exchange includes data on speed, position, planned manoeuvres, and hazard warnings. V2V communication helps prevent collisions, enhances traffic coordination, and boosts overall road safety.

2.2.3.3 V2I Communication

In addition to vehicle-to-vehicle communication, the IoV supports communication between vehicles and intelligent road infrastructure. Equipped with sensors and communication devices, infrastructure elements transmit vital information to vehicles, including road conditions, traffic lights, signs, and work zones. This V2I communication enhances traffic management, streamlines navigation, and reduces travel times for drivers.

2.2.3.4 Security and Confidentiality

Data security and confidentiality are paramount in the IoV. As vehicles connect to the internet and share sensitive information, implementing strong security measures is crucial to protect against cyber-attacks and unauthorised access. Essential technologies for safeguarding data include encryption, authentication of users and devices, and robust defences such as firewalls and intrusion detection systems. These measures are critical to ensuring the security and privacy of IoV data.

2.2.3.5 Interoperability and Standardisation

To facilitate the smooth operation of the Internet of Vehicles (IoV) on a large scale, establishing interoperable communication standards and protocols is essential. This ensures effective collaboration among various IoV stakeholders, including vehicle manufacturers, technology suppliers, and government bodies, and guarantees system and device compatibility. Initiatives such as the IoV Consortium and Institute

of Electrical and Electronics Engineers (IEEE) standards are crucial in promoting interoperability and fostering uniform adoption of IoV technologies globally.

The table 2.1 presents a clear and well-structured overview of each IoV concept, encompassing a succinct description, its implications for the system and users, and practical examples.

Table 2.1: Key Concepts of the IoV

Concept	**Description**	**Implications**	**Examples**
Ubiquitous Connectivity	Facilitates real-time, continuous communication among vehicles, infrastructure, and devices.	Improves traffic management, boosts safety, and enhances the driving experience.	Real-time communication between vehicles and road infrastructure.
V2V Communication	Allows direct sharing of vital information between vehicles without central infrastructure.	Prevents collisions, enhances traffic coordination, and boosts road safety.	Sharing data on speed, position, and planned manoeuvres between vehicles.
V2I Communication	Supports communication between vehicles and intelligent road infrastructure.	Enhances traffic management, streamlines navigation, and reduces travel times.	Transmission of information about road conditions, traffic lights, and signs.
Security and Confidentiality	Focuses on protecting sensitive information shared within the network.	Ensures the security and privacy of IoV data, protecting against cyber-attacks.	Use of encryption, user and device authentication, and robust intrusion detection systems.
Interoperability and Standardisation	Establishes communication standards and protocols for effective collaboration.	Guarantees system and device compatibility, promoting global adoption of IoV technologies.	IoV Consortium and IEEE standards for interoperability.

2.2.4 IoV Applications

2.2.4.1 Traffic Management and Assisted Navigation

The IoV delivers innovative traffic management and navigation assistance. By gathering real-time data on speed, traffic density, and weather conditions, IoV systems provide drivers with alternative routes and precise navigation to avoid congestion, shorten travel times, and enhance traffic efficiency[8]. Additionally, IoV-based traffic management algorithms optimise traffic light timings and signage to reduce delays and improve traffic flow.

2.2.4.2 Automated Driving and Autonomous Vehicles

The Internet of Vehicles is crucial in advancing automated driving and autonomous vehicles. It enables bidirectional communication between vehicles, infrastructure, and other road elements, enhancing the situational awareness of autonomous vehicles. This improved perception allows for more informed decision-making and proactive responses to dynamic road conditions[9]. Additionally, autonomous vehicles utilise IoV data to optimise routes, avoid obstacles, and interact safely with other road users, contributing to fewer road accidents and enhanced road safety.

2.2.4.3 Mobility Services and Vehicle Sharing

The IoV is ushering in a new era of mobility services and vehicle sharing[3]. IoV-enhanced car-sharing and ride-sharing platforms allow users to conveniently access a variety of shared vehicles, including traditional cars, electric vehicles, and self-service bicycles . Intelligent matching algorithms, powered by IoV data, optimise the utilisation of these shared vehicles. This not only lowers costs for users but also promotes a more efficient use of transportation resources. The Internet of Vehicles (IoV) significantly enhances road safety and accident prevention. By facilitating real-time communication between vehicles, infrastructure, and other road elements, the IoV detects hazardous situations and issues warnings to drivers, helping to prevent collisions and incidents. Additionally, in the event of an accident, the IoV swiftly coordinates emergency responses by transmitting crucial information, such

as the vehicle's location and the severity of injuries, to emergency services. This capability not only saves lives but also reduces response times during emergencies.

2.2.5 IoV Architecture Components

The architecture of IoV comprises several key elements that work together to enable its connectivity and advanced functionalities[10]. The primary components of the IoV architecture include:

2.2.5.1 IoV Architecture Communication Elements

IoV communication elements are crucial for facilitating information exchange among various stakeholders within the IoV system. They are integral in collecting, processing, and transmitting relevant data, ensuring the IoV network operates efficiently. Here's a detailed look at the main communication elements:

2.2.5.1.1 Connected Vehicles

Connected vehicles are central to the Internet of Vehicles (IoV), featuring advanced technologies that enable them to gather, process, and communicate essential data regarding their environment, status, and driving behaviour[11]. Here's a detailed analysis of the components and functionalities of connected vehicles:

- Onboard Sensors: These include cameras, radar, and lidar systems that monitor external conditions such as road obstacles, traffic, and weather, as well as internal parameters like speed and engine performance.
- Communication Modules: Vehicles are equipped with modules that support Vehicle-to-Everything (V2X) communication. This allows them to interact with other vehicles, road infrastructure, and pedestrian devices, facilitating a continuous exchange of information.
- Data Processing Units: These onboard computers analyse the data collected by sensors to make real-time decisions. They play a crucial role in functions such as automated braking, lane-keeping, and adaptive cruise control.

- User Interface: This includes display systems and control panels that provide drivers with information and alerts derived from IoV communications, such as traffic updates, navigation assistance, and safety warnings.
- Connectivity Solutions: Technologies like cellular networks, Wireless Fidelity (Wi-Fi), and Dedicated Short-Range Communications (DSRC) enable seamless and continuous connectivity, ensuring that vehicles can communicate effectively, even at high speeds or in dense urban areas.
- Software Systems : Software in connected vehicles includes operating systems and applications that manage hardware functions, integrate with mobile devices, and support updates and patches to improve vehicle functionality over time.

  Together, these components enable connected vehicles to function as intelligent agents within the IoV, enhancing road safety, optimising traffic flow, and improving the overall driving experience.

2.2.5.1.2 Intelligent Road Infrastructure

Intelligent road infrastructures are equipped with sensors, cameras, and integrated communication devices to collect data on road conditions, traffic, and incidents. These infrastructures include a variety of equipment such as intelligent traffic signs, adaptive traffic lights, collision detection systems, vehicle detectors, and electric vehicle charging stations. The data collected by these infrastructures is used to improve traffic management, road safety, and urban planning.

2.2.5.1.3 Servers and Data

Servers and data play a fundamental role in the Information Technology (IT) infrastructure that supports the IoV system. They serve as the central hub where data generated by connected vehicles and intelligent road infrastructures converges and is processed[12]. These servers centralise, store, analyse, and manage data, providing crucial information to improve traffic management, road safety, and mobility services. Here is a detailed analysis of their functions and significance:

- Data storage: Servers are equipped with massive storage capacity to hold the vast amounts of data generated by the IoV. This data can include information on road conditions, traffic, incidents, driving behaviour, software updates, vehicle diagnostics, and much more. Data is stored securely and is readily accessible, allowing for quick and efficient access when needed.

- Data Analysis: Stored data is analysed using advanced algorithms and data analysis tools to extract useful and relevant information. This analysis is used to detect traffic trends, identify behaviour patterns, predict potential incidents, evaluate the effectiveness of transport policies, and generate recommendations for improving urban mobility. The results of the analysis are utilised to make informed decisions and implement effective strategies to optimise IoV system performance.

- Data management: Servers ensure the effective management of data throughout its lifecycle, from initial collection to storage, analysis, and archiving. This includes data normalisation, version management, regular backup, protection against data loss, regulatory compliance, and data confidentiality. Effective data management ensures data integrity, reliability, and availability for IoV system users.

- Real-time information services: Servers provide real-time information services on traffic, road conditions, incidents, and navigation recommendations to IoV system users. These services enable drivers to make informed decisions based on current road conditions, optimise their routes to avoid traffic jams, and react quickly to emergency situations. Real-time information helps to improve driving safety, efficiency, and comfort.

- "Integration with other systems: Servers are often integrated with other transport management systems, such as urban traffic management systems (UTMS), intelligent transport systems (ITS), fleet management systems, and mobility control systems . This integration enables effective collaboration

between the various players in the transport system and overall optimisation of transport network performance[13].

- Security and confidentiality: Data security and confidentiality are major concerns in the field of IoV servers[14]. Since these infrastructures store and process sensitive data on road conditions, traffic, and driving behaviour, it is essential to implement robust security measures to protect this information against cyber-attacks and malicious intrusions. Servers use advanced security protocols, such as data encryption, firewalls, intrusion detection systems, and access controls, to ensure the integrity, confidentiality, and availability of stored data.

2.2.5.2 Types of Communications

The types of communications in the Internet of Vehicles (IoV) determine the nature and scope of information exchanges between the various entities within the system. These communications play an essential role in the rapid and reliable transmission of data, ensuring the smooth functioning of the IoV system[15]. Here is an expansion on the main types of communications, classified according to their scope:

2.2.5.2.1 Short Range Communications

Short range communications in the Internet of Vehicles (IoV) facilitate immediate and localised interactions between vehicles and their immediate environment. These interactions are crucial for ensuring real-time responses and safety measures that are essential in dense traffic scenarios and complex urban landscapes[16].

- **Vehicle-to-vehicle (V2V) Communication:** V2V communication involves the direct exchange of information between vehicles. This type of communication allows vehicles to share data on their speed, position, and direction. The primary objective is to enhance safety by enabling collision avoidance, cooperative driving, and efficient traffic management. The technologies used include Dedicated Short Range Communications (DSRC) and Cellular Vehicle-to-Everything (C-V2X)[17].

- **Vehicle-to-pedestrian (V2P) Communication:** V2P communication involves direct communication between vehicles and pedestrians. It aims to improve pedestrian safety by alerting them to the presence of nearby vehicles and vice versa. This type of communication helps to reduce the risk of accidents involving pedestrians, particularly in urban environments[18]. The technologies used include smartphone applications, DSRC, and C-V2X.

- **Vehicle-to-Device (V2D) Communication:** V2D communication involves interaction between vehicles and various devices, such as smartphones, tablets, and other personal devices carried by passengers[19]. The aim is to enhance the user experience by providing personalised services, infotainment, and connectivity. Technologies used include Bluetooth, Wi-Fi, and Near Field Communication (NFC).

2.2.5.2.2 Medium Range Communications

Medium range communications enhance the capabilities of the Internet of Vehicles (IoV) by enabling vehicles to interact with fixed infrastructure components, such as traffic signals, road signs, and management systems.

- **Vehicle-to-Infrastructure (V2I) communication:** V2I communication involves the exchange of data between vehicles and road infrastructure, including traffic lights, road signs, and traffic management systems. The aim is to improve traffic flow and safety by enabling vehicles to receive real-time updates on traffic conditions, signal timings, and road hazards. The technologies used include Dedicated Short Range Communications (DSRC), Cellular Vehicle-to-Everything (C-V2X), and Internet Protocol (IP)-based communication[15].

2.2.5.2.3 Long Range Communications

Long range communications in the Internet of Vehicles (IoV) bridge the gap between local vehicular networks and broader network services, including cloud-based platforms and traffic management centres[20].

- **Vehicle-to-Network (V2N) Communication:** V2N communication connects vehicles to the wider network, encompassing the internet and cloud services. It facilitates the exchange of information with traffic management centres, service providers, and other internet-based entities[21]. The objective is to provide services such as navigation, infotainment, software updates, and access to cloud-based applications. This communication relies on cellular networks (e.g., 4G LTE, 5G) for connectivity.

- **Vehicle-to-Electric Grid (V2G) Communication:** V2G communication enables electric vehicles (EVs) to interact with the electricity grid[22]. This interaction allows vehicles to either supply electricity to the grid or draw electricity from it during peak demand periods. The aim is to support energy management by balancing supply and demand and promoting the use of renewable energy sources. The technologies involved include smart grid technologies, bi-directional chargers, and advanced communication protocols.

- **Vehicle-to-Cloud (V2C) Communication:** V2C communication connects vehicles to cloud-based services, enabling data storage, processing, and access to various applications. The goal is to facilitate advanced data analysis, remote diagnostics, fleet management, and enhanced user services. Technologies employed include cloud computing platforms, cellular networks, and APIs for seamless integration[23].

Figure 2.2 illustrates the various communication types within the Internet of Vehicles (IoV). These types of communications collectively ensure the efficient, safe, and reliable operation of the IoV system, contributing to the advancement of intelligent transport and autonomous driving technologies.

Figure 2.2: Communication Types in IoV

To further illustrate the categorisation and specifics of these communications within the IoV system, Table 2.2 provides a detailed overview, summarising the scopes, descriptions, and technologies employed.

Table 2.2: Types of Communications in the Internet of Vehicles (IoV)

Scope	Communication Type	Description	Technologies Used
Short Range	Vehicle-to-Vehicle (V2V)	Involves direct exchange of information between vehicles to improve safety through collision avoidance, cooperative driving, and traffic management.	DSRC, Cellular V2X
	Vehicle-to-Pedestrian (V2P)	Facilitates communication between vehicles and pedestrians to enhance pedestrian safety and reduce urban accidents.	Smartphone applications, DSRC, Cellular V2X
	Vehicle-to-Device (V2D)	Interaction between vehicles and devices like smartphones and tablets, enhancing user experience with personalised services and infotainment.	Bluetooth, Wi-Fi, NFC
Medium Range	Vehicle-to-Infrastructure (V2I)	Exchange of data between vehicles and road infrastructure to improve traffic flow and safety through real-time updates.	DSRC, Cellular V2X, IP-based communication
Long Range	Vehicle-to-Network (V2N)	Connects vehicles to the wider network for exchanging information with traffic centres and accessing internet-based services like navigation.	Cellular networks (4G LTE, 5G)
	Vehicle-to-Electric Grid (V2G)	Enables electric vehicles to interact with the electric grid, supporting energy management by balancing supply and demand.	Smart grid technologies, Bi-directional chargers
	Vehicle-to-Cloud (V2C)	Connects vehicles to cloud services for data storage, processing, and accessing applications, facilitating remote diagnostics and fleet management.	Cloud computing platforms, Cellular networks, APIs

2.2.6 Background of MEC

The evolution of network architectures has profoundly influenced data processing and management, evolving from Cloud Computing to Mobile Cloud Computing, and subsequently to Edge Computing. Initially, Cloud Computing revolutionised IT by centralising resources, enhancing flexibility and access, despite issues such as high latency and data security concerns. In response, Edge Computing adopted a distributed approach to decrease latency and enhance performance, although it necessitates significant investment in peripheral equipment and introduces challenges in decentralised management. Mobile Cloud Computing combines the benefits of the aforementioned models by allowing local application operation on mobile devices, thereby improving responsiveness and resource management despite intermittent connectivity. MEC integrates these advancements, offering an innovative solution that caters to the performance needs of mobile applications through edge data processing and reduced latency. This integration optimises the real-time execution of critical applications, enhancing user experience and network resource utilisation. This convergence marks a new era in mobile networks, the IoT, and emerging applications, reshaping the landscape of mobile computing and wireless connectivity for the future.

The table 2.3 briefly summarises the fundamental elements of each technology, as well as their advantages, disadvantages, and application examples.

Table 2.3: Comparison of Data Processing Technologies: Advantages and Disadvantages

Technology	**Definition**	**Advantages**	**Disadvantages**	**Application Examples**
Cloud Computing	Data storage and processing on remote servers	Flexibility, accessibility, reduced costs	High latency, security and privacy issues	File storage, large-scale data processing
Edge Computing	Data processing at the edge of the network	Reduced latency, better reliability, data security	Additional investments, decentralised management and maintenance	Real-time data processing, critical applications
Mobile Cloud Computing	Local execution of applications on mobile devices	Improved responsiveness, efficient resource management, application portability	Intermittent connectivity, tight coordination between cloud resources and mobile devices	Mobile applications, real-time data processing
Mobile Edge Computing (MEC)	Moving cloud computing capabilities to the edge of the mobile network	Efficient execution of critical applications in real-time, optimised user experience, network resource optimisation	Additional investments, decentralised management and maintenance	Mobile applications, real-time data processing, IoT

2.2.7 Definition of MEC

MEC is a strategic technology approach that moves data processing and storage capabilities to the edge of the network, closest to the end user[24] [25] . Mainly used in mobile networks, this method improves application performance and significantly reduces latency.[26]. To summarise different definitions of MEC found in the literature, a synthesis table can be developed considering the various perspectives and terminologies used by researchers. Table 2.4 highlights the nuances and commonalities among MEC definitions, offering a comprehensive overview of the varied interpretations of this emerging technology.

2.2.8 key concepts of MEC

2.2.8.1 Proximity

The proximity of processing capacities at the edge of the network minimises the data transmission delay between user devices and processing points[27]. This results in a significant reduction in latency, making responses almost instantaneous. This feature is crucial for applications requiring high responsiveness, such as telemedicine or the control of critical industrial processes, where every millisecond counts.

2.2.8.2 Reduced Latency

By situating data processing and storage closer to the network's edge and near the end user, MEC significantly reduces latency[28]. This enhancement is pivotal for real-time interactive applications, such as online gaming, where response speed critically shapes the user experience. It's equally vital in augmented reality, where minimal latency is necessary to sustain seamless immersion and interaction with virtual elements. Additionally, for autonomous vehicles, the low latency facilitated by MEC allows for almost instantaneous responses to changing conditions, thereby enhancing the safety and efficiency of intelligent transportation systems.

Table 2.4: Summary of definitions of Mobile Edge Computing (MEC) in the literature

Author(s)	**Definition of MEC**
Yang, Binxu, et al.[29]	Approach to deploying applications and services at the network edge to meet the latency and bandwidth requirements of mobile applications
Zhang, Ke, et al.[30]	Strategic approach moving processing and data storage capacity to the edge of the network to improve application performance and reduce latency
Zhao et al.[24]	Move data processing and storage capabilities to the network edge, close to the user. This allows to offer services with minimal latency and better user experience network .
Mach et al.[31]	Involves deploying applications and services at network level, close to end users, to reduce latency and improve the efficiency of mobile applications
Nasir et al.[32]	Offers network-level computing, storage and communication services, closer to end users. This allows a significant reduction in latency and better use of network resources
Pham et al.[33]	MEC provides low-latency computing and storage infrastructure close to end-users. This enables efficient data processing and and real-time decision-making for mobile applications
Hu et al.[27]	Introduces a network architecture where applications and services are hosted close to the end-user. This approach ensures a better user experience and more efficient use of network resources
Mao et al.[34]	MEC exploits the computing and storage resources available age resources available at the network edge to offer low-latency, high-bandwidth services, which is crucial for demanding applications applications such as virtual reality and autonomous vehicles
Siriwardhana, Yushan, et al.[35]	Integrates data processing and storage capacities at network level, close to end users. As a result, it facilitates the deployment of low-latency, high-bandwidth applications such as cloud gaming and augmented reality

2.2.8.3 Improved Bandwidth

MEC enhances network efficiency by processing data at the network's edge, significantly reducing the volume of data that needs to be sent to the core network[36]. This decentralisation helps alleviate network congestion and optimises the use of available bandwidth. As a result, end users experience faster speeds and enhanced responsiveness, markedly improving service quality, especially in high-demand scenarios such as high-definition video streaming and real-time communications. These optimisations are crucial for supporting data-intensive applications and maintaining consistent quality of service, even during peak usage times.

2.2.8.4 Support for IoT Applications

Mobile Edge Computing offers an optimal solution for managing and processing the vast data generated by IoT devices, which are typically widespread and interconnected. This technology enables the immediate and local processing of data, crucial for maximising efficiency. By operating at the network's edge, MEC significantly reduces data transmission times and enhances responsiveness, which is vital for critical applications such as real-time health monitoring, smart urban infrastructure management, and industrial automation[37]. Moreover, this proximity in data processing helps secure sensitive information by minimising its exposure across expansive networks, thus bolstering IoT data defence against potential intrusions and leaks.

2.2.8.5 Optimisation of Network Resources

MEC enhances network resource utilisation by relocating data processing to the network's edge, which lightens the burden on the core network. By processing data close to the users and devices, MEC not only alleviates congestion within the core network but also boosts data transmission efficiency. This strategy ensures a more equitable distribution of workload across the network, proving invaluable during peak usage times. Furthermore, this decentralised processing approach enhances the network's resilience and scalability by dynamically adjusting processing

capabilities to meet varying demand and cater to the specific requirements of users and applications. Consequently, MEC provides a versatile and effective framework for optimal network resource management, leading to superior service quality and an improved user experience.

2.2.8.6 Scalability

Scalability is a key concept of MEC, enabling flexible and rapid deployment of new applications and services. Thanks to its distributed architecture, MEC can easily adapt to an increase in the number of users and data by adding additional computing and storage resources at the edge of the network [38]. This horizontal scalability contrasts with the vertical approach of centralised cloud computing, which requires powerful servers to be upgraded to handle a growing workload.

MEC takes advantage of virtualisation and containerisation technologies to deploy applications as lightweight, independent microservices [39]. This modularity makes it easy to add, update, and remove specific applications without affecting the rest of the system. Operators can therefore react quickly to changing user needs and market trends.

Finally, MEC's scalability is further enhanced by its integration with 5G networks. 5G brings enhanced network capabilities in terms of throughput, latency, and connection density, enabling MEC to support a growing number of devices and applications [40]. This synergy between MEC and 5G paves the way for innovative new use cases in areas such as autonomous vehicles, augmented reality, and Industry 4.0.

2.2.8.7 Privacy and Security

The localised data processing provided by MEC significantly enhances security and confidentiality [41]. By processing sensitive data at the network's edge, close to its source, MEC greatly reduces the risk of exposure during transit over long distances. This approach drastically lowers the likelihood of data interception and compromise by malicious actors. Furthermore, local processing facilitates the stringent enforcement of security policies, which can be specifically tailored to

meet regulatory requirements and adapt to the local environment. This strategy includes the deployment of advanced encryption techniques, consistent security audits, and strong authentication protocols. Collectively, these measures effectively safeguard data against unauthorised access and data breaches. Consequently, this not only secures data but also bolsters user and customer trust by ensuring that their information remains protected and confidential consistently.

Table 2.5 concisely outlines the foundational concepts and advantages of integrating MEC into network infrastructure.

Table 2.5: Synthesis of Key Concepts in Mobile Edge Computing (MEC)

Key Concept	**Description**	**Implications**	**Examples**
Proximity Processing	Processing capabilities near the network edge minimise data transmission delays, significantly reducing latency.	Enhances the performance of latency-sensitive applications by providing faster, more reliable responses, critical in environments where timing is crucial.	Telemedicine systems where immediate response can be life-saving; real-time monitoring and control of industrial processes.
Reduced Latency	Data processing and storage are located closer to the end-user, which drastically reduces latency.	Improves user experiences in real-time interactive environments by facilitating quicker reaction times and smoother communication.	Online gaming where real-time interaction is fundamental; augmented reality applications requiring seamless integration of virtual elements.
Improved Bandwidth	Local data processing significantly reduces the volume of data sent to the core network, easing network congestion.	Allows for better management of network traffic and higher data transfer speeds, especially critical during high usage periods.	High-definition video streaming without buffering; efficient real-time communication systems.

Support for IoT Applications	Efficient management and processing of data from interconnected IoT devices by operating at the network's edge.	Enhances the operational efficiency of IoT systems by reducing latency and increasing the speed of data processing, which is crucial for timely decision-making.	Real-time health monitoring devices; smart city infrastructure management like traffic and energy systems; industrial automation processes.
Optimization of Network Resources	Data processing at the network's edge lightens the load on the core network, leading to better overall network efficiency.	Promotes a more resilient and adaptive network infrastructure, capable of handling high demand efficiently without compromising performance.	Smart resource allocation during large public events or emergencies to maintain service quality and connectivity.
Scalability	The distributed nature of MEC allows for the easy expansion of network capabilities by adding more resources at the network edge.	Supports dynamic network growth and the rapid deployment of new services without significant infrastructure overhaul, making it suitable for evolving technological landscapes.	Adding computational resources to support a sudden increase in users or devices, such as during a festival or major sporting event; rapid deployment of new applications in a city.
Privacy and Security	Sensitive data is processed close to its source at the network's edge, minimising the risk of exposure during transit.	Reduces the likelihood of data breaches and unauthorised access, thereby enhancing user trust and compliance with data protection regulations.	Secure processing of financial transactions; local processing of personal health information in wearable devices, ensuring data privacy.

Overall, MEC stands as a pivotal force in advancing technology, catalysing the acceleration and efficiency of digital services. It spearheads a paradigm shift in

data processing, empowering real-time responsiveness and heightened connectivity crucial for the burgeoning IoT. MEC lays the groundwork for groundbreaking applications across various domains such as connected health, smart cities, and Industry 4.0 by furnishing an infrastructure adept at managing escalating data volumes and computational demands at the edge. Seamlessly integrating into existing architectures and adaptable to emerging technologies, MEC emerges not only as a vital solution for present requirements but also as a linchpin in shaping the future of our digital interactions.

2.2.9 Relationship between IoV and MEC

The relationship between the IoV and MEC is crucial for advancing technologies such as autonomous driving and intelligent traffic management. MEC supports IoV by providing infrastructure that processes vast amounts of vehicular data efficiently and in real time.

2.2.9.1 Proximity Data Processing

MEC processes data near its source, in this case, the vehicles, significantly reducing latency. This reduction is crucial for applications that require rapid responses, such as emergency systems or real-time route adjustments. For example, autonomous driving necessitates immediate decision-making for actions like braking or obstacle avoidance to ensure safety. By processing data locally, MEC ensures these critical decisions are made swiftly and effectively, enhancing overall vehicular safety and efficiency[42].

2.2.9.2 Autonomous Driving Support

In autonomous driving, MEC processes and analyses data from vehicle sensors (cameras, lidars, radars) right at the network edge. Local processing of sensory data eliminates the need to relay information to distant data centres, facilitating quicker and more reliable driving decisions based on real-time data.

2.2.9.3 Intelligent Traffic Management

MEC is essential for intelligent traffic management by analysing multiple vehicle data streams to optimise routes, decrease congestion, and enhance energy efficiency. Real-time traffic data can be locally processed to fine-tune traffic signals and streamline vehicle flows effectively.

2.2.9.4 Communication Between Vehicles and Infrastructure

MEC enhances V2V and V2I communications, allowing for the rapid exchange of critical information that helps synchronise traffic and prevent accidents. These communications may include alerts about imminent hazards, severe weather, or traffic build-ups.

In examining the potential synergies between the Internet IoV and MEC, it is critical to comprehensively understand both the foundational interactions and the distinct advantages and obstacles accompanying their integration. The subsequent table offers a detailed overview of these elements, emphasising their respective contributions to the development of communication and automotive technologies.

Overall, MEC acts as a catalyst for IoV, not only boosting the performance and safety of intelligent transportation systems but also enhancing their efficiency and reliability. This synergy between MEC and IoV enables numerous advanced applications, pushing the boundaries of automotive technology and urban mobility management.

Table 2.6 provides an in-depth analysis of each aspect of this relationship. Proximity data processing considerably reduces latency, which is critical for safety-sensitive applications such as emergency response systems, as detailed in Table 2.6. However, this approach faces significant challenges in managing vast amounts of data in real-time. Similarly, while intelligent traffic management and autonomous driving substantially enhance operational efficiency and safety, they necessitate robust, continuous connectivity and sophisticated data processing capabilities.

Table 2.6: Comprehensive Analysis of the Relationship Between Internet of Vehicles (IoV) and Mobile Edge Computing (MEC)

Aspect	Description	Benefits	Challenges
Proximity Data Processing	MEC processes data close to the vehicles, significantly reducing latency for real-time applications like emergency systems and route adjustments.	Reduced response times for safety-critical functions.	Managing large volumes of data in real time.
Autonomous Driving Support	MEC processes and analyses data from vehicle sensors (cameras, lidars, radars) at the network edge, aiding immediate decision-making for autonomous driving.	Enhances vehicle response to dynamic road conditions.	Requires highly reliable and continuous connectivity.
Intelligent Traffic Management	MEC analyses multiple vehicle data streams to optimise routes, reduce congestion, and enhance traffic flow efficiency by managing traffic signals in real time.	Improves traffic flow and reduces travel time.	Complexity in data integration and real-time analysis.
Communication Between Vehicles and Infrastructure	MEC facilitates fast and efficient V2V and V2I communications, allowing for immediate exchange of critical safety and operational data.	Increases road safety and coordination.	Security and privacy concerns of data transmission.

2.3 MEC Architectures

MEC architectures excel by positioning data processing and computational resources near end-users and data sources[43]. This proximity is especially advantageous in environments such as the Internet of Vehicles, where high bandwidth

and low latency are critical. By localising these resources, MEC enhances response times and efficiency, significantly improving performance in data-intensive and time-sensitive scenarios.

2.3.1 MEC Architecture Components

Mobile Edge Computing (MEC) is a revolutionary architecture that brings data processing and storage capabilities closer to end users and devices. This approach reduces latency, increases bandwidth, and improves the overall performance of mobile applications and Internet of Things (IoT) services. The MEC architecture consists of several key components, each of which plays a crucial role in the overall system[44].

2.3.1.1 MEC Servers

MEC servers are located close to end users, often within cellular base stations or local data centres[45]. These servers are responsible for running low-latency applications and services. Thanks to their strategic location, they enable data to be processed locally, reducing transit time and improving application responsiveness.

2.3.1.2 MEC Management Platform

The MEC management platform is the brain of the architecture. It manages computing and storage resources, orchestrates services and applications, and ensures communication between the various network components[27]. It is also responsible for implementing security policies, monitoring performance, and managing software updates.

2.3.1.3 MEC Applications

MEC applications are designed to take advantage of the local processing capabilities offered by MEC servers. These applications can include streaming services, online games, augmented/virtual reality applications, and telemedicine solutions. By operating at the edge of the network, these applications benefit from reduced latency and improved quality of service.

2.3.1.4 MEC Application Programming Interface (API)

MEC APIs enable developers to easily create and deploy applications on the MEC infrastructure. These standardised interfaces facilitate the integration of third-party services and applications, encouraging innovation and the creation of new services that can be rapidly implemented at the edge of the network[46].

2.3.1.5 MEC Security System

Security is a crucial element in the MEC architecture. The MEC security system includes data protection, access control, and intrusion detection mechanisms. These measures ensure that user data and network operations remain protected against cyber threats.

2.3.1.6 Communication Networks

Communication networks provide connectivity between end users, MEC servers, and other network components. They include cellular networks, Wi-Fi networks, and wired networks[44]. The performance and reliability of these networks are essential to ensure an optimal user experience and efficient data transmission.

2.3.1.7 MEC User Interface (UI)

The MEC user interface provides administrators and end users with the tools they need to interact with the system[47]. For administrators, it provides dashboards for monitoring network performance, managing resources, and configuring services. For end users, it provides intuitive interfaces for accessing MEC applications and services.

Figure 2.3 illustrates the key components of the Mobile Edge Computing (MEC) architecture, highlighting management platforms, servers, applications, APIs, security systems, communication networks, and user interfaces.

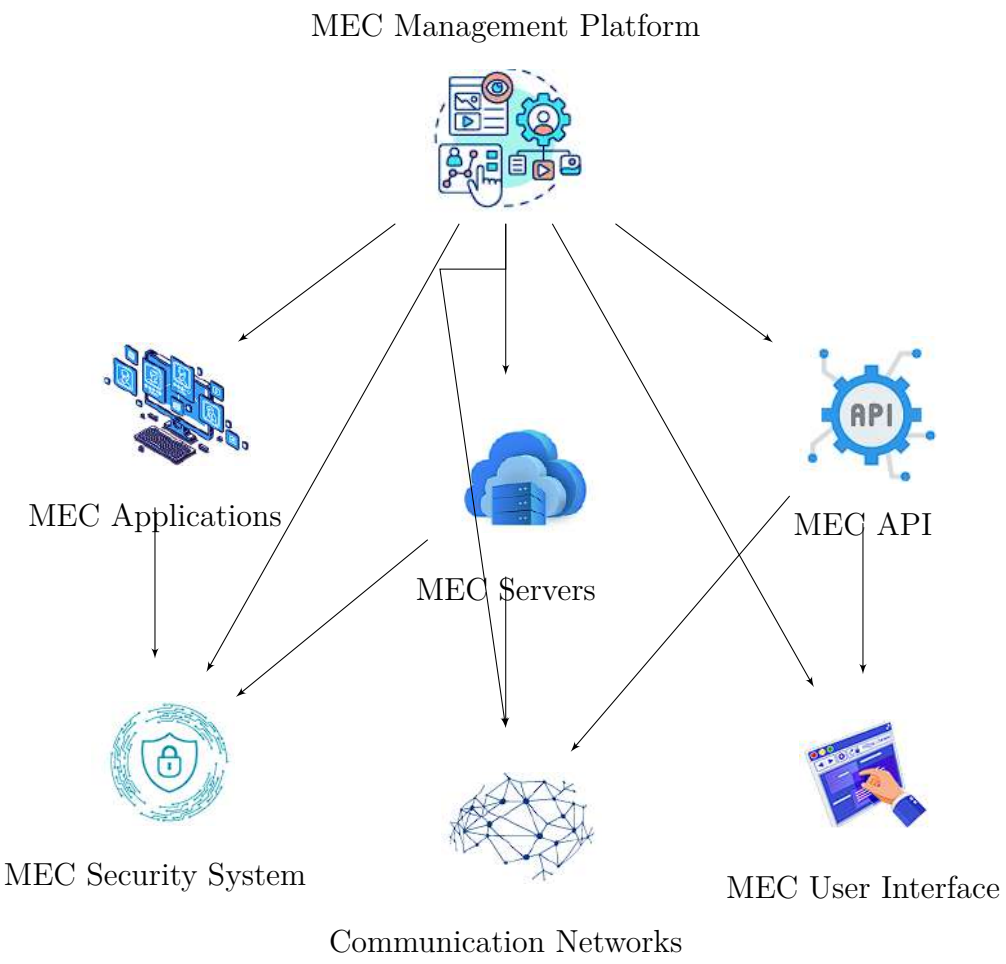


Figure 2.3: MEC Architecture Components

Figure 2.4 below illustrates the flow and interaction between the various components of the MEC architecture.

The Mobile Edge Computing (MEC) architecture is made up of various interconnected components that work together to deliver low-latency, high-performance services. By bringing processing capabilities closer to end users, MEC is transforming the way applications and services are deployed and consumed, opening up new possibilities in sectors ranging from healthcare to the entertainment industry.

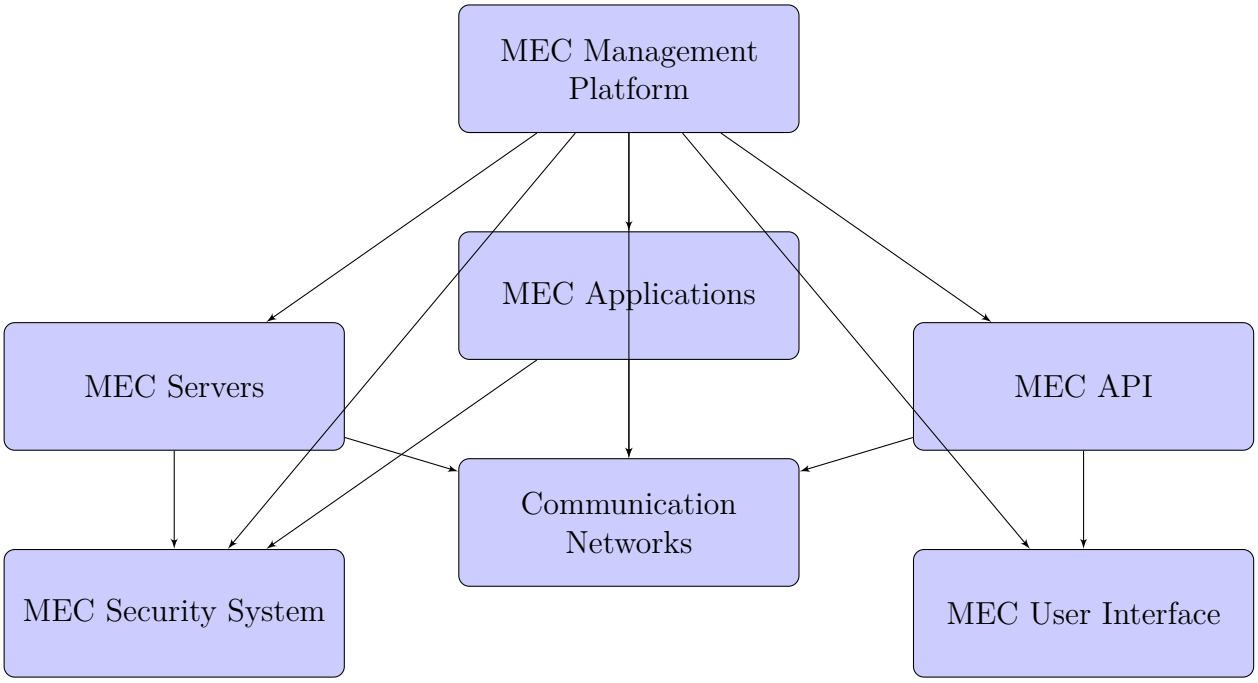


Figure 2.4: MEC Architecture Component Flowchart

2.4 Scalable Mobile Computing: From Cloud Computing to Mobile Edge Computing

2.4.1 The Mobile Edge Computing Overview

2.4.1.1 MEC service scenarios

There are numerous service scenarios for Mobile Edge Computing (MEC). The following typical examples are listed in [48]:

- Augmented Reality: The MEC server stores audio and video content for augmented reality, matching it one-to-one using positioning technology and location data. The server analyses the application content using deep packet inspection in response to the terminal's application request, calculates the location-based AR content, and delivers it to the user. This solution reduces content latency and enhances the user experience through content localisation.

- Accelerated Video with Intelligence: Files and media are often transmitted via HTTP streaming or downloading based on the TCP protocol over the

Internet. Connection capacity can fluctuate due to changes in the channel environment, end access, and output. MEC can address these issues by accelerating video delivery, as TCP might not quickly react to abrupt changes in the radio access network (RAN). MEC enhances video streaming by dynamically adjusting to these fluctuations, thereby improving video quality and user experience.

- Connected Vehicles: MEC servers can be installed at LTE base stations along roadsides. These servers collect local data from in-vehicle applications or roadside sensors, analyse it, and then broadcast relevant information to nearby vehicles. This setup informs drivers of emergencies or other critical conditions in real time, thus improving safety and situational awareness.

- IoT Convergence Gateway: IoT devices typically have limited processor and memory capacity, necessitating the use of aggregation gateways to collect and manage information from multiple IoT devices. These gateways reduce the processing and analysis burden on individual devices, thereby improving overall efficiency and responsiveness by centralising data handling.

2.4.1.2 The Basic Structure of the MEC

As shown in Figure 2.5, the MEC server platform is defined in [1]:

The MEC hosting infrastructure, as depicted in Figure 2.5, primarily consists of MEC hardware resources and the MEC virtualisation layer. The MEC Virtualisation Manager and the MEC Application Platform Services comprise the MEC Application Platform.

The MEC Virtualisation Manager provides Infrastructure-as-a-Service (IaaS) capabilities, offering a flexible, effective, multi-tenant hosting and operating environment for applications. For upper-layer applications running on the MEC server, the MEC Application Platform Services offer the following set of middleware services:

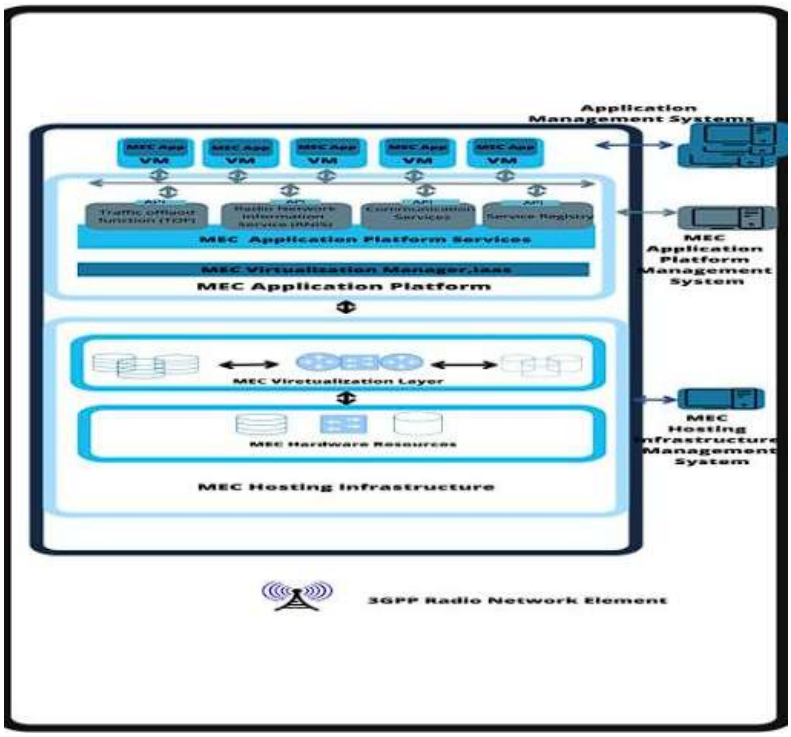


Figure 2.5: MEC server platform overview

- Infrastructure Services (IS): Infrastructure Services provide the foundational support for MEC operations, including compute, storage, and network resources. They ensure that applications have the necessary infrastructure to perform optimally.

- Communication Services (CS): Communication Services enable interaction between MEC server applications and this application platform . These services facilitate seamless data exchange and communication within the MEC ecosystem, enhancing the integration of various applications.

- Service Registry (SR): The Service Registry offers visibility of the MEC server's available services. It uses the concept of flexible coupling of services to allow for adaptable application deployment, enabling efficient discovery and utilisation of MEC services.

- Radio Network Information Services (RNIS): RNIS provides essential authentication and network information, such as cell ID, user location, cell load, and

throughput. It delivers this data to the wireless access network, supporting users and communities with relevant network insights.

- Traffic Offload Function (TOF): The TOF service selects priority traffic and assigns routing to applications authorised to receive data. It manages policy-based user data flow, ensuring that critical data is prioritised and routed appropriately.

The virtual appliance application operates on the virtual machine image within the IaaS. A unified API is used between each VM and the MEC application platform.

Based on the architecture of the MEC server platform illustrated in Figure 2.5, we describe the three-tier logical entities of the MEC server:

- MEC Platform Base Layer: This layer, built on NFV hardware resources and a virtualisation layer architecture, provides compute, storage, control functions, and hardware virtualisation components of the underlying hardware (including OpenStack-based virtual operating systems, KVM, etc.) to perform virtualised computing processes. It includes caching, virtual exchange, and related management functions.

- MEC Functional Components: These components manage the carrier service external interface adaptation function via the API. They complete the interface protocol encapsulation between the base station and the upper application layer, providing traffic bypass, VM communication services, wireless network information, service registration capabilities, and application support. The corresponding underlying functions include data packet analysis, content routing, top-level application registration management, and wireless information interaction. The associated API utilises SNMP network management, interacting with parameters and information through Get / Set Request / Set Response message instances.

- MEC Application Layer: In line with the virtualised VM application architecture of network functions, MEC functional components are integrated and packaged into virtual applications (such as local download, wireless caching, augmented reality technology, service optimisation, positioning, etc.). These applications are made accessible to third-party software developers or commercial applications, facilitating the utilisation and invocation of wireless network capabilities.

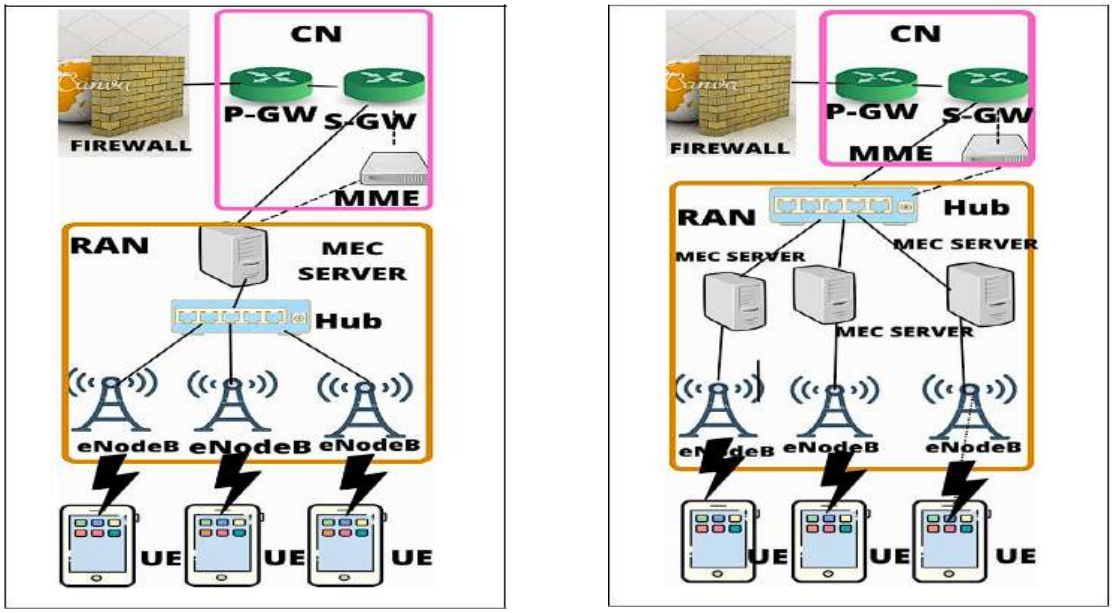
2.4.1.3 MEC Implementation Scenarios

The MEC server can be deployed in several locations. Some deployment scenarios for the MEC server are shown below:

- **MEC Solution Implemented on the RAN Side Based on 4G EPC Architecture**

  - The MEC server is deployed after the convergence point of the RAN-side base station: As illustrated in Figure 2.6a, this is a relatively common implementation method.

  - The MEC server is placed behind a single base station on the RAN side: As illustrated in Figure 2.6b, this deployment is ideal for hotspot areas such as campuses and shopping centres. This architecture offers the advantage of easier access to wireless-related information at the base station side by monitoring and analysing the S1 interface signalling. However, security concerns such as billing and legal monitoring need to be further addressed.



(a) After the Ran-Side Base Station

(b) behind a single base station on the RAN side

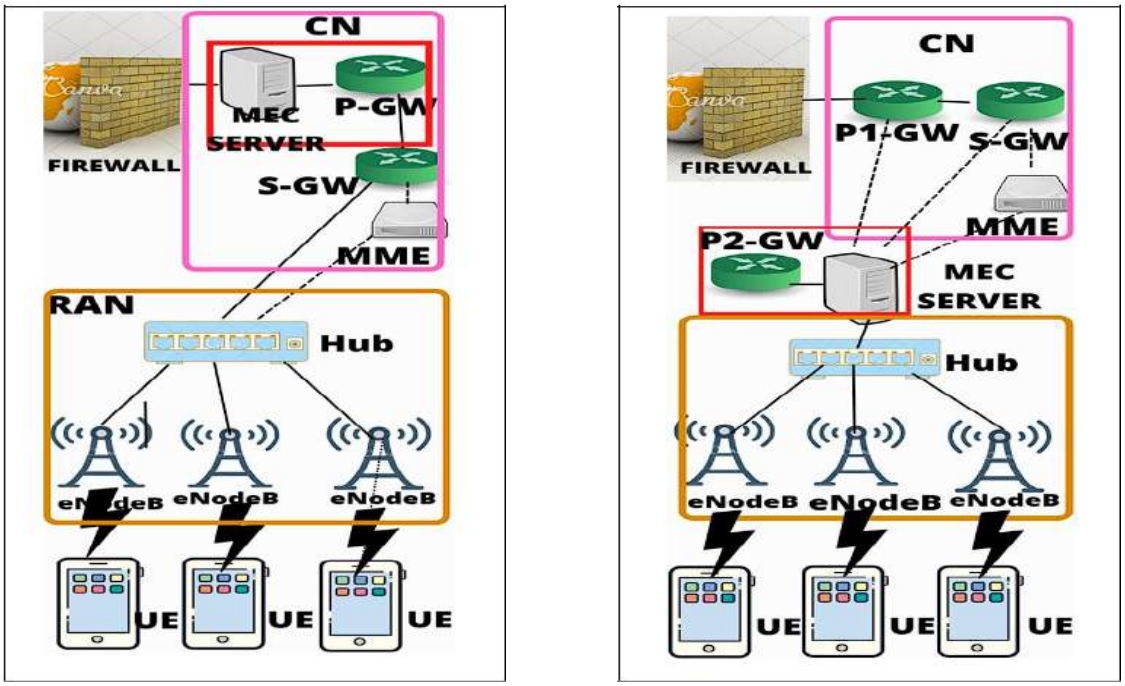
Figure 2.6: Implemented MEC Server

- **MEC Solution Implemented on the CN Side Based on 4G EPC Architecture**

  - The MEC server is deployed alongside the P-GW on the CN side: As illustrated in Figure2.7a, this approach maintains the existing EPC architecture. The MEC server is integrated with the P-GW. The data service initiated by the UE travels through the eNodeB, Hub Node, S-GW, P-GW + MEC, and then to the public Internet network. This deployment method avoids issues related to billing and security.

  - The MEC server is deployed alongside the D-GW on the CN side: As depicted in Figure 2.7b, this approach modifies the existing EPC architecture. The MEC server is integrated with the D-GW, and the original P-GW is divided into P1-GW and P2-GW (i.e., D-GW). P1-GW remains in its original location, while D-GW is relocated (either to the RAN side or the CN edge). D-GW manages functions such as charging, monitoring, and authentication. The MEC server and D-GW

can be integrated or deployed as separate network elements behind the D-GW. The private interface between P1-GW and D-GW necessitates equipment from the same manufacturer.



(a) the P-GW on the CN side **(b)** the D-GW on the CN side

Figure 2.7: Implemented MEC server together with

- **Deployment Plan of the MEC Server Based on 5G Architecture**

  – The MEC server is deployed in GW-UP: As illustrated in Figure 2.8, following the separation of the C/U function from the core network in the 5G architecture, the U-Plane function (corresponding to GW-UP) is moved down. It can be positioned either at the RAN side or at the CN edge, while the C-Plane (corresponding to GW-CP) remains on the CN side. The MEC server is located within the GW-UP. This deployment, compared to traditional public network solutions, offers users high bandwidth and low latency services.

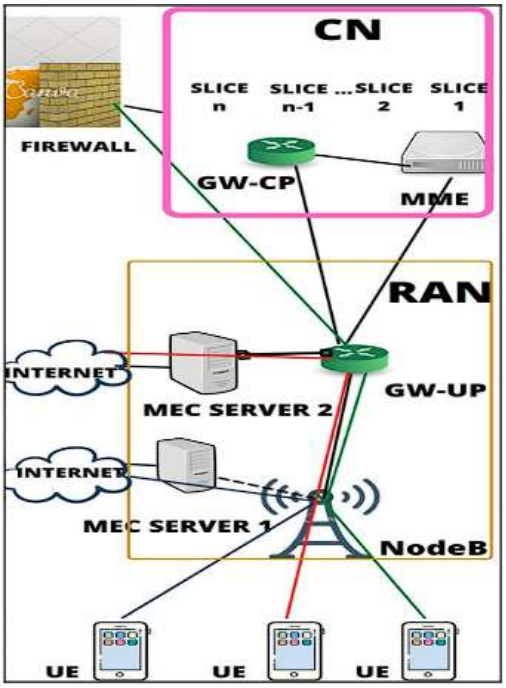


Figure 2.8: Deployment plan of the MEC server based on 5G architecture

2.4.2 Research Trends

Table 2.7: Researcher's Optimized Criteria in Different Technologies

Ref.	Mobile /Cloud /Edge Com-put-ing	Optimized Criteria						ApplicationArea
		Latency	Connectivity	Security	Transmitteddata volumes	Scalability	Cost	
[49]	CC			++		++++	+	
[50]	CC			++		++++	+	Online Book-shop
[51]	MEC	++++	++++	+++	+	++++	++	
[52]	MEC	++++	++++	+++	+	++++	++	
[53]	MEC	++++	++++	+++	+	++++	++	
[54]	MEC	++++	++++	+++	+	++++	++	IoT
[55]	MEC	++++	++++	+++	+	++++	++	IoT
[56]	MEC	++++	++++	+++	+	++++	++	
[57]	EC	++++	++++	++++	++++	++++		Video and Game Analytics

[58]	EC	++++	++++	++++	++++	++++		IoT	Application
[59]	CC			++		+	++++		
[60]	EC	++++	++++	++++	++++	++++		IoT	Application
[61]	EC	++++	++++	++++	++++	++++		IoT	Application
[62]	EC	++++	++++	++++	++++	++++			
[63]	MEC	++++	++++	+++	+	++++	++	IoT	Application
[64]	MCC			+		+	++++		
[65]	CC			++		+	++++		
[66]	MCC			+		+	++++		
[67]	MCC			+		+	++++		
[68]	MCC			+		+	++++		
[69]	MCC			+		+	++++		
[70]	MCC			+		+	++++		
[71]	MEC	++++	++++	+++	+	++++	++	IoT	Application
[72]	CC			++		+	++++		

[73]	MCC			+		+	++++	
[74]	EC	++++	++++	++++	++++	++++		
[75]	MCC			+		+	++++	
[76]	EC	++++	++++	++++	++++	++++		
[77]	EC	++++	++++	++++	++++	++++		Education System
[78]	EC	++++	++++	++++	++++	++++		
[79]	EC	++++	++++	++++	++++	++++		Sensing Applications
[80]	EC	++++	++++	++++	++++	++++		IoT
[81]	MEC	++++	++++	+++	+	++++	++	IoT Application
[82]	MEC	++++	++++	+++	+	++++	++	AR/VR

2.4.2.1 Discussion

Almost every object on Earth, including our residences, vehicles, and even our physical selves, will be connected to the Internet, consistently exchanging information about our everyday actions. Storing such a large amount of data poses a considerable difficulty for central data centres and network administrators who are already grappling with higher workloads caused by resource-intensive apps. The current data infrastructure is unable to cope with the unexpected surge in data, necessitating the immediate implementation of the next stage of computing at the network edge. It is logical to conclude that a novel style of digital engagement necessitates a distinct kind of infrastructure. Conventional data centres were well-suited for the self-contained realm of internal applications and communications. However, with the rise in popularity of e-commerce and other high-volume services, computing transitioned to the cloud.

Presently, modern apps produce a subsequent iteration of services, a significant portion of which will operate inconspicuously in the background. These services are built upon the principles of uninterrupted accessibility, swift efficiency, and predominantly autonomous operation.

The study highlighted in Table 2.7 underscores the significance of MEC technology. The graph depicted in Figure 2.9 offers a more lucid representation of the corresponding requirements for each technology: MEC, MCC, EC, or CC. It is evident that EC technology surpasses CC technology. Although Edge Computing incurs significant expenses, including maintenance costs, it provides numerous advantages in terms of latency, connection, security, data volume transmission, and network scalability. The graph also showcases MEC technology, which offers a superior architecture for a mobile environment when compared to MCC.

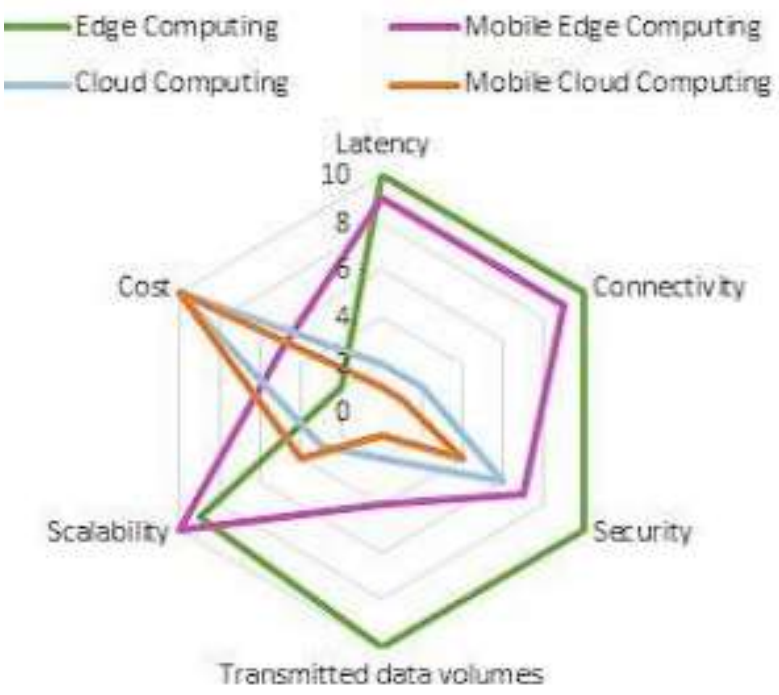


Figure 2.9: The Strongest Criteria for Each Technology (MEC,MCC,EC,or CC

2.5 Conclusion

Chapter 2 provided a detailed overview of Internet of Vehicles (IoV) and Mobile Edge Computing (MEC) technologies, highlighting their central role in transforming modern transport systems. By analysing the foundations, key concepts, and practical applications of these technologies, we have emphasised how IoV and MEC work together to enhance road safety, optimise traffic management, and support the development of autonomous vehicles. The synergy between IoV and MEC has emerged as a crucial element in meeting the low latency and local data processing requirements essential for IoV applications. MEC architectures bring computing resources closer to the vehicles, thereby reducing latency and enabling real-time responses, which are vital for autonomous driving and intelligent traffic management.

In conclusion, this chapter has established the theoretical and practical foundations necessary to understand the importance and impact of IoV and MEC in the field of intelligent transport. This understanding is fundamental for the

development of advanced and optimised caching strategies, as explored in the subsequent chapters of this thesis. Advances in these areas promise to lead to more efficient, safe, and responsive transport systems, thus fostering a quicker adoption of autonomous vehicles and better urban mobility management.

Chapter 3

Caching Management and Reinforcement Learning in IoV

Contents

3.1 Introduction

The emergence of the Internet of Vehicles (IoV) represents a revolution in the management of transport systems, where efficient communication and data processing are becoming critical. In this chapter, we explore two technological pillars at the heart of this revolution: caching management and reinforcement learning. These technologies offer promising solutions for overcoming the challenges associated with network latency, bandwidth management and the security of data exchanges.

With the exponential increase in data generated by connected vehicles, efficient management of this data is becoming imperative. Caching, which aims to temporarily store data in strategic locations to reduce latency and load on the network, needs to be optimised to dynamically adapt to changing network conditions. At the

same time, reinforcement learning, a branch of artificial intelligence, is emerging as a powerful tool for automating and optimising caching decisions based on ongoing interactions with the environment.

The aim of this chapter is to demonstrate how the combination of caching management and reinforcement learning can significantly improve IoV performance. We will analyse intelligent caching strategies that learn from experience and adapt in real time to the needs of the network. The aim is to show how these strategies can maximise the use of network resources while guaranteeing an optimal user experience.

This chapter will detail existing caching technologies, introduce the fundamentals of reinforcement learning, and examine how their integration can revolutionise operations in the IoV. It will use case studies and performance analysis to assess their potential impact.

3.2 Caching Technologies: Enhancing Connectivity from Wired Networks to IoT and IoV

Caching is a foundational technique in computing and communication systems, enhancing data access efficiency and speed. This method stores copies of frequently used files in a temporary storage area called a 'cache', enabling quicker access than retrieving data from more distant or slower sources. Caching is applied across various domains, such as web browsers, databases, and network architectures.

The pivotal role of caching in augmenting the efficiency and speed of computer networks has been paramount, evolving significantly across different technological eras. Originally deployed in wired networks to reduce latency and expedite access to information, caching facilitated rapid retrieval by storing frequently accessed data temporarily. This approach minimised the need for data retransmission across the network.

As technology progressed towards wireless networks, caching strategies adapted to challenges such as fluctuating connectivity and bandwidth limitations. Thus,

caching methodologies have evolved, becoming more dynamic and adaptive to efficiently manage data in environments with unstable network conditions.

The rise of the IoT has increased the complexity of cache management due to the massive volumes of data generated by dispersed devices. Distributed caching has become essential, optimising data processing and access by distributing cache storage across multiple nodes, thus reducing the load on central servers and improving response times [83].

The emergence of the IoV highlights the need for ultra-fast, highly responsive caching systems. In the IoV, vehicles constantly communicate in real time with each other and with connected infrastructures, requiring caching solutions that support high speeds and mobility. These systems ensure reliable and instantaneous information exchanges, enhancing the safety and efficiency of intelligent transportation systems. This development marks a further evolution in the importance of sophisticated caching strategies tailored for modern networks.

Table 3.1 provides an overview of the main caching techniques used, detailing their descriptions, benefits, and typical applications.The table provides a detailed comparison of various caching techniques, outlining their descriptions, benefits, and typical applications. For example, LRU is known for balancing simplicity and performance, making it suitable for operating systems and web browsers. FIFO, due to its straightforward implementation, is ideal for streaming data. Advanced methods like ARC and LIRS are designed for high-performance needs in complex environments. This overview helps understand how different caching strategies enhance network efficiency and connectivity, crucial for modern networks from wired to IoT and IoV.

In summary, LRU is widely used thanks to its good balance between simplicity, performance, and memory utilisation. The other techniques provide optimisations for specific cases, often at the cost of increased complexity.

Table 3.1: Overview of the Main Caching Techniques, Their Description, Benefits, and Typical Applications

Technique	**Description**	**Benefits**	**Typical Applications**
LRU	Deletes the data that has not been used for the longest time	- Simple to implement - Efficient in many scenarios - Balances performance and memory usage	- Operating systems (memory page management) - Databases - Web browsers
FIFO (First-In-First-Out)	Removes the oldest data, similar to a queue	- Very simple to implement	- Streaming data - Buffer memory management
LFU	Deletes the least frequently used data	- Good for stable environments - Data caches	- Scenarios where some data is accessed sporadically but must remain in cache
Random Replacement (RR)	Randomly deletes data	- Very simple to implement - No need to track access	- Where accurate prediction is difficult
2Q	Uses two queues: one for pages accessed once, and one for those accessed more frequently	- Improves LRU performance in some cases	- Scenarios with complex access patterns
ARC (Adaptive Replacement Cache)	Cache management algorithm that optimises memory by dynamically balancing recent and frequently accessed data	- Combines the advantages of LRU and LFU - Adapts dynamically	- Advanced file systems - High-performance disk caches
LIRS (Low Interreference Recency Set)	Differentiates between recency and recent inter-reference for better eviction decisions	- Maintains a high success rate even with difficult access models	- Scenarios with large access cycles

3.3 Caching in IoV

In the Internet of Vehicles (IoV), caching entails the preemptive storage of critical data at strategically located points within the vehicular network, including individual vehicles, base stations, and roadside units [84]. This localised storage is designed to minimise the latency involved in accessing essential information such as traffic conditions, navigation updates, safety alerts, and sensory data from the immediate surroundings of the vehicle. By doing so, it ensures timely delivery of vital information, crucial for the efficient functioning of vehicular systems.

3.3.1 Importance of Caching in IoV

Caching is crucial in the Internet of Vehicles (IoV) as it improves the efficiency, speed, and reliability of data transfers in vehicular networks [85]. The following are the primary reasons why caching is essential in the context of the Internet of Vehicles (IoV):

3.3.1.1 Reduced Latency

In the Internet of Vehicles (IoV), instantaneous data processing is essential for critical operations including navigation, traffic management, and safety functionalities such as collision avoidance systems. By caching frequently accessed data locally or at edge nodes, the time required to retrieve this data is significantly decreased, thereby reducing latency and enhancing the responsiveness of these applications [86].

3.3.1.2 Optimising Bandwidth

In IoV systems, the strategic local storage of data in caches significantly reduces bandwidth consumption by eliminating the need for continuous data transfers across the network [87]. This approach is particularly vital in automotive environments where network resources are often limited, and the volume of data generated and consumed is substantial. Such optimisation is crucial for maintaining system efficiency and ensuring the sustainability of network operations.

3.3.1.3 Enhanced System Efficiency

Caching significantly augments system performance by alleviating the load on backend servers [88]. By enabling vehicles to access stored data from proximate locations rather than relying on central servers, caching facilitates faster data retrieval and minimises server load. This reduction in server dependency not only decreases maintenance costs but also enhances the scalability of the system, which is critical for accommodating the expanding scope of IoV applications.

3.3.1.4 Improved Data Accessibility

Caching ensures the continuous availability of crucial data—including traffic conditions, weather updates, and navigation maps, despite intermittent network connectivity. This persistent data availability is critical for the uninterrupted operation of IoV applications, particularly in areas plagued by limited network coverage. By mitigating connectivity issues, caching not only enhances the reliability of vehicular networks but also supports seamless user experiences across diverse geographic regions.

3.3.1.5 Enhanced Energy Efficiency

Retrieving data from local caches significantly reduces energy consumption compared to data transmission over long distances. This reduction in energy demand markedly improves the energy efficiency of vehicular communication systems, thereby extending battery life and reducing operational costs. Such efficiency is crucial in promoting sustainable automotive technologies and enhancing the ecological footprint of IoV systems.

3.3.1.6 Enhanced Durability and Dependability

Caching introduces redundancy in data storage, thus bolstering the resilience of IoV systems against network disruptions. In the event of network failures, cached data provides a reliable backup that ensures continuous and efficient vehicle operation [89]. This enhanced dependability is vital for maintaining system integrity and user trust in IoV technologies.

3.3.1.7 Enhanced Support for Sophisticated Applications

Sophisticated IoV applications, such as autonomous driving, cooperative vehicle manoeuvring, and real-time traffic management, rely heavily on the immediate availability of accurate data [90]. Caching supports these applications by providing instant access to necessary data, thereby enabling more advanced, reliable, and secure vehicle operations. This immediate data accessibility is essential for the functionality of complex algorithms and systems that require real-time computational inputs.

3.3.1.8 Optimisation of Network Traffic Management

Caching plays a critical role in optimising data traffic during peak demand periods. By distributing the data load across multiple cache locations, it significantly mitigates network congestion. This strategic dispersal not only ensures a steady flow of data but also maintains high-quality service across the network.

As a result, caching enhances the operational efficiency, safety, and overall user satisfaction within the Internet of Vehicles. The strategic integration of caching is pivotal for harnessing the full potential of IoV technologies in modern transportation networks, promoting a seamless and efficient infrastructure.

3.3.2 Existing Caching Strategies in IoV

In the Internet of Vehicles, the implementation of effective caching strategies is paramount for enhancing communication efficiency and facilitating access to real-time data. This section provides a comprehensive overview of the predominant caching strategies employed within the IoV context:

3.3.2.1 Popularity-Based Caching in the Internet of Vehicles : Technical and Operational Implications

Popularity-based caching is a pivotal strategy within the IoV aimed at optimising system responsiveness and resource utilisation. This approach entails storing locally the data most frequently requested by users, thereby minimising latency and reducing the burden on central data processing facilities[91]. The following

exposition delineates the mechanics of this strategy along with its broader technical and operational implications:

3.3.2.1.1 Operations and Techniques of Popularity-Based Caching

- Popularity Detection
  - Query Analysis : Advanced algorithms are deployed to continuously monitor and analyse user requests, enabling the detection of demand patterns. This analysis helps in identifying the types of data that would most benefit from caching, ensuring efficient data retrieval.
  - Popularity Metrics: Data is assessed based on its frequency of access. Items that surpass a predetermined popularity threshold are earmarked for caching, optimising resource use and access speed.
- Data Caching
  - Targeted Selection: Specific types of data, such as frequently travelled routes, recurring traffic updates, and regional weather forecasts, are prioritised for caching. This selection is based on their expected utility and demand consistency.
  - Dynamic Adaptation: The cache is dynamically updated in response to shifts in demand patterns. This ongoing adaptation ensures the cache retains only the most pertinent and timely data, thus maintaining its relevance and effectiveness.
- Benefits
  - Reduced Latency: Storing frequently requested data locally significantly shortens the response time to user queries, thereby enhancing the system's responsiveness.

  - Reduced Server Load: By decreasing the number of requests that need to be processed by central servers, popularity-based caching lessens server workload, which in turn improves the system's overall efficiency and performance.

- Challenges and Solutions in Popularity-Based Caching
  - Data Volatility:
    * Challenge: The dynamic nature of data popularity necessitates frequent adjustments to the caching system to preserve its operational efficacy. As user preferences and data access patterns evolve, previously cached data may no longer align with current demands.
    * Solution: Implementation of sophisticated predictive algorithms is crucial. These algorithms forecast shifts in data popularity using historical access patterns and real-time analytics, enabling proactive adjustments to the cache contents.
  - Cache Space Management
    * Challenge: The finite capacity of cache storage mandates judicious management of its contents to ensure optimal performance. Effective cache space utilisation is pivotal in maintaining system efficiency.
    * Solution: Adoption of strategic eviction policies plays a fundamental role in cache management. Policies such as LRU and LFU are instrumental in optimising space utilisation. These policies facilitate the removal of data that is least likely to be accessed again, thereby making room for more pertinent data.

- Applications of Popularity-Based Caching in IoV
  - Navigation and Traffic Management: Navigation information and real-time traffic updates are indispensable components of IoV systems[92]. By employing popularity-based caching, critical data such as frequently

travelled routes and current traffic conditions can be readily accessed, facilitating efficient route planning and congestion management. This enhances user experience by providing timely and accurate navigation assistance, ultimately leading to smoother and safer journeys.

– Weather Forecasting: Weather forecasts play a pivotal role in ensuring safe and efficient travel. Through popularity-based caching, IoV platforms can pre-cache weather data for specific regions, empowering drivers to make informed decisions regarding route selection and journey planning. By considering weather conditions in advance, drivers can mitigate risks associated with adverse weather and optimise their travel routes accordingly, thereby enhancing overall safety and efficiency.

In summary, popularity-based caching constitutes a foundational strategy within the IoV ecosystem, offering multifaceted benefits such as improved user experience, cost-effective infrastructure management, and rapid responsiveness to user demands. However, its effective implementation necessitates meticulous technical oversight and continuous adaptation to evolving user preferences and operational requirements.

3.3.2.2 Geographic Caching: Enhancing System Efficiency in the Internet of Vehicles (IoV)

Geographic caching represents a critical strategy designed to augment the efficiency of systems that demand rapid local responsiveness, notably within the context of the Internet of Vehicles (IoV). This advanced method leverages the anticipated geographic locations of users to pre-load essential data, thereby minimising latency and optimising resource utilisation. The following sections provide a comprehensive exploration of the operational mechanics, inherent benefits, and potential challenges associated with geographic caching, along with a detailed discussion of its practical applications in IoV systems [93].

3.3.2.2.1 Mechanics of Geographic Caching Geographic caching operates by storing data tailored to the anticipated geographic locations of users. This strategic approach employs predictive analytics to forecast routes and vehicle movements, thus facilitating the swift retrieval of pertinent data tailored to specific local contexts. By proactively identifying areas that users are likely to visit, the system efficiently pre-loads essential information, such as detailed maps and updates on local conditions, in advance of the user's arrival in these areas. This pre-emptive data management not only enhances the responsiveness of the system but also improves the overall user experience by ensuring that relevant data is immediately accessible when needed.

3.3.2.2.2 Benefits of geographic caching

- Reduced Latency

  Geographic caching substantially decreases the response time of navigation systems by ensuring that relevant data is pre-loaded locally. This capability is critical for real-time navigation applications, where the ability to provide prompt feedback based on user requests directly impacts system reliability and effectiveness.

- Bandwidth Optimisation

  By minimising the necessity for continuous data exchanges between the vehicle and cloud services, geographic caching effectively reduces bandwidth usage. This leads to a significant enhancement in the overall efficiency of the system, alleviating network congestion and facilitating a more sustainable use of networking resources.

- Enhanced User Experience

  The implementation of geographic caching leads to markedly faster response times and more fluid interactions with navigation systems. These improvements contribute to a significantly enhanced driving experience, increasing

user satisfaction by providing timely and accurate navigational assistance without delays.

3.3.2.2.3 Challenges and Solutions in Geographic Caching

- Challenge: Accuracy of Predictions

  The efficacy of geographic caching hinges significantly on the precision of the predictive algorithms that forecast future vehicle movements. Inaccurate predictions can result in caching errors, leading to data misalignment and suboptimal system performance. This issue underscores the importance of algorithmic accuracy in pre-emptive data loading strategies.

- Solution: Enhancing Prediction Algorithms

  To address these challenges, there is a continuous need to enhance predictive algorithms through sophisticated machine learning techniques and the integration of both historical and real-time data. By refining these algorithms, the system can achieve a more accurate understanding of vehicle movement patterns, thereby increasing the reliability of the caching process. Ongoing adjustments and learning processes ensure that the system adapts to changing conditions and improves its predictive capabilities over time.

3.3.2.2.4 Specific Applications of Geographic Caching in the Internet of Vehicles

- Navigation and Traffic Management:

  Geographic caching significantly enhances the functionality of navigation systems by pre-loading route information in regions prone to high traffic or along corridors designated for major events. This proactive data management strategy ensures that navigation systems can deliver route guidance with greater accuracy and reduced latency, thereby facilitating smoother traffic flow and enhanced route planning.

- Emergency Response:

  In the realm of emergency services, the ability to access detailed, up-to-date geographic information rapidly is crucial. Geographic caching supports this need by ensuring that maps and route information for potential response areas are readily available, which is particularly critical when addressing urgent situations such as accidents or natural disasters. This readiness can drastically cut response times and improve the effectiveness of emergency operations.

In conclusion, geographic caching in IoV plays a key role in improving the performance of navigation and emergency response systems, offering tangible benefits in terms of responsiveness and efficiency [94]. However, its effectiveness depends on the ability to accurately anticipate local data needs, which requires an advanced technological infrastructure and robust predictive algorithms.

3.3.2.3 Collaborative Caching in the Internet of Vehicles : A Strategic Overview

Collaborative caching within the Internet of Vehicles (IoV) represents an advanced strategy that leverages inter-vehicle communication to optimise data distribution and access [95]. This method enables vehicles to function both as data consumers and providers, sharing cached information such as traffic conditions, navigation updates, and safety alerts directly with one another. This process not only reduces reliance on centralised servers, thereby decreasing latency and bandwidth consumption, but also enhances data availability and system resilience [96].

3.3.2.3.1 Mechanisms and Benefits

- Increased Efficiency

  Collaborative caching enables direct data sharing among vehicles, substantially reducing dependency on centralised servers for information retrieval.

This strategy significantly diminishes both latency and bandwidth consumption, as data does not need to traverse the conventional network infrastructure to reach its destination. This direct exchange facilitates quicker responses and more efficient data usage, which is crucial for real-time applications within the IoV.

- Improved Reliability
  By dispersing cached data across a network of vehicles, this approach introduces a high level of redundancy that enhances the overall reliability of the system. In the event of a network node failure, the distributed nature of the data ensures that information remains accessible, thereby preventing a single point of failure from compromising the network's operational integrity.

- Adaptability to Dynamic Conditions
  Collaborative caching exhibits exceptional adaptability in dynamic environments, such as during major public events or within highly variable traffic conditions. This flexibility is paramount as it allows the system to rapidly update and disseminate new information, ensuring that data remains relevant and up-to-date. The ability to quickly adapt to changing circumstances makes collaborative caching particularly valuable in scenarios where timely information is critical to safety and efficiency.

3.3.2.3.2 Challenges and solutions

- Managing confidentiality and security: Sharing data between vehicles raises significant issues of confidentiality and data security
  - Solution: Implementation of robust security protocols and encryption techniques to protect the data exchanged.

- Data synchronisation and consistency: To ensure that the data shared is up to date and consistent between all participating vehicles.

- Solution: Utilise consensus mechanisms or periodic verification to validate the accuracy and freshness of the data.

3.3.2.3.3 Specific Applications

- Convoys of Vehicles:
  - Real-time Updates: Implement a collaborative caching system that utilises V2V communication protocols. This system facilitates the exchange of real-time updates on road conditions amongst vehicles within a convoy.
  - Dynamic Routing: Integrate intelligent routing algorithms that utilise cached data to dynamically reroute convoy vehicles in response to traffic congestion, accidents, or other road hazards.
  - Edge Computing: Utilise edge computing nodes installed along the convoy route to store and distribute cached data, reducing latency and enhancing the responsiveness of the system.
  - Security Measures: Implement robust encryption and authentication mechanisms to ensure the integrity and privacy of cached data shared amongst convoy vehicles.
- Emergency Response:
  - Priority Communication Channels: Establish dedicated communication channels for emergency response vehicles to ensure rapid and reliable transmission of critical information.
  - Data Prioritisation: Implement prioritisation schemes within the collaborative caching system to ensure that essential data related to emergency response, such as route information and situational updates, receive precedence.
  - Integration with Command Centres: Integrate the collaborative caching system with centralised command centres to facilitate seamless coordination between response vehicles and emergency management personnel.

- RResilient Networking: Deploy redundant communication technologies, such as cellular networks and satellite links, to maintain connectivity in areas with limited infrastructure or during network congestion.

In Conclusion, Optimising collaborative caching for convoy management and emergency response in the IoV involves integrating advanced technical solutions tailored to the specific requirements of each application. By addressing challenges such as security, data management, and real-time communication, these optimised systems can significantly enhance the efficiency and reliability of intelligent transport systems.

3.3.2.4 Proactive Caching in the IoV: An Extensive Examination

Proactive Caching: Proactive caching represents an innovative strategy within the Internet of Vehicles (IoV) framework, aiming to anticipate user requirements by preemptively loading data based on predictive behavioural patterns and scheduled events. This sophisticated approach leverages predictive analytics to analyse historical user interactions and forecast future needs, enabling the system to offer highly responsive and personalised services[97]. Below, we delve into a comprehensive analysis of its operational mechanisms and the significant benefits it offers:

3.3.2.4.1 Functionality of Proactive Caching

Proactive caching harnesses sophisticated algorithms to predict forthcoming user demands. Through the analysis of historical data and real-time trend monitoring, the system discerns recurring patterns in user behaviour and identifies scheduled events expected to influence data needs. For instance, anticipating a significant event like a concert or sports match, the system proactively loads alternative routes and pertinent traffic data to mitigate anticipated congestion. This proactive approach ensures timely access to relevant information, thereby optimising user experiences within the IoV framework.

3.3.2.4.2 Benefits of Proactive Caching

- Enhanced Responsiveness: Proactive caching anticipates user needs, enabling swift responses to requests and fostering a seamless, personalised user experience. By preloading relevant data, the system minimises latency and ensures timely access to information within the Internet of Vehicles ecosystem.

- Resource Optimisation: By strategically preloading data, proactive caching reduces reliance on real-time data transfers, alleviating network congestion and optimising resource utilisation. This minimises the load on networks and servers, enhancing overall system efficiency and performance.

- Improved Predictability: Proactive caching enhances the predictability of driving conditions by providing relevant information in advance. By preemptively delivering data before it is explicitly requested, the system empowers drivers to make informed decisions and optimise their journeys, contributing to safer and more efficient travel experiences.

- Traffic and Congestion Management: Proactive caching plays a pivotal role in predicting and mitigating anticipated congestion scenarios. By leveraging predictive analytics, proactive caching systems can forecast traffic patterns and identify potential bottlenecks. This enables the timely provision of alternative routes and real-time traffic updates to drivers, promoting smoother traffic flow and reducing overall congestion levels. Additionally, proactive caching enhances the efficiency of traffic management systems by preemptively distributing relevant data, thereby facilitating more proactive decision-making and adaptive traffic control strategies.

- Major Event Planning: Proactive caching is instrumental in facilitating the efficient management of traffic flow during major events such as concerts, sporting events, or festivals. By preemptively preloading relevant information, including event venue locations, parking availability, and recommended routes, proactive caching systems assist participants and event attendees in

navigating the event area with minimal disruption. This proactive approach not only improves the overall mobility of attendees but also enhances public safety and reduces the environmental impact of traffic congestion associated with large-scale events. Moreover, proactive caching contributes to the seamless coordination of event logistics and emergency response efforts, ensuring optimal resource allocation and operational readiness during critical periods.

In summary, proactive caching represents a substantial advancement in optimising the performance of intelligent transport systems within the IoV. By proactively anticipating user needs and preloading pertinent data, it not only enhances system responsiveness but also optimises resource utilisation, fostering a more efficient and personalised user experience. The widespread adoption of proactive caching holds the promise of fundamentally reshaping the design and utilisation of future intelligent transport systems. Its integration promises to revolutionise the efficiency, reliability, and adaptability of IoV ecosystems, paving the way for safer, smoother, and more sustainable transportation networks.

3.3.2.5 Content-Based Caching in the IoV: An Elaborate Strategy

Content-based caching represents a tailored approach focused on anticipating and caching specific user-requested content within the IoV ecosystem. This strategic framework is designed to proactively identify and cache multimedia resources such as videos, music, or other frequently accessed data, offering significant advantages across diverse contexts, particularly in the realm of in-car entertainment services[98]. Here, we delve into a comprehensive examination of its operational intricacies and the manifold benefits it confers:

3.3.2.5.1 Functionality of Content-Based Caching

Content-based caching works by identifying and preloading specific content items at the user's request[99]. This strategy uses sophisticated algorithms to analyse content consumption patterns and user preferences to determine the most relevant media resources to cache. For example, in the context of in-vehicle entertainment

services, content-based caching can anticipate which videos, songs, or applications are most popular among passengers, preloading these elements for a smooth and uninterrupted entertainment experience.

3.3.2.5.2 Benefits of Content-Based Caching

- Personalisation and Usability: By specifically targeting user-preferred content, content-based caching delivers a more personalised and user-friendly experience. By preloading relevant media assets, it ensures that users have access to their preferred content without delay, enhancing their overall satisfaction.

- Bandwidth Optimisation: By reducing the need to download content in real time, content-based caching helps to optimise bandwidth usage. This minimises network congestion and improves overall system performance, particularly in environments where connectivity may be limited or intermittent.

- Reduced Latency: By preloading specific pieces of content, content-based caching reduces the latency associated with downloading media assets, providing a smoother, more responsive entertainment experience for users on the move.

3.3.2.5.3 Applications of Content-Based Caching

- In-Car Entertainment: One of the most direct applications of content-based caching is in in-car entertainment systems. By preloading popular videos, music, and podcasts based on historical and real-time user preference data, vehicles can offer a seamless entertainment experience that minimises loading times and disruptions due to poor connectivity.

- Real-Time Traffic and Navigation Updates: Content-based caching can be applied to navigation systems within the IoV. By caching frequently used routes and real-time traffic updates, the system can provide quicker access

to navigation information, enhancing the driving experience and improving route efficiency.

- Software and Firmware Updates: Automobiles today receive regular software updates that can include critical security patches and functionality improvements. Content-based caching can predict when a vehicle is likely to be in range of reliable networks and automatically download updates in advance, ensuring that updates can be applied more quickly and at a convenient time without requiring the vehicle to be connected to a network.

- Emergency and Safety Messages: For safety features, such as collision avoidance systems and emergency notifications, it is vital that there is no delay in message delivery. Content-based caching can help by pre-storing critical safety information and updates locally in the vehicle, thus ensuring instant access when needed.

- Advertising and Promotional Content: The IoV also opens up possibilities for targeted advertising, where content-based caching can preload ads based on user interests and past behaviour. This could enhance the user experience by delivering personalised advertising content that is relevant and timely, potentially while the vehicle is parked or charging.

- V2V and V2I Communications: In V2V and V2I communications, vehicles exchange information with each other and with road infrastructure to enable smarter, safer driving decisions. Content-based caching can help by storing frequently exchanged data types, such as local hazard notifications or construction updates, making them available immediately as needed.

- Augmented Reality (AR) Applications: In the future, augmented reality applications could benefit from content-based caching by preloading location-based media content, such as tourist information or historical data about a location, enhancing the informational value and interactivity of AR features during a drive.

Overall, content-based caching represents a crucial evolution in IoV technology, focusing on user experience, network efficiency, and the effective management of the increasingly data-intensive environment of modern vehicles.

The table 3.2 provides a comprehensive comparative analysis of key caching strategies used in the Internet of Vehicles (IoV). These strategies include Popularity-Based, Geographic, Collaborative, Proactive, and Content-Based Caching. Each caching method is meticulously evaluated for its functionality, advantages, and applications, providing detailed insights into their effectiveness across various IoV scenarios. By examining these strategies, the table highlights how different caching techniques can significantly enhance data retrieval speed, reduce latency, optimise bandwidth usage, and improve overall network efficiency. This summary aids in understanding the strategic impact of each caching method on the efficiency, performance, and reliability of IoV systems, facilitating the development of more robust and responsive vehicular networks.

In summary, caching is a cornerstone of the IoV, crucial for boosting performance, minimising latency, and enhancing the reliability of intelligent transportation systems. Strategically devised caching techniques are vital, tailored to accommodate the varied needs of connected vehicle applications. These strategies ensure an optimised user experience and swift responses in critical situations, thereby reinforcing the effectiveness and dependability of IoV frameworks.

Table 3.2: Comparative Overview of Caching Strategies in the IoV

Caching Strategy	**Functionality**	**Benefits**	**Applications**
Popularity-Based Caching	Monitors user requests to detect demand patterns and caches data based on popularity metrics.	Reduced latency and server load. Efficient data retrieval.	Navigation and traffic management, Weather forecasting.
Geographic Caching	Predicts user routes and pre-loads data for specific geographic areas.	Reduced latency, Bandwidth optimisation, Enhanced user experience.	Enhanced navigation and traffic management, Emergency response.
Collaborative Caching	Vehicles share cached data directly with each other, reducing reliance on centralised servers.	Increased efficiency and reliability, Adapts to dynamic conditions.	Convoys of vehicles, Emergency response.
Proactive Caching	Uses predictive analytics to preload data based on anticipated needs, analyses historical and real-time data.	Enhanced responsiveness, Resource optimisation, Improved predictability.	Traffic and congestion management, Major event planning.
Content-Based Caching	Identifies and pre-loads content based on user preferences and consumption patterns, targets specific media resources.	Personalisation and usability, Bandwidth optimisation, Reduced latency.	In-car entertainment, Real-time traffic and navigation updates.

3.4 Reinforcement Learning (RL) in IoV

3.4.1 Introduction to Artificial Intelligence in Vehicle Networks

Artificial intelligence is increasingly used in vehicular networks to improve safety, traffic management, and user services. Vehicular networks, such as VANETs (Vehicular Ad Hoc Networks), enable vehicles to communicate with each other (V2V) and with the road infrastructure (V2I) to exchange crucial information in real time. This exchange of information facilitates the ubiquitous connectivity of vehicles and their interaction with the environment, contributing to applications such as traffic management, infrastructure maintenance, air quality monitoring, and many other services linked to mobility and intelligent transport.

AI is also used to optimise routing in vehicular networks. For example, approaches based on deep reinforcement learning are employed to generate optimal routing strategies according to the variable distribution of traffic. These techniques enable efficient and adaptive routing decisions, taking into account the real-time conditions of the road network.

In addition, the integration of AI into vehicular networks brings with it challenges and opportunities, particularly regarding security and the protection of user privacy. Mechanisms are being implemented to guarantee the anonymity of transmitting and receiving vehicles, as well as the confidentiality of personal data exchanged on these networks.

In conclusion, artificial intelligence plays a crucial role in the evolution of vehicular networks, enabling advanced functionalities, efficient traffic management, and an overall improvement in the driving and mobility experience.

Table 3.3 presents various artificial intelligence approaches employed in vehicular networks, accompanied by succinct descriptions of each method.

Table 3.3: Description of Artificial Intelligence Approaches used in Vehicle Networks

AI approach	Description
Supervised learning	Use of historical labelled data to train predictive models. For example, predicting future traffic from past data on road congestion.
Unsupervised learning	Identification of patterns and structures in label-free data. Useful for detecting anomalies and malicious behaviour in vehicular communications.
Reinforcement Learning	Reward-based learning to optimise routing and traffic management decisions. Vehicles learn from their interactions with the road environment.
Deep Neural Networks	Use of neural networks with many hidden layers for complex tasks such as obstacle detection and traffic sign recognition, etc.

These different AI approaches can significantly improve the performance and safety of vehicle networks by optimising data routing, traffic management, incident detection, and real-time decision-making. They contribute to the development of intelligent transport systems (ITS) by making mobility more efficient and safer.

3.4.2 Reinforcement Learning

Reinforcement Learning is a branch of artificial intelligence that focuses on training agents to make optimal decisions through continuous interactions with their environment[100]. Unlike traditional supervised learning methods that require labelled datasets, reinforcement learning uses a system of rewards and punishments to guide the agent towards achieving its goals. Each action taken by the agent in a given state leads to a reward or punishment, and the agent gradually learns to maximise cumulative rewards in the long run. This approach is particularly useful in dynamic and complex environments such as vehicle networks, logistics management, games, and robotics, where decisions must be made in real time and

the consequences of actions are not always immediately apparent. Reinforcement learning enables the development of autonomous systems capable of adaptation and continuous optimisation in diverse and constantly evolving contexts[101].

3.4.3 RL Model

A Reinforcement Learning (RL) model is structured around several essential components: the agent, the environment, states, actions, and rewards.

- Agent: The agent is the entity that learns and makes decisions. It can be a robot, an autonomous vehicle, software, or any other system capable of interacting with its environment. The agent observes the current state of the environment and chooses actions based on a learned policy.

- Environment: The environment encompasses everything that surrounds the agent and with which it interacts. This includes other agents, objects, obstacles, and changing conditions. The environment provides observations or states to the agent and reacts to the agent's actions by changing state and providing rewards.

- States: States represent the current conditions in the environment. A state can contain a variety of information, such as the position of a robot, the speed of a vehicle, or the current configuration of a game. States help the agent understand the context in which it finds itself.

- Actions: Actions are the decisions the agent can take from a given state. Each action chosen by the agent modifies the state of the environment. Actions can be discrete (e.g., moving a character to the left or right) or continuous (e.g., accelerating to a certain speed).

- Rewards: Rewards are feedback signals that the agent receives after performing an action. Rewards can be positive, negative, or zero. The agent's goal is to maximise the sum of rewards obtained over time. Rewards help the agent evaluate the quality of its actions.

Figure 3.1 illustrates the key components and processes of the Reinforcement Learning model. The model involves an agent that observes the state, chooses an action, acts on the environment, receives a reward, and learns. The environment also changes state.

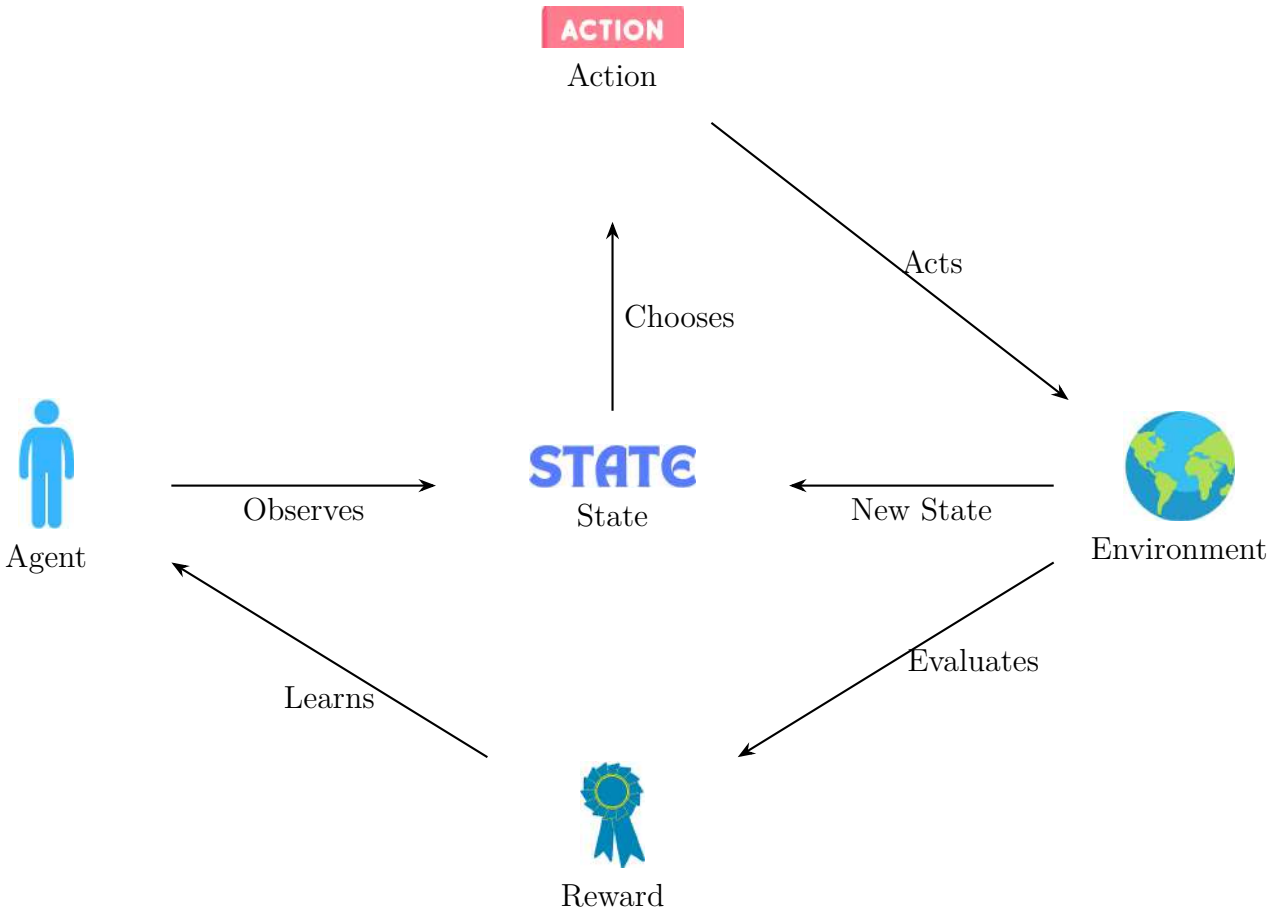


Figure 3.1: Reinforcement learning model

3.4.3.1 Model Operation

- Policy: The policy is the strategy the agent uses to decide which action to take in each state. It can be deterministic (a fixed action for each state) or stochastic (actions chosen according to a probability distribution).

- Value Function: The value function evaluates the quality of states or actions by estimating the sum of future rewards an agent can expect to receive from

that state or action. The value function helps the agent distinguish good states and actions from bad ones.

- Learning Process: The agent uses RL algorithms, such as Q-learning or Deep Q-Networks, to improve its policy by learning from past experience. It explores different actions to discover those that yield the best rewards and exploits the learned actions to maximise future rewards.

The ultimate goal of an RL model is to enable the agent to develop an optimal policy that maximises cumulative rewards over the long term. This continuous learning process allows the agent to adapt to changes in the environment and improve its performance over time.

3.4.4 Advancements in Caching Strategies Through Reinforcement Learning in IoV

Advances in caching strategies thanks to reinforcement learning are revolutionising the management of computing resources, particularly in the field of the Internet of Vehicles (IoV). Reinforcement learning, a branch of artificial intelligence, enables systems to make optimal decisions based on experience and reward. By applying these techniques to caching strategies, systems can learn to anticipate and proactively store the most requested data, thereby improving performance and reducing latency[102].

Reinforcement learning uses autonomous agents that interact with their environment to maximise a reward function. In the context of caching, these agents can analyse data access patterns and adapt caching policies in real time. For example, an agent can identify the most frequently used files or data and cache them before they are even requested, thus reducing access time for the end-user. This is particularly crucial in the field of IoV, where connected vehicles generate and consume large quantities of data in real time.

In the context of the IoV, agents can analyse data access patterns of connected vehicles and adapt caching policies in real time. For instance, a vehicle could

preload navigation or entertainment data before entering an area with limited network coverage, ensuring an uninterrupted user experience. Additionally, vehicles can share critical information, such as road conditions or traffic incidents, in real time, improving transport safety and efficiency.

This dynamic, adaptive approach optimises the use of storage and network resources. Unlike traditional caching methods, which often follow predefined, static rules, reinforcement learning enables greater flexibility and responsiveness. Systems can automatically adjust to variations in demand, improving overall efficiency.

Reinforcement learning algorithms can be continually enhanced through continuous learning techniques, enabling systems to remain efficient even as conditions and user behaviour change[103]. In the IoV, this means that systems can adapt to new trends and technologies, such as autonomous vehicles and 5G networks, offering more efficient and intelligent solutions for data management in complex, high-demand environments.

In conclusion, caching strategies based on reinforcement learning offer significant advantages for the IoV in terms of performance, responsiveness, and efficiency, contributing to an enhanced user experience and optimised resource management.

3.5 Gaps in existing research

3.5.1 Caching with the use of AI and ML

Network technology has progressed considerably with the integration of AI and ML in edge caching. These innovations offer two primary benefits: predictive caching and dynamic cache management. Predictive caching, driven by AI, allows edge devices to anticipate user demands, thereby reducing latency and enhancing the user experience. ML furthers this capability by enabling dynamic cache management, which adapts to changing network conditions in real-time. Together, AI and ML not only optimise data storage and access at the network's edge but also pave the way for more efficient and responsive network systems, which are crucial for 6G and autonomous applications. Several studies have employed a reinforcement learning

approach to improve caching algorithms.Reference [104] tackled the Device-to-Device (D2D) caching issue as a multi-agent Multi-Armed Bandit (MAB) learning problem. The study utilised Q-learning to identify the optimal coordination of caching decisions among multiple agents, thereby maximising caching benefits. Reference [105] employed a recurrent neural network (RNN) to predict user behaviour and content popularity. The proposed Q-learning strategy, which incorporates learning automata, facilitates cooperative caching by considering content popularity and the locations of mobile users, thus enabling optimal decision-making in a static environment. In [106], the authors examined user terminal (UT) edge caching within D2D-enabled cellular networks, focusing on content popularity and UT positioning. They modelled the problem as a stochastic game, resolving it through a multi-agent cooperative alternating Q-learning (CAQL) algorithm. This approach allows UTs to update their caching placement policies for improved performance. Reference [107] investigated the tidal effect in a mobile edge computing (MEC) network featuring multiple users and multicast data. The study formulated the problem as an infinite-horizon average cost Markov Decision Process, aiming to optimise bandwidth usage and minimise data transmission. By recasting the problem in a reinforcement learning framework, the study proposed Q-learning and a deep Q-network (DQN) to understand file popularity and user requests, offering valuable insights into network caching design. Previous studies have predominantly leveraged AI and ML technologies to optimise caching strategies at the edge or user end. However, these approaches have often neglected the specific preferences of individual users or groups, underscoring the necessity for a more holistic and effective application of AI and ML in caching optimisation.

3.5.2 Vehicle Caching

The caching issues arising from vehicle mobility are notably different from those encountered in cellular networks. Several studies have focused on vehicular caching. In reference [108], the authors investigated the concept of vehicles serving as mobile cache nodes, creating an automobile cloud to deliver requested content to users via

6G technology. To equip edge smart technologies for the upcoming 6G vehicular network, [109] proposed using parked vehicles as additional edge nodes, thereby supplementing existing ground infrastructure with abundant resources. The architecture outlined in [110] consists of three layers: an airship, UAVs, and vehicles, all utilised for vehicular caching. The authors applied the DQN method to determine the optimal caching strategy. Another study employed an advanced machine learning technique in Information-Centric Networking-based vehicular networks [111]. This method predicts future user requests by forecasting their evaluations of videos, making Information-Centric Networking highly adaptable to vehicular network environments.

- **Caching in autonomous driving**

  Deep Reinforcement Learning (DRL) and Federated Learning (FL)-based content caching methods have been proposed to enhance transmission efficiency while meeting latency requirements [112]. These methods optimise latency by accounting for regional preferences and constraints. Multi-Agent Reinforcement Learning (MARL) and FL were employed to forecast content popularity and decide on the cached content for each region.

  Another study proposed a deep learning system for content caching in autonomous vehicles, which utilises MEC servers to cache high-probability content and a Multi-Layer Perceptron (MLP) to predict content demands in specific regions [113]. A Convolutional Neural Network (CNN) was employed to predict passengers' age, emotion, and gender, while the self-driving car used binary classifications and k-means clustering to select and cache relevant infotainment content. In [114], researchers addressed challenges related to automated vehicle control and proactive caching in Roadside Units. This study utilised deep reinforcement learning to enhance efficiency and user quality of experience (QoE) by implementing proactive caches and determining proactive caching actions. The authors in [115] proposed caching

infotainment content for self-driving cars on MEC servers, leveraging CNN-derived passenger features to achieve high expected probabilistic values in their regions.

All the previously mentioned studies examined vehicles and edge devices separately, neglecting the potential advantages of integrated caching and regional preferences.

3.5.3 Edge Caching

Recent research has increasingly focused on edge caching. In [116], a learning-based cooperative content caching policy for MEC architecture is proposed, designed for scenarios where users' preferences are unknown and only historical content demands are visible. This study employs a MARL-based approach to address the cooperative content caching problem by modelling it as a multi-agent multi-armed bandit problem. In [117], the authors introduced an innovative edge-assisted intelligent caching framework that improves cache hit rates. This framework autonomously develops a caching strategy in real-time by analysing request sequences without the need for preliminary data processing or feature engineering. Another study explored the prediction of spatial content preferences using distributed learning techniques and mobility predictions [118]. Unlike most other studies, [119] applied deep reinforcement learning to collaborative caching and addressed content of varying sizes. In [120], a collaborative caching approach for mobile edge computing servers integrated multi-agent reinforcement learning.

These studies generally assume that users remain static. However, in the context of autonomous driving, the high-speed mobility of vehicles leads to frequent changes in network topology, complicating the application of these methods. Therefore, it is essential to enable local caching for vehicle users and investigate optimal content placement strategies to mitigate the impact of vehicle movement on users' Quality of Experience (QoE).

3.5.4 Combining Vehicle and Edge Caching

Recent studies have explored the integration of local vehicle and edge device caches, leveraging user and regional preferences to optimise vehicle usage. The architecture proposed in [121] introduced a collaborative caching approach in the IoV, using content request prediction (CCCRP) to reduce latency in accessing content. The authors employed K-means clustering to group vehicles, LSTM networks to predict content requests, and reinforcement learning to enhance caching decisions, thus improving the quality of service for vehicle requests. Similarly, the study [122] presented a cooperative caching strategy for content downloading using reinforcement learning, incorporating K-means clustering for vehicle grouping, long short-term memory networks for content prediction, and the DRL algorithm alongside DQN to determine the most efficient caching techniques.

However, both studies do not effectively integrate vehicle caches with those of base stations (BS) and roadside units (RSU). A multi-level cache integration could significantly reduce redundancy and enhance overall system efficiency. Moreover, these studies overlook the optimisation of caching strategies based on regional preferences. Better understanding and utilisation of local preferences could improve content relevance and further reduce latency.

Table 3.4 summarises the techniques, Benefits, and limitations of AI-enhanced caching strategies in the Internet of Vehicles (IoV).

Table 3.4: Strategic Analysis of AI-Enhanced Caching in IoV: Techniques, Benefits, and Limitations

Ref	AI/ML Technique	Focus Area	Primary Benefit	Limitations
[104]	Q-learning (MAB)	D2D Caching	Optimal coordination of caching decisions among multiple agents	Complexity of modelling the problem as a multi-agent multi-arm bandit problem; challenges in designing an efficient multi-agent reinforcement learning algorithm to coordinate caching.
[105]	RNN + Q-learning	Cooperative Edge Caching	Adapts to user behaviour and content popularity	Technique complexity due to neural networks complicates implementation; data dependence necessitating accurate historical data; computational demands limit real-time use; challenges in scalability and adaptability; potential convergence to locally optimal solutions only.
[106]	CAQL (Q-learning)	UT Edge Caching	Dynamically updates caching placement policies	Complex implementation, particularly in large-scale dynamic cellular networks; reliance on precise historical data; resource intensity limiting real-time application; scalability and adaptability issues in dynamic environments.

[107]	Q-learning, DQN	MEC Network Caching	Optimises bandwidth usage and reduces data transmission	Challenges in accurately predicting future user demands; constrained by computational and memory limitations; errors introduced by approximations; limited real-world applicability based on simulation parameters.
[108]	Markov Model, GBDT	Mobile Edge Caching for 6G	Efficient content allocation with low delay	Limited by short communication ranges and high mobility challenges.
[109]	`Conv_LSTM`	Extensive EI in Vehicular Networks	High accuracy in resource prediction, reduced deployment costs	Complexity in managing extensive and dynamic edge resources.
[110]	DQN	Vehicular Caching Layers	Optimises caching approach in a multi-layered vehicle system	Dependency on accurate network modelling.
[111]	Machine Learning	ICN-based Vehicular Network	Predicts future user requests	Potential inaccuracies in long-term predictions.
[112]	DRL, FL	Autonomous Driving	Efficient transmission with regional preferences	FL requires careful data privacy management.

[113]	MLP, CNN	Autonomous Vehicles	Predicts content demand and personalises caching	Heavy reliance on accurate demographic predictions.
[114]	Deep Reinforcement Learning	Proactive Caching in Vehicles	Improves efficiency and QoE	High computational requirements for real-time decisions.
[115]	CNN	Infotainment Content Caching	Utilises learned passenger features for caching	May not generalise well across diverse passenger profiles.
[116]	MARL, MAB	Edge Caching	Addresses unknown user preferences	Challenging to implement in highly dynamic environments.
[117]	Intelligent Caching	Edge Caching	Improves cache hit rate without preliminary data processing	May struggle with non-standard request patterns.
[118]	Distributed Learning	Edge Caching	Predicts spatial content preferences	Difficulties in scalability and real-time updates.
[119]	Deep Reinforcement Learning	Collaborative Edge Caching	Manages collaborative caching effectively	Potentially high overhead in cooperative setups.
[120]	MARL	Collaborative Caching	Enhances caching in mobile edge computing servers	Coordination complexity among multiple agents.

[121]	K-means, LSTM, RL	IoV Caching	Minimises latency and enhances QoS for vehicle requests	Simplified assumptions about vehicle movement.
[122]	K-means, LSTM, DRL, DQN	IoV Caching	Effective caching through cooperative approaches	Complexity and resource demands of multiple AI techniques.

This table presents a strategic analysis of AI-enhanced caching techniques in the Internet of Vehicles (IoV), highlighting the main Benefits and limitations of each technique.

3.6 Conclusion

This chapter has conducted an in-depth examination of caching management and reinforcement learning within the Internet of Vehicles (IoV), illustrating how these technologies can substantially enhance the performance of vehicular networks. Through a detailed analysis of caching methods and reinforcement learning models, we have identified strategies capable of dynamically responding to the fluctuating demands of IoV networks, whilst optimising the efficiency of intelligent transport systems.

The caching techniques explored provide a significant reduction in latency and more efficient bandwidth management, which are essential for real-time IoV applications. The integration of reinforcement learning enables these techniques to proactively adapt, anticipating data needs and adjusting resources accordingly, thus enhancing the network's capability to manage ever-changing requirements effectively.

The implementation of advanced caching strategies, coupled with reinforcement learning, shows considerable potential for improving traffic flow, vehicle safety, and energy efficiency. These enhancements are crucial for the future development of autonomous and connected transport systems, providing a robust platform for continued innovation in the transport sector.

In conclusion, this chapter has highlighted the importance of advanced cache management and reinforcement learning in enhancing the performance of IoV networks. The strategies and models presented provide a solid foundation for future research and practical applications in the field of intelligent transportation. The theoretical and practical contributions of this chapter are essential for realising the full potential of IoV and MEC technologies, paving the way for more innovative and efficient transportation systems.

Chapter 4

Advanced Data-Driven Caching Strategy

Contents

4.1 Introduction

Caching is an essential computer strategy that enhances the performance and efficiency of a system by storing frequently accessed data in quick-access memory. This approach reduces data access time and transmission latency, thereby improving the data processing rate and significantly diminishing the time required to retrieve data for applications demanding real-time and high-performance capabilities. Caching manages recurrent data requests, minimising the workload on the main system and enabling effective service scalability during periods of high traffic. This enhances user experience by accelerating data recovery, particularly beneficial for web development and rapid website loading. Economically, it lowers the costs associated with data transfer and processing and facilitates access to offline data, thus reducing bandwidth usage and network congestion. Moreover, caching boosts system scalability by eliminating data access bottlenecks.

4.2 Data Collection

4.2.1 Sources of data

The data from the Big Data Challenge, utilised for evaluating caching strategies, comprises a comprehensive and detailed collection of user requests for various types of content over a span of two months. Specifically, the dataset records user interactions with ten distinct types of content, offering a diverse range of behaviours and preferences for analysis. Each entry in this dataset represents a snapshot of requests made within a ten-minute interval, meticulously documenting the dynamics of content demand within a tightly controlled timeframe[123].

The dataset's structure, encompassing 1,000 individual data points, is methodically divided into two segments to facilitate a robust training and validation process for the predictive models under evaluation. The initial segment, consisting of the first 800 data points, is designated as the training set. This portion is employed to develop and refine the predictive capabilities of the LSTM model, ensuring that it accurately learns the patterns and trends inherent in the content request data. The model's ability to generalise and effectively predict future behaviour is then rigorously assessed using the remaining 200 data points, serving as the test set.

This division allows for a clear evaluation of the model's performance, distinguishing between its capacity to recall learned information and to apply this knowledge to new, previously unseen data. This approach is crucial for validating the effectiveness of the proposed caching strategies, as it simulates a realistic scenario in which a model must operate effectively in dynamic and changing conditions typical of real-world environments.

By employing this structured and phased approach to data utilisation, the study aims to ensure that the enhancements in caching strategies driven by the model are both reliable and applicable to actual operational environments, thus paving the way for more efficient content delivery networks and optimised resource usage in vehicular and other mobile networks.

4.2.2 Data analysis techniques

Data analysis techniques encompass the use of Recurrent Neural Networks (RNN) and LSTM models to predict content popularity. Clustering methods, such as the K-means algorithm, are employed to categorise vehicles according to their mobility patterns. Reinforcement learning algorithms, such as Thompson Sampling (TS), are utilised to optimise caching decisions based on predictions of content popularity and the mobility characteristics of vehicles.

4.2.2.1 LSTM for predicting content requests

Our proposed design architecture highlights the significance of predicting content popularity, with the caching technique reliant on this popularity. Many previous studies assume that content popularity is pre-established, using request frequency as the assessment criterion. However, in practical scenarios, content popularity must include an element of freshness. Vehicle content requests have shown a consistent trend over time [124]. By analysing historical request data, it is possible to identify the fundamental trends that influence the frequency of content requests. These trends can then be utilised to forecast the volume of requests in future periods. In this context, the forecasted request volume is defined as the content's popularity.

4.2.2.1.1 Machine Learning Techniques

Leveraging machine learning techniques like Recurrent Neural Networks (RNNs) allows us to predict the popularity of newly generated content. RNNs can identify underlying temporal patterns by analysing historical request times for various content types. However, conventional RNNs often encounter issues such as vanishing or exploding gradients during extended propagation. To mitigate these problems, Long Short-Term Memory (LSTM) networks are favoured as they represent an enhanced version of RNNs. LSTM networks incorporate memory units and multiple gates, effectively overcoming these challenges and improving predictive accuracy.

- **LSTM Structure**

– Components:

The structure of the LSTM unit is shown in Fig. 4.1. Let μ_t and h_t denote the input and output data at time step t, respectively. At each time step t, the cell receives a new input μ. The cell contains three gates: the input gate i, the forget gate f, and the output gate o. The values of these gates are calculated as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, \mu_t] + b_f) \tag{4.1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, \mu_t] + b_i) \tag{4.2}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, \mu_t] + b_o) \tag{4.3}$$

Here, W_f, W_i, and W_o denote the weight matrices, while b_f, b_i, and b_o represent the biases for the three gates. The function $\sigma(\cdot)$ refers to the nonlinear sigmoid activation function.

– Updating Information:

The process for updating information in an LSTM is detailed as follows [125]:

* The forget gate f_t decides which part of C_{t-1} should be discarded.

$$f_t \cdot C_{t-1} \tag{4.4}$$

* The current state information is updated

$$i_t \cdot \tilde{C}_t \tag{4.5}$$

Where

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, \mu_t] + b_c) \tag{4.6}$$

Here, b_c and W_c represent the bias and weight matrix of the memory cell, respectively, and $\tanh(\cdot)$ is the hyperbolic tangent function.

∗ The current unit is updated as follows:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{4.7}$$

– Output Data Calculation:

As a result, the output data is computed as follows:

$$h_t = o_t \cdot \tanh(C_t) \tag{4.8}$$

- **Application of LSTM**

LSTM can forecast future content request volumes, allowing for optimised caching for vehicle users. By utilising time series data, the LSTM model takes the number of content requests from the previous time step $t-1$ as its input, denoted by

$$\mu(t-1) = \{\mu_1(t-1), \mu_2(t-1), \ldots, \mu_Q(t-1)\} \tag{4.9}$$

The LSTM output forecasts the number of content requests for the forthcoming time period as a series of values, represented by

$$\theta(t) = \theta_1(t), \theta_2(t), \ldots, \theta_Q(t). \tag{4.10}$$

- **Predicting Content Popularity**

The elements within the function $\theta(t)$ are sorted to create an ordered index vector:

$$\lambda = \lambda_1, \lambda_2, \ldots, \lambda_Q \tag{4.11}$$

This ordered index vector is then input into the Zipf model to determine the popularity of contents in the repository, as described in the caching model, producing a popularity distribution:

$$\pi = \pi_1, \pi_2, \ldots, \pi_Q \tag{4.12}$$

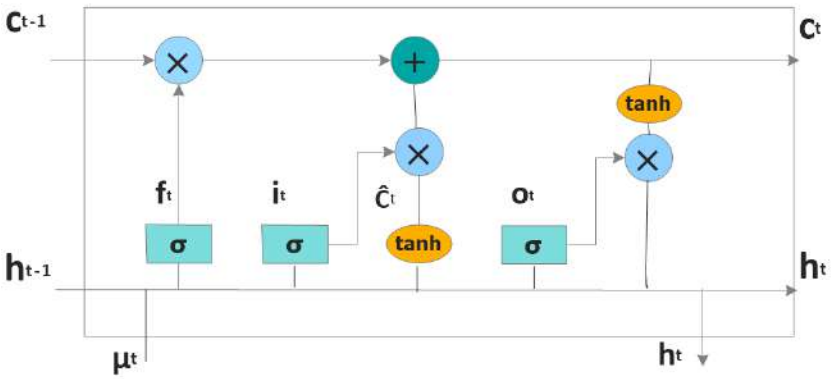


Figure 4.1: LSTM cell

4.2.2.1.2 Detailed Architectural Description of Our LSTM-Based Prediction Model

Our LSTM-based prediction model for forecasting content request volumes is organised as follows:

- **Input Layer:**
  - Inputs historical data of content request numbers over time, processed and normalized.
- **LSTM Layers:**
  - **First LSTM Layer:**
    - Consists of 128 units (number of memory cells in the layer) with `return_sequences=True` to maintain temporal sequence processing for deeper layers.
    - Activation function: 'ReLu'(Rectified Linear Unit) for intermediate layers to introduce non-linearity.
    - Includes a dropout rate of 0.2 to mitigate overfitting by randomly omitting a subset of features during training.
  - **Second LSTM Layer:**

* Comprises 64 units configured with `return_sequences=True`, This layer takes the sequence output from the previous LSTM layer and processes it further, outputting a sequence of 64-dimensional hidden states.
* Activation function: 'ReLu'.
* This layer also includes a dropout of 0.2.

– **Third LSTM Layer:**

* Number of units: 32 units
* Activation Function: ReLu
* Return Sequences: False (to return the final hidden state for the next layer)
* Dropout: 0.2

This layer takes the sequence output from the previous LSTM layer and processes it to output a 32-dimensional hidden state representing the entire sequence.

- **Dense Layer (Output Layer):**

  – Features a single neuron with a linear activation function to predict the count of future content requests, reflecting the anticipated popularity.

4.2.2.1.3 Model Compilation

- **Optimizer:**

  – The Adam optimizer is utilized for its efficient computation and adaptive learning rate, enhancing the convergence of training.

  – Learning Rate: 0.001

- **Loss Function:**

  - Mean Squared Error (MSE) is employed to quantify the accuracy of predictions, providing a clear measure of prediction error in the context of regression.

- **Metrics:**

  - Mean Absolute Error (MAE) provides a measure of the mean absolute difference between observed values and predictions, offering a more direct interpretation than the MSE.

4.2.2.1.4 Training Configuration

- **Epochs and Batch Size:**

  - The model is trained over 100 epochs with a batch size of 32, balancing the model's exposure to the training data against computational efficiency.

- **Early Stopping:**

  - Implemented with a patience of 10 epochs.
  - Training is stopped early if the validation loss does not improve for 10 consecutive epochs, preventing overfitting.

4.2.2.1.5 Model Training

- **Validation Split:** 20% of the data is used for validating the model performance during training, which helps in tuning and preventing overfitting.

- **Callbacks:** Early Stopping

  - Early stopping is used to halt training when the model's performance on the validation set stops improving

4.2.2.1.6 Evaluation of Our LSTM-based content prediction model

The dataset used for predicting content requests originates from the Big Data Challenge [123]. It includes data for ten types of content, collected from November 1, 2013, to January 1, 2014, with each data point representing a 10-minute interval. This analysis is based on a dataset of 1,000 such data points.

The first 800 data points are utilized as the training set, and the model's performance is subsequently tested on the last 200 data points. Once trained, the model's performance is evaluated on the validation set. The fig4.2 illustrating the training and validation losses over epochs reveals several insights:

- **Sharp Decline in Initial Epochs:**There is a significant drop in both training and validation losses within the first few epochs, indicating that the model quickly captures the dominant patterns in the data.

- **Convergence:** After the initial sharp decline, both losses level off, showing only slight further reductions. This trend suggests that continuing training beyond these epochs might not yield substantial improvements, pointing towards potential diminishing returns.

- **Closeness of Losses:** The training and validation losses remain close throughout the process, suggesting that the model generalizes well and is not overfitting. This closeness is indicative of a model that performs well on both seen and unseen data.

This comprehensive evaluation underscores the LSTM-based content prediction model's ability to learn and predict content request volumes effectively, showcasing its utility in practical scenarios, as evidenced by the dataset from the Big Data Challenge. The analysis of the loss plot further confirms the model's capacity to generalise, making it a reliable tool for forecasting content requests in similar settings. This affirms the model's robustness and adaptability, crucial for applications within dynamic environments.

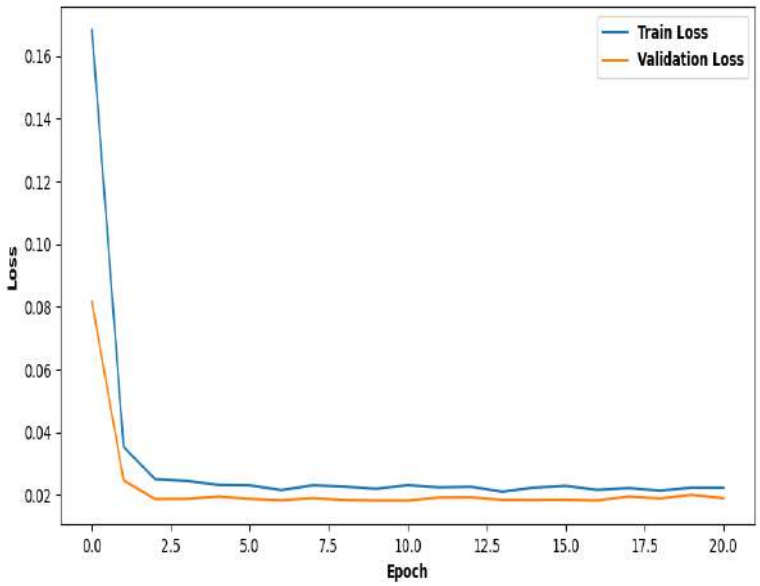


Figure 4.2: Model loss over epoch

4.2.2.1.7 Implementation in Content Caching

Integrating LSTM predictions into our caching strategy allows for dynamic adjustments to cached content based on real-time popularity forecasts. This method not only increases the efficiency of the caching system but also ensures that users have prompt access to the most relevant and in-demand content, thereby reducing latency and enhancing user experience. Our advanced LSTM framework underpins this predictive model, providing robust and accurate forecasts of content popularity, which is essential for the effective management of cache resources in vehicular networks.

4.2.2.2 K-means Clustering for Vehicle Data

Our approach involves clustering vehicles based on their mobility to reduce signalling load from V2V broadcasting and enhance vehicular communication connections. Within each cluster, the cluster head communicates directly with other members using single-hop V2V communication, ensuring efficient and direct communication pathways. This optimises the network's overall functionality and reduces the need for complex multi-hop communication strategies. Vehicles are classified

into distinct clusters within a single BS coverage area according to their mobility characteristics.

We formed Mc clusters from a set of Nv generated vehicles by using normalised attributes such as position and speed, obtained from each vehicle. The primary challenges in vehicle clustering are determining the clustering technique and selecting the cluster head. This study employs the k-means algorithm for clustering. As outlined in Algorithm 1, the process consists of four main steps: selecting the Mc cluster centres, grouping the remaining vehicles, updating the cluster centres, and repeating the process until the cluster centres remain constant. To expedite clustering, the algorithm prioritises selecting Mc vehicles that are widely separated. It is important to note that the k-means algorithm relies on Euclidean distance calculations, which may result in cluster centres that do not perfectly align with individual vehicles. Choosing cluster heads that align with these clusters is an additional refining step. The cluster head serves as the content cache source and manages access, requiring a reliable connection with other vehicles. Therefore, we utilised a parameter called "Connectivity Lifetime" (CL) to select cluster heads based on the durability of vehicle connections [126].Let $N_{i,l}$ represent the l-th car in cluster i. The vehicle's location and speed at time t are denoted as $L_{i,l}(t)$ and $S_{i,l}(t)$, respectively. Vehicles periodically send a "hello" beacon packet to other vehicles within the coverage area of the same BS. The "hello" beacon packet includes the vehicle's mobility information, such as its location $L_{i,l}(t)$ and speed $S_{i,l}(t)$. Using this information, each car can calculate its associations with other vehicles. The Connectivity Lifetime (CL) is the duration for which a communication link is maintained between two vehicles. When the CL expires, the link between the two vehicles is expected to terminate, indicating that their relative distance will have reached the maximum broadcasting range, denoted as δ, within this period. The requirement for maintaining a link between cars $N_{i,l}$ and $N_{i,l'}$ (where $l' \neq l$) is based on this principle:

$$\left(L_{i,l}(t + \mathrm{CL}_{i,l}^{i,l'}) - L_{i,l'}(t + \mathrm{CL}_{i,l}^{i,l'})\right)^2 = \delta^2 \tag{4.13}$$

If the vehicle maintains the same speed over the time interval τ, its location is given by:

$$L(t+\tau) = L(t) + \tau \cdot S(t) \tag{4.14}$$

Thus, equation (4.13) can be expressed as:

$$\left(L_{i,l}(t) - L_{i,l'}(t) + \mathrm{CL}_{i,l}^{i,l'}\left[S_{i,l}(t) - S_{i,l'}(t)\right]\right)^2 = \delta^2 \tag{4.15}$$

Therefore, $\mathrm{CL}_{i,l}^{i,l'}$ can be calculated by solving equation (4.15), where speed and location are represented as 2-D coordinates:

$$\mathrm{CL}_{i,l}^{i,l'} = \frac{\delta^2 - |\mathbf{L}_{i,l}(t) - \mathbf{L}_{i,l'}(t)|^2}{|\mathbf{S}_{i,l}(t) - \mathbf{S}_{i,l'}(t)|} \tag{4.16}$$

Once the clusters are formed using the K-means algorithm, the average CL for each vehicle $N_{i,l}$ in cluster i can be computed as follows:

$$\overline{\mathrm{CL}}_{i,l} = \frac{1}{n_i - 1} \sum_{l'=1}^{n_i} \mathrm{CL}_{i,l}^{i,l'}, \quad l' \neq l \tag{4.17}$$

Subsequently, the vehicle with the highest average CL within the cluster is designated as the cluster head. This selection ensures that the cluster head and all other vehicles within the same cluster maintain optimal link stability.

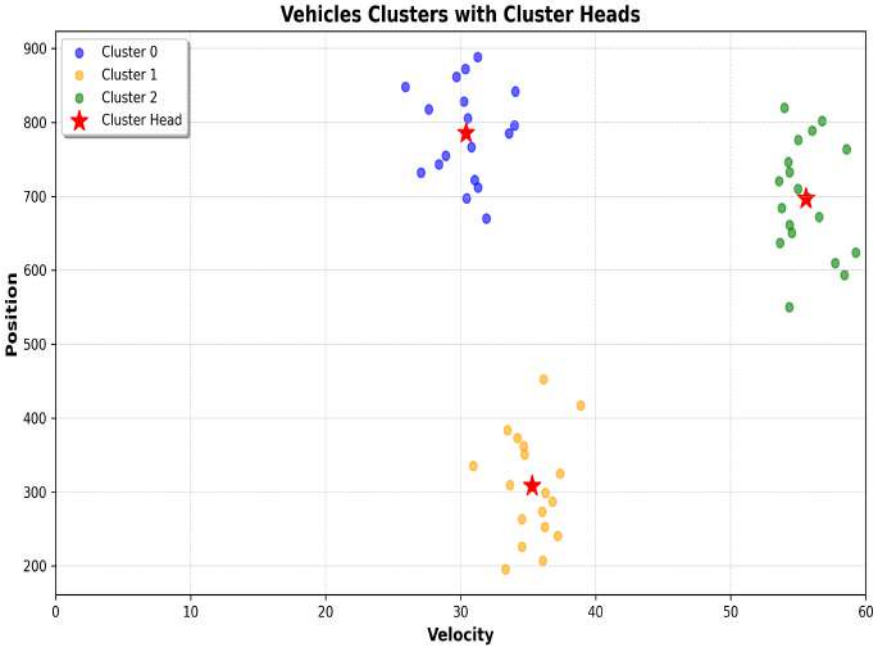


Figure 4.3: Vehicle clustering

Algorithm 1 Clustering Algorithm

Input: Vehicle set $V = \{V1, V2, \ldots, V_{nv}\}$, Number of vehicle clusters Mc
Output: Vehicle cluster set $\{c1, c2, \ldots, c_{Mc}\}$
Select initial cluster centres $\{t1, t2, \ldots, t_{Mc}\}$
repeat
    for $i = 1$ to Mc **do**
        Determine the distance between N_l and each centre t_i, denoted as $d_{i,l} = \|N_l - t_i\|_2$;
        Assign N_l to the cluster whose centre is the closest.
    end for
    for $i = 1$ to Mc **do**
        Calculate the new cluster centre $t_i^* = \frac{1}{|C_i|} \sum_{N_l \in C_i} N_l$;
        if $t_i^* \neq t_i$ **then**
            Update the cluster centre $t_i = t_i^*$;
        else
            Keep the current cluster centre constant;
        end if
    end for
until all cluster centres become fixed

4.2.2.3 Cooperative caching-based TS algorithm

Reinforcement learning is an algorithmic approach centred on mapping behaviours based on the environmental state [127]. In this framework, the agent within the reinforcement learning system selects and performs actions from a set based on the system's state, subsequently receiving a reward for the action taken. This reward serves as a metric to assess the effectiveness of the chosen action. After the training phase, the agent can swiftly identify and perform actions associated with higher rewards, depending on the system's state.

In the context of Multi-Armed Bandit (MAB) problems, each arm corresponds to a distinct action that yields a reward upon selection. However, the likelihood of receiving a reward varies across arms, presenting the challenge of efficiently selecting arms within a limited number of trials to optimise rewards. Balancing exploration and exploitation is a widely recognised challenge in this scenario. To maximise rewards, decision-makers must strike a balance between exploring new arms and exploiting the knowledge gained from prior exploration. The MAB problem, using

strategies like Thompson Sampling (TS), aims to maximise the expected reward, ensuring the best possible outcome [128].

TS is particularly effective for caching issues with high uncertainty, as its exploration of varied strategies can enhance overall performance. TS is a stochastic algorithm used for decision making in situations of uncertainty, commonly applied to MAB problems. It assigns scores to each arm by assuming the reward probability for each arm conforms to a Beta distribution [129]. This is a continuous probability distribution within the interval $[0, 1]$, characterised by two positive shape parameters, $\mathbf{a}$ and $\mathbf{b}$. The average value of the Beta distribution is given by $\frac{\mathbf{a}}{\mathbf{a}+\mathbf{b}}$. A larger $\mathbf{a}$ results in a higher average value of Beta$(\mathbf{a}, \mathbf{b})$, whereas a higher $\mathbf{b}$ decreases this average. The TS algorithm estimates the return, represented by ϕ, by sampling Beta$(\mathbf{a}, \mathbf{b})$ [129]. Following the outcome of arm selections, updates are made to the selected arm. Receiving a reward of 1 increases the corresponding $\mathbf{a}$ value by 1, while a reward of 0 increases the $\mathbf{b}$ value by 1. The inherent randomness of sampling enables TS to naturally achieve a balance between exploration and exploitation (EE). Additionally, adjustments to the EE balance can be made by altering the update mechanism for the TS parameters $\mathbf{a}$ and $\mathbf{b}$.

We designated the edge server, base station, cluster head, and each user with a local cache as agents. Each category is considered an arm, with each arm having a unique likelihood of being chosen. Once the file category is identified based on the probability, the file is selected from that category to cache. In this study, the TS technique is used to iteratively select actions (arms) based on their predicted reward probability. The algorithm tracks reward distribution estimations for each arm using Beta distributions. At each iteration, the algorithm samples Beta distributions to evaluate the likelihood of each arm being the best selection. It then chooses the arm with the highest calculated likelihood and adjusts its parameters according to the received rewards. In this context, rewards refer to cache hit rates, which indicate the ratio of successful task completions to the total number of requested tasks. To update the parameters of the Beta distribution at time-slot $t + 1$, the following formula is used:

$$\begin{aligned} a_{d,\text{cat}_i}^{t+1} &= a_{d,\text{cat}_i}^{t} + R_{d,\text{cat}_i}^{t}, \\ b_{d,\text{cat}_i}^{t+1} &= b_{d,\text{cat}_i}^{t} + (1 - R_{d,\text{cat}_i}^{t}) \end{aligned} \tag{4.18}$$

Here, R_{d,cat_i}^{t} is the hit rate (reward) of the selected category (arm) of the file in the cache of each device d at time t.

The TS algorithm process is outlined in Algorithm 3. The TS-based algorithm sorts contents into various categories. Without categorising the contents, each agent would have an action space of approximately $2^{|L|}$ when selecting from $|L|$ files, making the algorithm ineffective. The TS-based approach is suggested to decrease the action space of Q-learning [130].

Initially, the algorithm initialises the Beta distribution of each arm as follows:

$$\phi_{\text{cat}_i}^{0} \sim \text{Beta}(a_{\text{cat}_i}^{0}, b_{\text{cat}_i}^{0}) \leftarrow (1,1), \text{ where } 1 \leq i \leq N \tag{4.19}$$

where N is the number of arms. Consequently, the arm with the highest sampled value is selected as the optimal arm. According to the stored probabilities of each arm, the contents will be cached based on their requested probabilities in the subsequent time. Once the items are stored in the caches of each device, the rewards are acquired. The posterior distribution of each selected arm is updated as follows:

$$\phi_{\text{cat}_i}^{t+1} \sim \text{Beta}(a_{\text{cat}_i}^{t+1}, b_{\text{cat}_i}^{t+1}), \quad \text{where } 1 \leq i \leq N \tag{4.20}$$

Algorithm 2 TS-based Mobility-Aware Multi-Hierarchical Caching Model with Vehicle Clustering and Content Popularity Prediction Methods(TS-MMCM)

Phase 1: Initialization
Initialise parameters: number of vehicles, vehicle clusters, period T, content categories $|cat|$, repository, content sizes,Number of arms N, parameter s, cache capacity.
Initialise the Beta distribution parameters:

$$\phi^0_{\text{cat}_i} \sim \text{Beta}(a^0_{\text{cat}_i}, b^0_{\text{cat}_i}) \leftarrow (1,1), \quad , \text{ where } 1 \leq i \leq N$$

Phase 2: Vehicle Clustering
Perform Algorithm 1 to obtain vehicle clusters.
Phase 3: Obtain cluster heads
Choose cluster heads based on CL selection criterion: $\overline{\text{CL}}_{i,l} = \frac{1}{n_i - 1} \sum_{l'=1}^{n_i} \text{CL}_{i,l}^{i,l'}, \quad l' \neq l.$
Phase 4: Find the best caching decision
repeat (for each round)
    Obtain the predicted number of content requests $\theta(t)$ using LSTM.
    Sort $\theta(t)$ to obtain sorted index vector λ_Q.
    Obtain content popularity vector π_Q using Zipf model: $P(R; s, N) = \frac{1/R^s}{\sum_{q=1}^{N} \frac{1}{q^s}}$, where $q \in \{1, 2, \ldots, l\}$.
    for each cluster H_i **do**
        for each vehicle user $H_{i,j} \in H_i$ **do**
            Determine caching method in each local cache using a TS-based algorithm.
        end for
        Determine the caching policy in each cluster head $H_{i,1}$ using a TS-based algorithm.
    end for
    Determine caching policy in the edge server using a TS-based algorithm.
    Update contents in each cache.
until convergence criteria are met or fixed number of rounds T

Algorithm 3 Thompson Sampling Algorithm

Input: Initialise Parameters
Output: Report the selected arms and their corresponding probabilities.
Number of categories N
Initialise the posterior distributions of each arm as a function of (4.19).
Initialise variable for reward .
repeat
    for each round **do**
        Pull Arms:
            Sample from the posterior distributions of each arm.
            Select the arm with the highest sampled value.
        Obtain Reward:
            Execute the chosen action/arm in the environment.
            Update the cache.
            Observe the reward obtained.
        Update Parameters for the chosen arm:
            Update the posterior distribution of the selected arm based on observed reward according to (4.20).
            Update the reward for the chosen arm.
    end for
Next round
until convergence or fixed number of rounds

4.3 Proposed Caching Strategy

4.3.1 Description of the new caching strategy

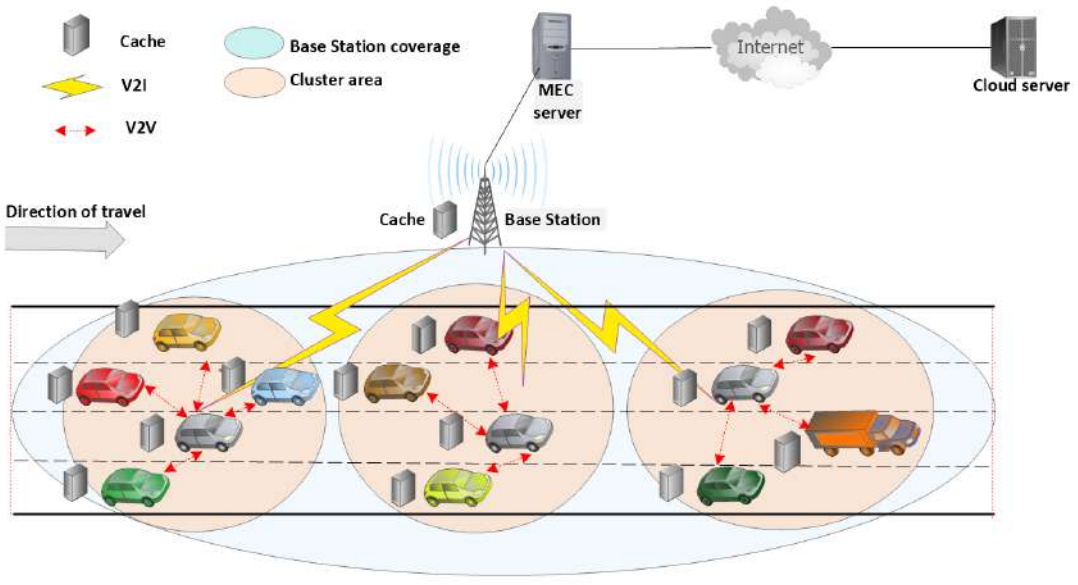


Figure 4.4: Proposed architecture

In this section, we introduce an architecture for multilevel data caching (depicted in Fig. 4.4), designed to optimize communication among cluster member users, cluster heads, base stations (BSs), and edge servers for caching various content types. Each edge server manages BSs within its coverage area, with multiple users typically found in each BS's predominant region, highlighting the vertical architecture's caching capacity. Vehicles within a BS's coverage are grouped into clusters based on their mobility traits. Each cluster is assigned a cluster head, a role that remains consistent over time slots.

Cluster heads are denoted as $H_{i,1}$, where i signifies the cluster index within the BS coverage area. Vehicles within clusters are designated as $H_{i,j}$, where j ($j \geq 2$) identifies individual vehicles within cluster i. Refer to Table 4.1 for a summary of all notations used.

For example, $H_{i,1}$, the cluster head, serves the members of its cluster. In this caching structure, when a vehicle initiates a request, it first checks its local cache for the requested content. If found, the content is accessed directly; otherwise, it seeks the content from $H_{i,1}$. If $H_{i,1}$ caches the content, the vehicle retrieves it directly; otherwise, it requests the BS coverage area. If the content is cached there, it's accessed immediately; otherwise, the request proceeds to the edge server and then to the cloud server if needed.

Grouping vehicles into clusters based on mobility traits enhances communication reliability among vehicles within the same BS coverage. Cluster heads are chosen consistently to ensure stability across time slots. Vehicles' normalized characteristics, such as position and speed, are used to form Mc clusters from Nv-generated vehicles. With every vehicle, including cluster heads, capable of caching content, cluster heads can communicate directly with other cluster members. However, communication between vehicles in different clusters is not established.

4.3.1.1 Caching model

Let $L = \{1, 2, \ldots, \mathrm{l}\}$ denote the content repository, where l represents the total number of content items. The cloud server stores the entire content repository L,

while the edge server collaborates with base stations, cluster heads, and vehicle users to cache this content. Each content item has a specific size, denoted by $Z = \{z_1, z_2, \ldots, z_l\}$, with l being an element of L. These contents are categorised into N categories, labelled as $cat = \{\text{category1}, \text{category2}, \ldots, \text{categoryN}\}$.

The edge server has a local cache capacity of C_{edge}. Each base station is equipped with a cache capacity denoted as C_{BS}, and each cluster head has a local cache with a capacity of $C_{H_{i,1}}$. Users within a cluster are represented by $V = \{v1, v2, \ldots, vu\}$, where specific content is stored in their local caches with a capacity of $C_{H_{i,j}}$.

The variables $S(l, \text{edge})$, $S(l, \text{BS})$, $S(l, H_{i,1})$, and $S(l, H_{i,j})$ are binary indicators:

- $S(l, \text{edge}) = 1$ indicates content l is stored at the edge; $S(l, \text{edge}) = 0$ otherwise.

- $S(l, \text{BS}) = 1$ indicates content l is stored in the base station; $S(l, \text{BS}) = 0$ otherwise.

- $S(l, H_{i,1}) = 1$ indicates content l is stored in cluster head $H_{i,1}$; $S(l, H_{i,1}) = 0$ otherwise.

- $S(l, H_{i,j}) = 1$ indicates content l is cached in the local cache of user $H_{i,j}$ within cluster i; $S(l, H_{i,j}) = 0$ otherwise.

Given the constrained caching capacities of the edge server, base stations, cluster heads, and cluster members $H_{i,j}$, it is vital to ensure that the total size of cached contents on each device remains within its maximum capacity at every time slot t.

Table 4.1: Summary of Important Notations

Symbol	Description
BS	Base station
$H_{i,1}$	Cluster head
$H_{i,j}$	Cluster member
Nv	Total number of vehicles in the system
Mc	Number of clusters grouping the vehicles
L	Content repository
Z	Size of each file repository
cat	Category of content
V	Set of vehicles
C_{edge}	Cache capacity of edge cache
C_{BS}	Cache capacity of base station cache
$C_{H_{i,1}}$	Cache capacity of cluster head cache
$C_{H_{i,j}}$	Cache capacity of local cache
$S(l, \text{edge})$	Cache state of the file l in the edge
$S(l, BS)$	Cache state of the file l in the BS
$S(l, H_{i,1})$	Cache state of the file l in the cluster head
$S(l, H_{i,j})$	Cache state of the file l in the cluster member
$HitR_t^{\text{edge}}$	Cache hit rate of the edge
$HitR_t^{BS}$	Cache hit rate of the BS
$HitR_t^{H_{i,1}}$	Cache hit rate of the cluster head
$HitR_t^{H_{i,j}}$	Cache hit rate of the cluster member
Θ_{edge}	Caching edge policy
$\Theta_{H_{i,1}}$	Caching cluster head policy
$\Theta_{H_{i,j}}$	Caching cluster member policy
$L_{i,l}(t)$	Location of vehicle
$S_{i,l}(t)$	Speed of vehicle
CL	Connectivity lifetime

$$\begin{cases} \sum_{e=1}^{l} z_e \cdot S(e, \text{edge}) \leq C_{\text{edge}}, & \forall e \in L \\ \sum_{r=1}^{l} z_r \cdot S(r, \text{BS}) \leq C_{\text{BS}}, & \forall r \in L \\ \sum_{b=1}^{l} z_b \cdot S(b, H_{i,1}) \leq C_{H_{i,1}}, & \forall b \in L \\ \sum_{d=1}^{l} z_d \cdot S(d, H_{i,j}) \leq C_{H_{i,j}}, & \forall d \in L \end{cases}$$

Let $L = \{1, 2, \ldots, \text{l}\}$ represent the content repository, where l is the total number of content items. The cloud server stores the complete content repository L, while the edge server collaborates with base stations (BSs), cluster heads, and vehicle users to cache content. Each content item has a distinct size, denoted by $Z = \{\text{z1}, \text{z2}, \ldots, \text{zl}\}$. These contents are grouped into N categories, represented as $cat = \{\text{category1}, \text{category2}, \ldots, \text{categoryN}\}$.

The edge server has a local cache capacity of C_{edge}, the BS has a capacity of C_{BS}, and each cluster head $H_{i,1}$ has a local cache capacity of $C_{H_{i,1}}$. Users within clusters, denoted as $H_{i,j}$ (where $j \geq 2$), have local cache capacities of $C_{H_{i,j}}$.

The cache state of content at each location or equipment is represented as $S(e, \text{edge})$, $S(r, \text{BS})$, $S(b, H_{i,1})$, and $S(d, H_{i,j})$, indicating whether content is cached $(= 1)$ or not cached $(= 0)$.

Due to the limited caching capacities of the edge, BS, $H_{i,1}$, and $H_{i,j}$, it is crucial to ensure that the total size of cached contents at each device does not exceed its maximum capacity at every time slot t.

We propose a cache-placement technique based on content popularity. Content request probabilities are influenced by their popularity ranking, with more popular content having higher request probabilities. The Zipf distribution is commonly used to model these probabilities. The probability of content ranked R in popularity, denoted as $P(R; s, N)$, is given by:

$$P(R; s, N) = \frac{1/R^s}{\sum_{q=1}^{N} \frac{1}{q^s}} \quad \text{where } q \in \{1, 2, \ldots, l\} \tag{4.21}$$

Here, s is the exponent parameter characterising the Zipf distribution, and l is the total number of content items.

To evaluate caching strategies, we assess the cache hit rate, which indicates the effectiveness of the caching approach. The cache hit rates for the edge server, BS, cluster heads, and cluster members are calculated as follows:

$$HitR_t^{\text{edge}} = \sum_{e=1}^{l} P_e \cdot S(e, \text{edge}) \quad \forall e \in L \tag{4.22}$$

$$HitR_t^{\text{BS}} = \sum_{r=1}^{l} P_r \cdot S(r, \text{BS}) \quad \forall r \in L \tag{4.23}$$

$$HitR_t^{H_{i,1}} = \sum_{b=1}^{l} P_b \cdot S(b, H_{i,1}) \quad \forall b \in L \tag{4.24}$$

$$HitR_t^{H_{i,j}} = \sum_{d=1}^{l} P_d \cdot S(d, H_{i,j}) \quad \forall d \in L \tag{4.25}$$

Here, $\mathbf{P}_l = [p_1, p_2, \ldots, p_l]$ denotes the probability vector indicating the likelihood of each content being requested by vehicle users in the upcoming time period.

4.3.1.2 Content Request Model

All components of our system, including vehicle users, cluster heads, base stations, and edge servers, are equipped to cache content. When a vehicle within the cluster requests specific content, it can be obtained from one of five potential locations:

4.3.1.2.1 Local Cache The vehicle user's local cache is the first point of access to locate requested content. If the content is available in the local cache, it can be accessed directly.

4.3.1.2.2 Cluster Head If the content is not found in the local cache, the user retrieves it from the cluster head. The requested content is directly retrieved from the cluster head if it has been cached, using vehicle-to-vehicle (V2V) connectivity.

4.3.1.2.3 Base Station If the content is not available at the cluster head, the user sends the request to the BS that covers its location. The content is instantly accessed from the BS if cached, using vehicle-to-infrastructure (V2I) connectivity.

4.3.1.2.4 Edge Server If the content is not available at the BS, the request is transmitted to the edge server through the BS. If cache retrieval fails, the request is forwarded to the cloud server, resulting in increased latency.

4.3.1.3 Communication Model

This section outlines two communication modes, V2I and V2V, used by vehicles. Vehicles select one mode at a time to access content.

4.3.1.3.1 V2V Communication Link In V2V communication, vehicles within a cluster exchange data directly, managed by a cluster head. Incoming communications involve receiving traffic updates, hazard alerts, and data requests from other vehicles, which can be cached or forwarded. Outgoing communications involve sending responses, sharing local traffic information, and broadcasting emergency alerts. The cluster head facilitates efficient data distribution, reducing latency and minimising load on the central base station, ensuring quick access and dissemination of information within the network.

4.3.1.3.2 V2I Communication Link The base station uses the V2I link to send cached content to the requesting vehicle within its coverage range.

The V2I link supports various applications, including real-time traffic management, navigation assistance, and infotainment services. When a vehicle requests specific content, the base station checks its cache and sends the data directly to the vehicle if available. This reduces redundant data retrieval from external servers, improving network efficiency and reducing latency.

4.3.1.4 Rationale behind the strategy

The cache hit rate is a critical metric in caching techniques, indicating how well cached content aligns with user requests. Our goal was to develop an online caching technique that maximises this hit ratio. Given the vast number of files, we categorised content into types and assigned probabilities for their selection. Once identified based on probability, one file of each type is chosen for caching. The

objective for caching at each vehicle user, cluster, and edge is to maximise the hit rate, formulated as:

$$\max_{\Theta_{H_{i,j}}} HitR_t^{H_{i,j}}(\Theta_{H_{i,j}}) \tag{4.26}$$

$$\max_{\Theta_{H_{i,1}}} HitR_t^{H_{i,1}}(\Theta_{H_{i,1}}) \tag{4.27}$$

$$\max_{\Theta_{\text{edge}}} HitR_t^{\text{edge}}(\Theta_{\text{edge}}) \tag{4.28}$$

Our approach aims to enhance cache hit rates both at the edge server and the vehicle user level. Considering the architecture model and constraints on caching capabilities of vehicle users, cluster heads, and edge servers, our problem formulation is:

$$\text{maximise } \{\theta_{H_{i,j}}, \theta_{H_{i,1}}, \theta_{\text{edge}}\} \quad \text{for } HitR_t^{H_{i,j}}, \tag{4.29a}$$

$$\text{maximise } \{\theta_{H_{i,j}}, \theta_{H_{i,1}}, \theta_{\text{edge}}\} \quad \text{for } HitR_t^{H_{i,1}}, \tag{4.29b}$$

$$\text{maximise } \{\theta_{H_{i,j}}, \theta_{H_{i,1}}, \theta_{\text{edge}}\} \quad \text{for } HitR_t^{\text{edge}}, \tag{4.29c}$$

$$\text{maximise } \{\theta_{H_{i,j}}, \theta_{H_{i,1}}, \theta_{\text{edge}}\} \quad \text{for } HitR_t^{\text{total}}. \tag{4.29d}$$

Under constraints $C1$:

$$\begin{cases} S(l, \text{edge}) \\ S(l, \text{BS}) \\ S(l, H_{i,1}) \\ S(l, H_{i,j}) \end{cases} \in \{0, 1\}, \quad \forall l \in L$$

And under constraints $C2$:

$$\begin{cases} \sum_{e=1}^{l} z_e \cdot S(e, \text{edge}) \leq C_{\text{edge}}, & \forall e \in L \\ \sum_{r=1}^{l} z_r \cdot S(r, \text{BS}) \leq C_{\text{BS}}, & \forall r \in L \\ \sum_{b=1}^{l} z_b \cdot S(b, H_{i,1}) \leq C_{H_{i,1}}, & \forall b \in L \\ \sum_{d=1}^{l} z_d \cdot S(d, H_{i,j}) \leq C_{H_{i,j}}, & \forall d \in L \end{cases}$$

Here, each content is either cached or not (represented as 1 or 0 for $S(l, \text{edge})$, $S(l, \text{BS}), S(l, H_{i,1})$, and $S(l, H_{i,j})$). The final constraint ensures that the total size of cached content at the edge, BS, $H_{i,1}$, and $H_{i,j}$ does not exceed their respective capacities. Reinforcement Learning (RL) is proposed as an effective approach to address this multi-agent decision-making problem.

4.4 Evaluation

4.4.1 Metrics for Evaluating the Strategy

The study assessing a caching strategy, rooted in content popularity and Zipf distribution, employed various metrics to gauge the effectiveness and efficiency of the proposed method. Utilising Python simulations, the performance was evaluated through key metrics including hit rate and latency.

Hit Rate

- The hit rate denotes the percentage of data or resource requests effectively retrieved from the cache, reflecting the ratio of successful task completions to the total number of requested tasks.
- The hit rate was computed across various cache locations, spanning the edge server, cluster head, and cluster members, providing insights into the efficacy of the caching strategy across the network.

Latency

- Latency is defined as the time duration from when a request is sent by the vehicle user to when the last data packet is received.
- This metric plays a critical role in assessing caching strategies, particularly in contexts where reducing delays is vital for network efficiency and optimal user experience.

4.4.2 Simulation Setup

- The simulation scenario featured a library comprising 1,000 items, each sized between 5 MB and 100 MB, accumulating to a total content size of 51,000 MB.
- The cache capacity of the edge server was configured to be between 10% and 25% of the entire volume of content. The cache capacity allocated to each user and cluster head was adjusted to be between 10% and 25% of the cache size of the edge server.
- The simulation utilised a TS-based Mobility-Aware Multi-Hierarchical Caching Model with Vehicle Clustering and Content Popularity Prediction Methods (TS-MMCM).
- The arrival of requests from vehicles was modelled as a Poisson process in each time slot, and user requests were simulated by varying the shape parameter values of the Zipf distribution.

Table 4.2 lists the simulation's parameters and their values.

Table 4.2: Simulation Parameters

Parameter	Value
Simulation area	2 km
Vehicle speed	[20, 60] km/h
Request rate	[20, 30] requests /m
Number of vehicle users	55
Number of file categoies	5
cache capacity rate	[10,15,20, 25]%

The evaluation metrics for hit rate and latency, along with the simulation configuration details, provide insight into how the performance of the caching strategy was assessed and the parameters involved in the simulation process. The

study's focus on these metrics and simulation methodologies highlights the thorough analysis conducted to validate the proposed caching approach.

4.5 Results Interpretation

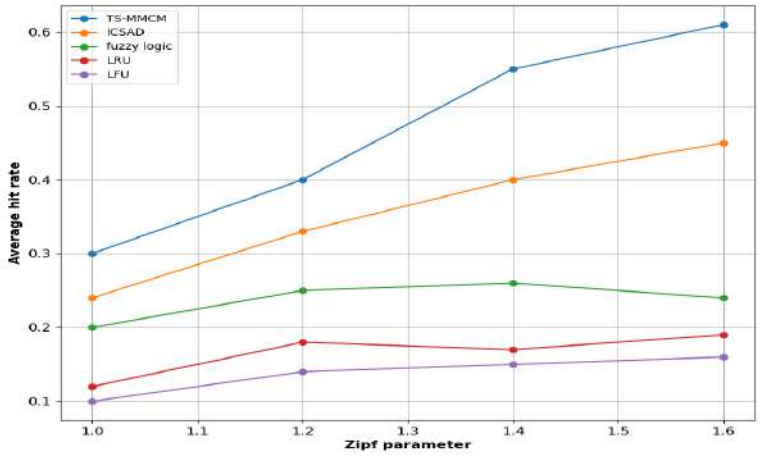
4.5.1 Analysis and Interpretation of Data

The data presented in the simulation results offers a comprehensive evaluation of the proposed TS-MMCM algorithm's performance across different caching strategies and under varying conditions influenced by the Zipf parameter and the cache capacity rates.

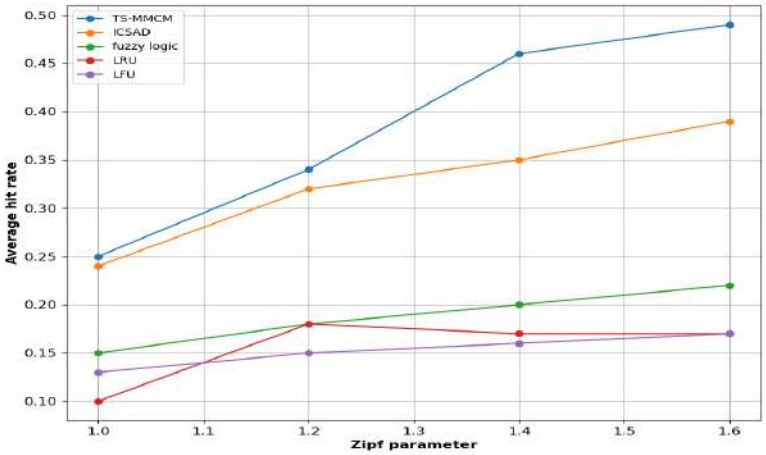
4.5.1.1 Hit Rate Analysis

- The TS-MMCM algorithm demonstrates a clear superiority in terms of hit rate, which progressively increases with the rise in the Zipf parameter (from 1.0 to 1.6), as depicted in Fig.4.5a. The hit rate increases from 0.3 to over 0.6, highlighting the algorithm's adaptive efficiency to the popularity distribution of content requests.

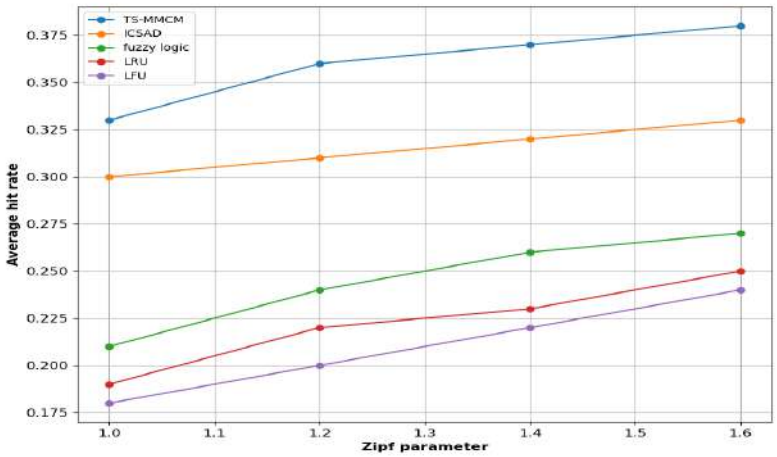
- A comparative analysis in Fig.4.5b and Fig.4.5c further illustrates how TS-MMCM consistently outperforms other algorithms (LRU, LFU, Fuzzy Logic, ICSAD) across various cache configurations and cluster head caches.



(a) Average hit rate vs. Zipf parameter for local cache



(b) Average hit rate vs. Zipf parameter for cluster head cache



(c) Hit rate vs. Zipf parameter for edge cache

Figure 4.5: Comparative analysis of caching strategies across different cache locations

4.5.1.2 Capacity Utilisation

- Fig4.6 explores the relationship between cache capacity rates (ranging from 0.1 to 0.25) and hit rates. TS-MMCM consistently achieves the highest hit rates across all capacity rates, indicating optimal utilisation of available cache space for storing frequently requested content.

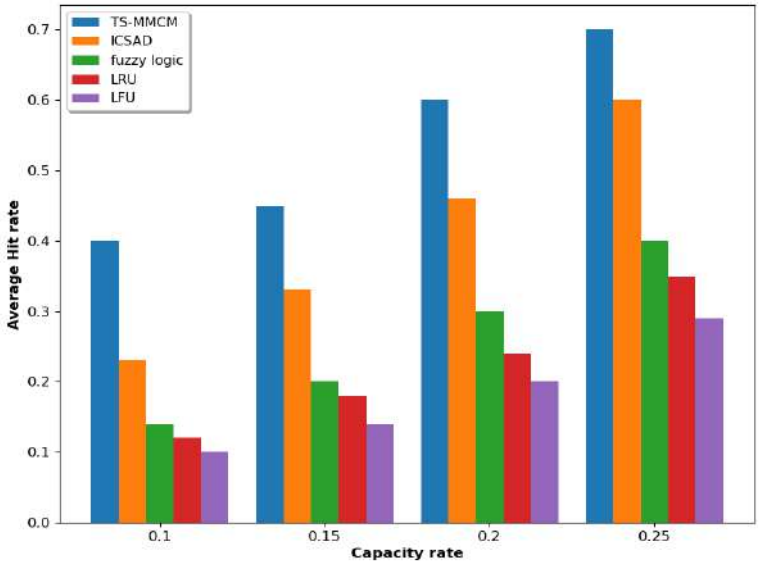


Figure 4.6: Cache hit ratio versus total caching capacity rate

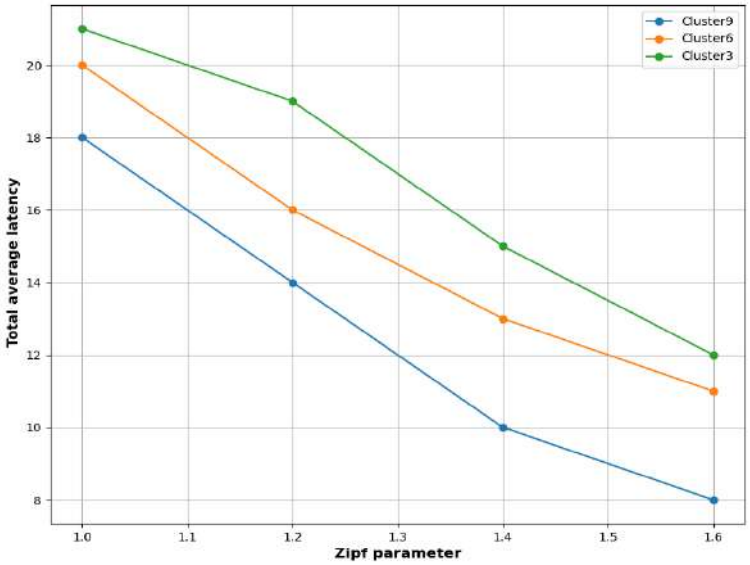
4.5.1.3 Latency Optimization

Advanced clustering techniques based on vehicle speed and position significantly reduce latency, as shown in Fig.4.7a and Fig.4.7b. The total average latency decreases as the number of clusters increases, confirming the clustering mechanism's effectiveness in reducing response times.

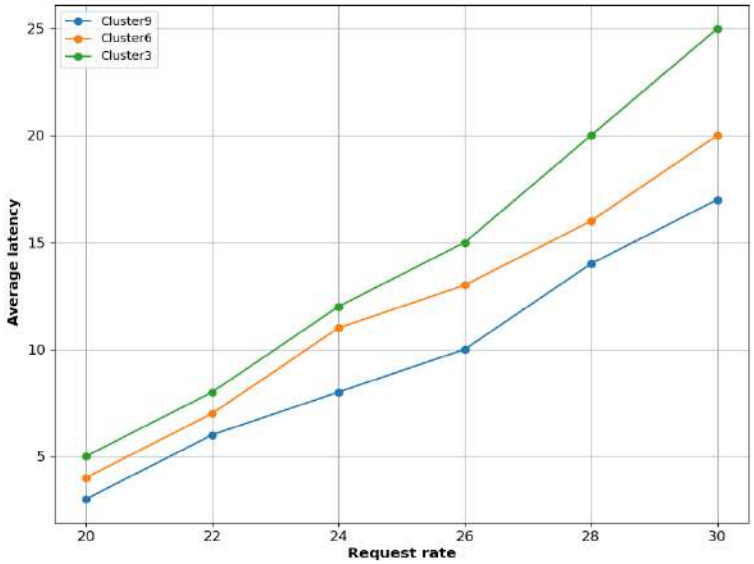
4.5.2 Comparison with existing strategies

4.5.2.1 Comparison to Conventional Caching Methods

- LRU and LFU: These traditional caching strategies exhibit a lower performance trend in comparison to TS-MMCM. As depicted in Fig.4.5b, LRU shows only a modest improvement in hit rates as the Zipf parameter increases,



(a) Total average latency vs. Zipf parameter



(b) Average latency vs. Request rate

Figure 4.7: Vehicle clustering performances

while LFU, illustrated in Fig.4.6, remains the least effective across all tested capacity rates.

- Fuzzy Logic and ICSAD: ICSAD[130] stands out as the second-best performer, particularly in Fig.4.6, demonstrating a moderate increase in hit rate as the Zipf parameter rises. Fuzzy Logic, though showing slight improvements in Fig.4.5c, is consistently outperformed by the adaptive capabilities of TS-MMCM.

4.5.2.2 Effectiveness in Different Caching Contexts

The diverse scenarios simulated, including different numbers of clusters (Fig. 4.7) and varying cache capacity rates (Fig.4.6), also underscore TS-MMCM's robustness. Whether in edge caching scenarios or localized cluster head caches, TS-MMCM consistently adapts better than other tested strategies, maintaining higher hit rates and reduced latency.

4.5.2.3 Innovative Algorithmic Approach

TS-MMCM's ability to outperform both established and novel caching strategies confirms its innovative approach to handling data within Internet of Vehicles (IoV) environments. Its design allows for quick adaptation to the evolving network structures and user behavior patterns, which are critical in time-sensitive applications. Fig.4.8 shows the effectiveness of vehicle clustering in enhancing cache hit rates, demonstrating the algorithm's predictive accuracy and proactive caching capabilities.

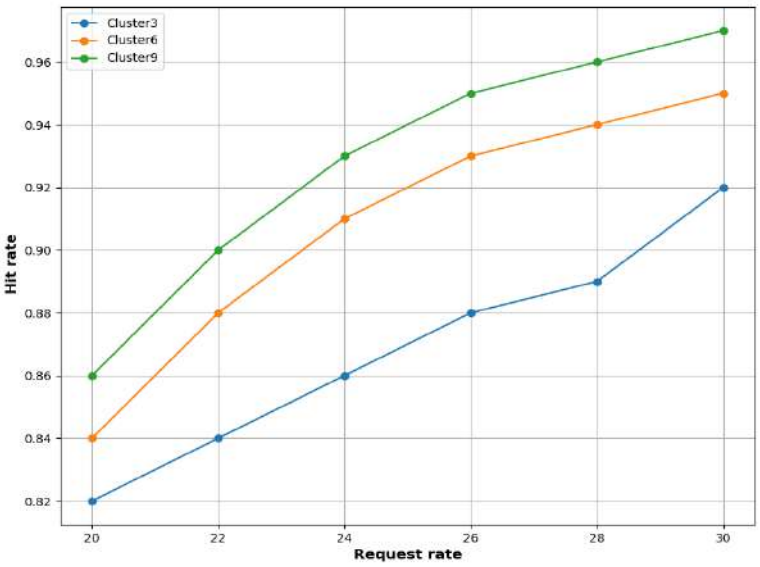


Figure 4.8: Cache hit ratio vs Request rate

4.6 Conclusion

The proposed TS-MMCM framework not only addresses the limitations of existing caching strategies but also sets a new benchmark for performance in IoV systems. It ensures that the most demanded content is available at the network edge, thereby reducing latency and enhancing the user experience. This study confirms the efficacy of integrating machine learning and optimisation techniques in network traffic management and opens up new avenues for future research in caching strategies within the burgeoning field of vehicular networks.

Chapter 5

Reflection and Future Directions

Contents

5.1 Introduction

This thesis investigates caching system optimisation within vehicular networks by integrating advanced techniques such as Thompson sampling-based learning, vehicle clustering, and content popularity prediction. The primary goal is to enhance cache management in these dynamic networks, with a focus on boosting cache hit rates and minimising latency. This research is crucial in the context of increasing mobility and the need for real-time data access, where effective cache management significantly impacts network performance and user security. Employing machine learning and predictive analytics, this study presents novel strategies to tailor cache resources to the evolving demands of vehicular networks.

5.2 Summary of Findings

5.2.1 Recap of key findings

This research demonstrates a significant advancement in the efficiency of caching systems for vehicular networks through the adoption of advanced machine learning and clustering techniques. The developed model, the Thompson Sampling-based Mobility-Aware Multi-Hierarchical Caching Model (TS-MMCM), was rigorously evaluated through simulations, showcasing superior effectiveness over popular cache management approaches.

The incorporation of Thompson sampling allowed our model to dynamically adapt caching strategies in real-time to fluctuations in demand and content popularity, markedly enhancing cache hit rates beyond conventional methods. Moreover, the use of clustering techniques has enabled effective vehicle grouping based on position and velocity, significantly reducing latency by minimising data transmission distances. This strategic grouping also improved content distribution efficiency, further lowering system latency.

Simulations confirmed that TS-MMCM excels in dynamic environments, where network conditions and user mobility patterns frequently change. Additionally, the integration of content popularity predictions through deep learning enabled accurate forecasting of future requests, ensuring data availability in the cache precisely when needed.

These results affirm the effectiveness of TS-MMCM as a superior cache management solution in vehicular networks and open avenues for future research into optimising intelligent networks and adapting to emerging technologies.

5.2.2 Contributions to the Field

This thesis makes a substantial contribution to the domain of vehicular networks and caching systems through the development of the TS-MMCM (Thompson Sampling-based Mobility-Aware Multi-Hierarchical Caching Model). This innovative model

integrates multiple advanced technologies to enhance cache management within the intricate and dynamic environments of vehicular networks.

5.2.2.1 Integrating Thompson Sampling Learning

This research incorporates Thompson Sampling, a reinforcement learning method, to refine cache management decisions within vehicular networks. By applying this technique, the model dynamically and efficiently selects content for caching, adjusting to the shifting popularity of data based on continuously updated probabilities. This adaptive strategy marks a significant enhancement over conventional approaches, offering more precise and responsive cache management to meet fluctuating user demands effectively.

5.2.2.2 Vehicle Clustering for Efficient Cache Management

Vehicle clustering techniques enhance cache management by grouping vehicles based on their positions and speeds. This strategy improves cache placement, leading to reduced response times and more efficient bandwidth usage, which is essential in environments characterised by high density and mobility.

5.2.2.3 Advanced Content Popularity Prediction

This thesis incorporates content popularity prediction algorithms based on deep learning techniques, facilitating proactive anticipation of data requests. This predictive capability transforms cache management from a traditionally reactive strategy to a proactive one, thereby reducing network load and enhancing the user experience with shorter loading times.

5.2.2.4 Adaptability to Dynamic Environments

The TS-MMCM model is specifically engineered to respond effectively to rapid shifts in network conditions, making it highly suitable for real-time applications such as navigation, vehicle safety, and vehicle-to-infrastructure communications.

5.3 Recommendations

5.3.1 Practical Recommendations

Based on the compelling outcomes of this research, practical recommendations can be formulated for decision-makers and engineers in the domains of intelligent transport networks and vehicle-to-vehicle (V2V) communications. These recommendations are designed to enhance resource management and boost the overall efficiency of vehicular network systems through the integration of advanced predictive analytics and machine learning techniques.

5.3.1.1 Adoption of Predictive Machine Learning Models

We recommend the adoption of predictive models, such as LSTM networks, to forecast content demand. This strategy enables caching systems to preload necessary data before it is requested, thereby reducing response times and alleviating network congestion. Additionally, predictive models are valuable for identifying usage patterns and optimising network resources to meet changing demands.

5.3.1.2 Implementing Dynamic Clustering

For effective management of the diversity and mobility of vehicles in vehicular networks, it is crucial to implement dynamic clustering techniques. These techniques should utilise criteria such as vehicle position and speed to enhance network performance and reliability.

5.3.1.3 Integration of Hierarchical Cache Systems

Deploy hierarchical cache systems that use both local and network caches to respond to requests more flexibly and efficiently. This strategy reduces dependency on central connections and cuts data retrieval times.

5.3.1.4 Ongoing Training and Awareness

It is crucial to provide continuous training for technical teams on the latest advancements in machine learning and cache management. This ensures that the

system remains current with cutting-edge innovations and best practices, thereby enhancing overall operational effectiveness.

5.3.1.5 Regular Evaluation and Updating of Cache Policies

It is essential to conduct periodic evaluations of cache system performance to identify areas for enhancement. Cache policies should be frequently reviewed and modified based on data analysis from predictive models. This ongoing adjustment ensures that the system remains optimised to effectively handle evolving traffic patterns and technological advancements.

By integrating these strategies, decision-makers and engineers can significantly enhance the efficiency and responsiveness of vehicular networks. Additionally, these measures contribute to improving user experience, reducing operational costs, and laying the groundwork for the future of intelligent, connected transport systems.

5.3.2 Suggestions for Future Research

The findings from this study present several promising directions for future research in the realm of optimised cache systems and intelligent transport networks. Suggestions for expanding and deepening this body of work include:

5.3.2.1 Integration of More Advanced Deep Learning Models

Future research could incorporate and test the effectiveness of even more sophisticated deep learning models, such as convolutional neural networks (CNNs) or generative adversarial networks (GANs), for predicting content popularity. These models could potentially capture more complex nuances in traffic data and usage behaviour, offering even more accurate and effective predictions.

5.3.2.2 Data Security and Confidentiality

A crucial aspect for further exploration is the enhancement of data security in machine learning-optimised caching systems. Developing robust mechanisms to protect data against unauthorised access and attacks is essential, while ensuring data confidentiality and integrity. Future research could investigate the integration

of advanced encryption techniques or blockchain-based solutions to secure data transactions in distributed environments.

5.3.2.3 Multi-Criteria Optimisation of Network Performance

Future research could explore multi-criteria optimisation approaches that consider not just latency and cache hit rates but also additional factors such as energy consumption and operational costs. This comprehensive perspective would facilitate the development of more holistic and economically sustainable cache management systems.

5.3.2.4 Cache Adaptation and Customisation

Future research could explore strategies for tailoring cache management to individual user preferences and specific network conditions. This approach might involve employing reinforcement learning to dynamically adjust cache policies in real time, allowing for more personalised and efficient cache utilisation.

These research directions could enhance the understanding and effectiveness of caching systems in vehicular networks and also broaden the scope for applying these technologies across a wider array of communication and data scenarios. This could lead to significant improvements in network efficiency and user experience in diverse technological environments.

5.4 Conclusion

This thesis has made significant contributions to optimising caching systems in vehicular networks through the application of innovative techniques such as Thompson sampling, vehicle clustering, and content popularity prediction. The findings have shown marked improvements in cache hit rates and reductions in latency, thereby enhancing cache management within dynamic network settings.

The recommendations derived from this research underscore the necessity of continuing to incorporate predictive analytics and machine learning to refine the performance of intelligent transport networks. Looking forward, it is recommended

to investigate more advanced deep learning models, enhance data security measures, and adopt multi-criteria optimisation strategies to create more comprehensive and economically sustainable cache management systems.

In conclusion, this thesis establishes a solid groundwork for future developments in network technologies, paving the way for a more interconnected and intelligent future in transport systems.

Chapter 6

Conclusion and Publications

Contents

6.1 Final thoughts

This thesis has extensively examined the optimisation of cache management in vehicular networks, employing advanced methodologies such as Thompson sampling and vehicle clustering. The implementation of the TS-MMCM model showcased its exceptional ability to adapt dynamically to fluctuations in data demand, marking a substantial evolution from traditional approaches. Empirical tests validate that this strategy significantly enhances network performance by optimising bandwidth utilisation and reducing latency.

The findings underscore the critical importance of integrating predictive analytics and machine learning technologies to proactively address variations in data demand—a key component for the advancement of tomorrow's intelligent transport systems. Moving forward, it is recommended to delve into deeper learning architectures and to develop robust security measures to improve both the reliability and security of data within vehicular networks.

6.2 Closing remarks

This thesis significantly advances the fields of network engineering and data science by presenting innovative cache management solutions tailored for dynamic and demanding network environments. The research directions proposed pave the way for enhancements in vehicular communication systems, particularly through the adoption of sophisticated data management and predictive analytics techniques. Advancing our understanding in these areas is crucial for addressing future challenges and unlocking the full potential of the Internet of Vehicles. This work not only contributes to the technological landscape but also sets the stage for transformative changes in how we manage and optimise vehicular networks.

6.3 Publication list

Conference publications

- B. Radouane, G. Lyamine, K. Ahmed and B. Kamel, " Scalable Mo bile Computing : From Cloud Computing to Mobile Edge Compu ting", 2022 5th International Conference on Networking, Information Systems and Security : Envisage Intelligent Systems in 5g//6G-based Interconnected Digital Worlds (NISS), Bandung, Indonesia, 2022, pp. 1-6, (https ://doi.org/10.1109/NISS55057.2022.10085600).

Journal publications

- B. Radouane, G. Lyamine and K. Ahmed, "TS-based Mobility-Aware Multi-Hierarchical Caching Model with Vehicle Clustering and Content Popularity Prediction Methods",In:Journal of Telecommunications and Information Technology,3(2024),(https://doi.org/1026636/jtit.2024.3.1616)

References

[1] Imrich Chlamtac, Marco Conti, and Jennifer J-N Liu. "Mobile ad hoc networking: imperatives and challenges." In: *Ad hoc networks* 1.1 (2003), pp. 13–64. DOI: https://doi.org/10.1016/S1570-8705(03)00013-1.

[2] J. A. Guerrero-Ibáñez, C. Flores-Cortés, and Sherali Zeadally. "Vehicular Ad-hoc Networks (VANETs): Architecture, Protocols and Applications." In: *Next-Generation Wireless Technologies: 4G and Beyond*. Ed. by Naveen Chilamkurti, Sherali Zeadally, and Hakima Chaouchi. London: Springer London, 2013, pp. 49–70. DOI: 10.1007/978-1-4471-5164-7_5. URL: https://doi.org/10.1007/978-1-4471-5164-7_5.

[3] Jawaher Abdulwahab Fadhil and Qusay Idrees Sarhan. "Internet of Vehicles (IoV): A survey of challenges and solutions." In: *2020 21st International Arab Conference on Information Technology (ACIT)*. IEEE. 2020, pp. 1–10. DOI: 10.1109/ACIT50332.2020.9300095.

[4] Juan Contreras-Castillo, Sherali Zeadally, and Juan Antonio Guerrero-Ibañez. "Internet of Vehicles: Architecture, Protocols, and Security." In: *IEEE Internet of Things Journal* 5.5 (2018), pp. 3701–3709. DOI: 10.1109/JIOT.2017.2690902.

[5] S. EL Madani, S. Motahhir, and A. EL Ghzizal. "Internet of vehicles: concept, process, security aspects and solutions." In: *Multimedia Tools and Applications* 81 (2022), pp. 16563–16587. DOI: 10.1007/s11042-022-12386-1. URL: https://doi.org/10.1007/s11042-022-12386-1.

[6] Hamideh Taslimasa et al. "Security issues in Internet of Vehicles (IoV): A comprehensive survey." In: *Internet of Things* 22 (2023), p. 100809. DOI: 10.1016/j.iot.2023.100809. URL: https://www.sciencedirect.com/science/article/pii/S2542660523001324.

[7] P Sathya Narayanan and C Sheeba Joice. "Vehicle-to-vehicle (v2v) communication using routing protocols: a review." In: *2019 International Conference on Smart Structures and Systems (ICSSS)*. IEEE. 2019, pp. 1–10. DOI: 10.1109/ICSSS.2019.8882828.

[8] Aldosary Saad, Ahmed Shalaby, and Abdallah A Mohamed. "Research on the internet of vehicles assisted traffic management systems for observing traffic density." In: *Computers and Electrical Engineering* 101 (2022), p. 108100. DOI: 10.3390/s22155535. URL: https://www.mdpi.com/1424-8220/22/15/5535.

[9] Guangzhen Cui et al. "Cooperative perception technology of autonomous driving in the internet of vehicles environment: A review." In: *Sensors* 22.15 (2022), p. 5535. DOI: 10.3390/s22155535. URL: https://www.mdpi.com/1424-8220/22/15/5535.

[10] Baofeng Ji et al. "Survey on the internet of vehicles: Network architectures and applications." In: *IEEE Communications Standards Magazine* 4.1 (2020), pp. 34–41. DOI: 10.1109/MCOMSTD.001.1900053.

[11] Peter Arthurs et al. "A taxonomy and survey of edge cloud computing for intelligent transportation systems and connected vehicles." In: *IEEE Transactions on Intelligent Transportation Systems* 23.7 (2021), pp. 6206–6221. DOI: 10.1109/TITS.2021.3084396.

[12] Salahadin Seid Musa et al. "Mobility-aware proactive edge caching optimization scheme in information-centric iov networks." In: *Sensors* 22.4 (2022), p. 1387. DOI: 10.3390/s22041387.

[13] Pierfrancesco Bellini et al. "Data Sources and Models for Integrated Mobility and Transport Solutions." In: *Sensors* 24.2 (2024), p. 441. DOI: 10.3390/s24020441. URL: https://www.mdpi.com/1424-8220/24/2/441.

[14] Muhammad Sameer Sheikh, Jun Liang, and Wensong Wang. "Security and privacy in vehicular ad hoc network and vehicle cloud computing: a survey." In: *Wireless Communications and Mobile Computing* 2020.1 (2020), p. 5129620. DOI: 10.1155/2020/5129620. URL: https://doi.org/10.1155/2020/5129620.

[15] Sulaiman M Karim et al. "Architecture, protocols, and security in IoV: Taxonomy, analysis, challenges, and solutions." In: *Security and Communication Networks* 2022.1 (2022), p. 1131479. DOI: /10.1155/2022/1131479.

[16] Naser Zaeri. "A heterogeneous short-range communication platform for internet of vehicles." In: *International Journal of Electrical and Computer Engineering (IJECE)* 11.3 (2021), pp. 2165–2177. DOI: 10.11591/ijece.v11i3.pp2165-2177.

[17] Sumit Kumar and Jaspreet Singh. "Internet of Vehicles over VANETs: smart and secure communication using IoT." In: *Scalable Computing: Practice and Experience* 21.3 (2020), pp. 425–440. DOI: 10.12694/scpe.v21i3.1741.

[18] Waleed Albattah et al. "An overview of the current challenges, trends, and protocols in the field of vehicular communication." In: *Electronics* 11.21 (2022), p. 3581. DOI: 10.3390/electronics11213581. URL: https://www.mdpi.com/2079-9292/11/21/3581.

[19] Arif Hakimi et al. "A survey on internet of vehicle (iov): A pplications & comparison of vanets, iov and sdn-iov." In: *ELEKTRIKA-Journal of Electrical Engineering* 20.3 (2021), pp. 26–31.

[20] Gregor A Aramice, Abbas H Miry, and Tariq M Salman. "Vehicle Black Box Implementation For Internet Of Vehicles Based Long Range Technology." In: *Journal of Engineering and Sustainable Development* 27.2 (2023), pp. 245–255. DOI: 10.31272/jeasd.27.2.8.

[21] Haibo Zhou et al. "Evolutionary V2X technologies toward the Internet of vehicles: Challenges and opportunities." In: *Proceedings of the IEEE* 108.2 (2020), pp. 308–323. DOI: 10.1109/JPROC.2019.2961937.

[22] Mustafa İnci, Murat Mustafa Savrun, and Özgür Çelik. "Integrating electric vehicles as virtual power plants: A comprehensive review on vehicle-to-grid (V2G) concepts, interface topologies, marketing and future prospects." In: *Journal of Energy Storage* 55 (2022), p. 105579. DOI: https://doi.org/10.1016/j.est.2022.105579. URL: https://www.sciencedirect.com/science/article/pii/S2352152X22015675.

[23] Kai Li Lim et al. "State of data platforms for connected vehicles and infrastructures." In: *Communications in transportation research* 1 (2021), p. 100013. DOI: https://doi.org/10.1016/j.commtr.2021.100013. URL: https://www.sciencedirect.com/science/article/pii/S2772424721000135.

[24] Jihong Zhao et al. "Dynamic Computing Offloading Strategy for Multi-dimensional Resources Based on MEC." In: *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery*. Ed. by Ning Xiong et al. Cham: Springer International Publishing, 2023, pp. 1290–1303. DOI: https://doi.org/10.1007/978-3-031-20738-9_140.

[25] Zubair Sharif et al. "A Taxonomy for Resource Management in Edge Computing, Applications and Future Realms." In: *2022 International Conference on Digital Transformation and Intelligence (ICDI)*. 2022, pp. 46–52. DOI: 10.1109/ICDI57181.2022.10007397.

[26] Wenjian Lu and Xinglin Zhang. "Computation Offloading for Partitionable Applications in Dense Networks: An Evolutionary Game Approach." In: *IEEE Internet of Things Journal* 9.21 (2022), pp. 20985–20996. DOI: 10.1109/JIOT.2022.3175729.

[27] Yun Chao Hu et al. "Mobile edge computing—A key technology towards 5G." In: *ETSI white paper* 11.11 (2015), pp. 1–16. URL: http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf..

[28] Kai Jiang et al. "Mobile edge computing for ultra-reliable and low-latency communications." In: *IEEE Communications Standards Magazine* 5.2 (2021), pp. 68–75. DOI: 10.1109/MCOMSTD.001.2000045.

[29] Binxu Yang et al. "Cost-efficient NFV-enabled mobile edge-cloud for low latency mobile applications." In: *IEEE Transactions on Network and Service Management* 15.1 (2018), pp. 475–488. DOI: 10.1109/TNSM.2018.2790081.

[30] Ke Zhang et al. "Mobile edge computing and networking for green and low-latency Internet of Things." In: *IEEE Communications Magazine* 56.5 (2018), pp. 39–45. DOI: 10.1109/MCOM.2018.1700882.

[31] Pavel Mach and Zdenek Becvar. "Mobile Edge Computing: A Survey on Architecture and Computation Offloading." In: *IEEE Communications Surveys Tutorials* 19.3 (2017), pp. 1628–1656. DOI: 10.1109/COMST.2017.2682318.

[32] Nasir Abbas et al. "Mobile Edge Computing: A Survey." In: *IEEE Internet of Things Journal* 5.1 (2018), pp. 450–465. DOI: 10.1109/JIOT.2017.2750180.

[33] Quoc-Viet Pham et al. "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art." In: *IEEE access* 8 (2020), pp. 116974–117017. DOI: 10.1109/ACCESS.2020.3001277.

[34] Yuyi Mao et al. "A survey on mobile edge computing: The communication perspective." In: *IEEE communications surveys & tutorials* 19.4 (2017), pp. 2322–2358. DOI: 10.1109/COMST.2017.2745201.

[35] Yushan Siriwardhana et al. "A survey on mobile augmented reality with 5G mobile edge computing: Architectures, applications, and technical aspects." In: *IEEE Communications Surveys & Tutorials* 23.2 (2021), pp. 1160–1192. DOI: 10.1109/COMST.2021.3061981.

[36] Jie Li et al. "Task offloading mechanism based on federated reinforcement learning in mobile edge computing." In: *Digital Communications and Networks* 9.2 (2023), pp. 492–504. DOI: https://doi.org/10.1016/j.dcan.2022.04.006. URL: https://www.sciencedirect.com/science/article/pii/S2352864822000554.

[37] Hao Wu et al. "Wireless powered mobile edge computing for industrial internet of things systems." In: *IEEE Access* 8 (2020), pp. 101539–101549. DOI: 10.1109/ACCESS.2020.2995649.

[38] Liljana Gavrilovska, Valentin Rakovic, and Daniel Denkovski. "Aspects of resource scaling in 5G-MEC: Technologies and opportunities." In: *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE. 2018, pp. 1–6. DOI: 10.1109/GLOCOMW.2018.8644205.

[39] Damian Arellanes and Kung-Kiu Lau. "Evaluating IoT service composition mechanisms for the scalability of IoT systems." In: *Future Generation Computer Systems* 108 (2020), pp. 827–848. DOI: https://doi.org/10.1016/j.future.2020.02.073. URL: https://www.sciencedirect.com/science/article/pii/S0167739X19320291.

[40] Madhusanka Liyanage et al. "Driving forces for multi-access edge computing (MEC) IoT integration in 5G." In: *ICT Express* 7.2 (2021), pp. 127–137. DOI: https://doi.org/10.1016/j.icte.2021.05.007. URL: https://www.sciencedirect.com/science/article/pii/S2405959521000631.

[41] Ashish Singh et al. "Ai-based mobile edge computing for iot: Applications, challenges, and future scope." In: *Arabian Journal for Science and Engineering* 47.8 (2022), pp. 9801–9831. DOI: https://doi.org/10.1007/s13369-021-06348-2.

[42] Wajdi Farhat et al. "A novel cooperative collision avoidance system for vehicular communication based on deep learning." In: *International Journal of Information Technology* 16.3 (2024), pp. 1661–1675.

[43] Tomasz W Nowak et al. "Verticals in 5G MEC-use cases and security challenges." In: *IEEE Access* 9 (2021), pp. 87251–87298.

[44] Francesco Spinelli and Vincenzo Mancuso. "Toward enabled industrial verticals in 5G: A survey on MEC-based approaches to provisioning and flexibility." In: *IEEE Communications Surveys & Tutorials* 23.1 (2020), pp. 596–630.

[45] Dario Sabella et al. "Mobile-edge computing architecture: The role of MEC in the Internet of Things." In: *IEEE Consumer Electronics Magazine* 5.4 (2016), pp. 84–91.

[46] Francesco Giannone et al. "Orchestrating heterogeneous MEC-based applications for connected vehicles." In: *Computer Networks* 180 (2020), p. 107402.

[47] Madhusanka Liyanage et al. "Driving forces for Multi-Access Edge Computing (MEC) IoT integration in 5G." In: *ICT Express* 7.2 (2021), pp. 127–137. DOI: https://doi.org/10.1016/j.icte.2021.05.007. URL: https://www.sciencedirect.com/science/article/pii/S2405959521000631.

[48] Y. C. Hu et al. *ETSI White Paper #11 Mobile edge computing - A key technology towards 5G*. ETSI White Pap. No. 11 Mob. [Online]. Available: http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf. 2015. URL: http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf.

[49] M. Becker, S. Lehrig, and S. Becker. "Systematically deriving quality metrics for cloud computing systems." In: *ICPE 2015 - Proc. 6th ACM/SPEC Int. Conf. Perform. Eng.* 2015, pp. 169–174. DOI: 10.1145/2668930.2688043.

[50] Sebastian Lehrig and Matthias Becker. "Approaching the Cloud: Using Palladio for Scalability, Elasticity, and Efficiency Analyses." In: *SoSP*. 2014, pp. 141–151.

[51] Juyong Lee, Jeong-Weon Kim, and Jihoon Lee. "Mobile personal multi-access edge computing architecture composed of individual user devices." In: *Applied Sciences* 10.13 (2020), p. 4643.

[52] J. Zeng et al. "Mobile edge communications, computing, and caching (mec3) technology in the maritime communication network." In: *China Commun.* 17.5 (2020), pp. 223–234. DOI: 10.23919/JCC.2020.05.017.

[53] H. Mei, K. Wang, and K. Yang. "Joint cache content placement and task offloading in C-RAN enabled by multi-layer MEC." In: *Sensors (Switzerland)* 18.6 (2018), pp. 1–21. DOI: 10.3390/s18061826.

[54] Meng Qin et al. "Service-oriented energy-latency tradeoff for IoT task partial offloading in MEC-enhanced multi-RAT networks." In: *IEEE Internet of Things Journal* 8.3 (2020), pp. 1896–1907.

[55] Dawei Chen et al. "Matching-theory-based low-latency scheme for multitask federated learning in MEC networks." In: *IEEE Internet of Things Journal* 8.14 (2021), pp. 11415–11426.

[56] Dawei Chen et al. "Fedsvrg based communication efficient scheme for federated learning in mec networks." In: *IEEE Transactions on Vehicular Technology* 70.7 (2021), pp. 7300–7304.

[57] Shanhe Yi et al. "Lavea: Latency-aware video analytics on edge computing platform." In: *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*. 2017, pp. 1–13. DOI: 10.1109/ICDCS.2017.182.

[58] M. S. Ansari et al. "Security of distributed intelligence in edge computing: threats and countermeasures." In: *Palgrave Stud. Digit. Bus. Enabling Technol.* (2020), pp. 95–122. DOI: 10.1007/978-3-030-41110-7_6.

[59] Y. Tamura, M. Kawakami, and S. Yamada. "Reliability, modeling and analysis for open source cloud computing." In: *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.* 227.2 (2013), pp. 179–186. DOI: 10.1177/1748006X12475110.

[60] M. Bukhsh, S. Abdullah, and I. S. Bajwa. "A decentralized edge computing latency-aware task management method with high availability for IoT applications." In: *IEEE Access* 9 (2021), pp. 138994–139008. DOI: 10.1109/ACCESS.2021.3116717.

[61] A. Abouaomar et al. "Resource provisioning in edge computing for latency-sensitive applications." In: *IEEE Internet Things J.* 8.14 (2021), pp. 11088–11099. DOI: 10.1109/JIOT.2021.3052082.

[62] Richard Cziva, Christos Anagnostopoulos, and Dimitrios P Pezaros. "Dynamic, latency-optimal vNF placement at the network edge." In: *Ieee infocom 2018-ieee conference on computer communications*. IEEE. 2018, pp. 693–701. DOI: 10.1109/INFOCOM.2018.8486021.

[63] A. Zhou et al. "LMM: latency-aware micro-service mashup in mobile edge computing environment." In: *Neural Comput. Appl.* 32.19 (2020), pp. 15411–15425. DOI: 10.1007/s00521-019-04693-w.

[64] R. Kemp et al. "Cuckoo: A computation offloading framework for smartphones." In: *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST*. Vol. 76. 2012, pp. 59–79. DOI: 10.1007/978-3-642-29336-8_4.

[65] Y. Tian, J. Tian, and N. Li. "Cloud reliability and efficiency improvement via failure risk based proactive actions." In: *J. Syst. Softw.* 163 (2020), p. 110524. DOI: 10.1016/j.jss.2020.110524.

[66] D. Chicco, N. Tötsch, and G. Jurman. "The matthews correlation coefficient (Mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation." In: *BioData Min.* 14 (2021), pp. 1–22. DOI: 10.1186/s13040-021-00244-z.

[67] P. Angin, B. K. Bhargava, et al. "Real-time mobile-cloud computing for context-aware blind navigation." In: *Int. J. Next-Generation Comput.* 2.2 (2011). [Online]. Available: http://www.ijngc.perpetualinnovation.net/index.php/ijngc/article/view/89, pp. 1–13.

[68] B. G. Chun et al. "CloneCloud: Elastic execution between mobile device and cloud." In: *EuroSys'11 - Proc. EuroSys 2011 Conf.* 2011, pp. 301–314. DOI: 10.1145/1966445.1966473.

[69] T. D. Nguyen, M. Van Nguyen, and E. N. Huh. "Service image placement for thin client in mobile cloud computing." In: *Proc. - 2012 IEEE 5th Int. Conf. Cloud Comput. CLOUD 2012*. 2012, pp. 416–422. DOI: 10.1109/CLOUD.2012.39.

[70] W. Ren et al. "Lightweight and compromise resilient storage outsourcing with distributed secure accessibility in mobile cloud computing." In: *Tsinghua Sci. Technol.* 16.5 (2011), pp. 520–528. DOI: 10.1016/S1007-0214(11)70070-0.

[71] M. Merluzzi et al. "Dynamic computation offloading in multi-access edge computing via ultra-reliable and low-latency communications." In: *IEEE Trans. Signal Inf. Process. over Networks* 6 (2020), pp. 342–356. DOI: 10.1109/TSIPN.2020.2981266.

[72] A. B. M. B. Alam, A. Haque, and M. Zulkernine. "CREM: A cloud reliability evaluation model." In: *2018 IEEE Global Communications Conference (GLOBECOM)*. 2018, pp. 1–6. DOI: 10.1109/GLOCOM.2018.864748.

[73] N. Mallikharjuna, C. S.-, and V. Satyendra. "Cloud computing through mobile-learning." In: *Int. J. Adv. Comput. Sci. Appl.* 1.6 (2010). DOI: 10.14569/ijacsa.2010.010607.

[74] L. Chen et al. "REMR: A reliability evaluation method for dynamic edge computing network under time constraints." In: *arXiv* 14.8 (2021), pp. 1–9. URL: http://arxiv.org/abs/2112.01913.

[75] D. Huang et al. "MobiCloud: Building secure cloud framework for mobile computing and communication." In: *Proc. - 5th IEEE Int. Symp. Serv. Syst. Eng. SOSE 2010*. 2010, pp. 27–34. DOI: 10.1109/SOSE.2010.20.

[76] L. Miao et al. "Mean field games theoretic for mobile privacy security enhancement in edge computing." In: *Wirel. Pers. Commun.* 111.3 (2020), pp. 2045–2063. DOI: 10.1007/s11277-019-06971-1.

[77] A. Wheeldon et al. "Low-latency asynchronous logic design for inference at the edge." In: *Proc. -Design, Autom. Test Eur. DATE*. Vol. 2021-February. 2021, pp. 370–373. DOI: 10.23919/DATE51398.2021.9474126.

[78] N. Vance et al. "Towards reliability in online high-churn edge computing: A deviceless pipelining approach." In: *Proc. - 2019 IEEE Int. Conf. Smart Comput. SMARTCOMP 2019*. 2019, pp. 301–308. DOI: 10.1109/SMARTCOMP.2019.00066.

[79] I. Farris et al. "Federations of connected things for delay-sensitive IoT services in 5G environments." In: *IEEE Int. Conf. Commun.* 2017, pp. 1–6. DOI: 10.1109/ICC.2017.7996644.

[80] S. Battula et al. "Online ocean monitoring using edge IoT." In: *2020 Glob. Ocean. 2020 Singapore - U.S. Gulf Coast*. 2020. DOI: 10.1109/IEEECONF38699.2020.9389430.

[81] X. Wang et al. "In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning." In: *IEEE Netw.* 33.5 (2019), pp. 156–165. DOI: 10.1109/MNET.2019.1800286.

[82] D. Zhao et al. "A service migration strategy based on multiple attribute decision in mobile edge computing." In: *Int. Conf. Commun. Technol. Proceedings, ICCT*. Vol. 2017-October. 2018, pp. 986–990. DOI: 10.1109/ICCT.2017.8359782.

[83] Seyedeh Shabnam Jazaeri et al. "Toward caching techniques in edge computing over SDN-IoT architecture: A review of challenges, solutions, and open issues." In: *Multimedia Tools and Applications* 83.1 (2024), pp. 1311–1377.

[84] Lucas Bréhon–Grataloup, Rahim Kacimi, and André-Luc Beylot. "Mobile edge computing for V2X architectures and applications: A survey." In: *Computer Networks* 206 (2022), p. 108797.

[85] Zhaolong Ning et al. "Intelligent edge computing in internet of vehicles: A joint computation offloading and caching solution." In: *IEEE Transactions on Intelligent Transportation Systems* 22.4 (2020), pp. 2212–2225.

[86] Li Chunlin and Jing Zhang. "Dynamic cooperative caching strategy for delay-sensitive applications in edge computing environment." In: *The Journal of Supercomputing* 76.10 (2020), pp. 7594–7618.

[87] Salahadin Seid Musa et al. "Convergence of information-centric networks and edge intelligence for IoV: Challenges and future directions." In: *Future Internet* 14.7 (2022), p. 192.

[88] Chenmeng Wang et al. "Integration of networking, caching, and computing in wireless systems: A survey, some research issues, and challenges." In: *IEEE Communications Surveys & Tutorials* 20.1 (2017), pp. 7–38.

[89] Minrui Xu et al. "Unleashing the Power of Edge-Cloud Generative AI in Mobile Networks: A Survey of AIGC Services." In: *IEEE Communications Surveys Tutorials* (2024), pp. 1–1. DOI: 10.1109/COMST.2024.3353265.

[90] Ansif Arooj et al. "Big data processing and analysis in internet of vehicles: architecture, taxonomy, and open research challenges." In: *Archives of Computational Methods in Engineering* 29.2 (2022), pp. 793–829.

[91] Zhang Degan et al. "A content distribution method of internet of vehicles based on edge cache and immune cloning strategy." In: *Ad Hoc Networks* 138 (2023), p. 103012.

[92] Quadri Noorulhasan Naveed et al. "An intelligent traffic surveillance system using integrated wireless sensor network and improved phase timing optimization." In: *Sensors* 22.9 (2022), p. 3333.

[93] Xiaoge Huang et al. "Delay-Aware Caching in Internet-of-Vehicles Networks." In: *IEEE Internet of Things Journal* 8.13 (2021), pp. 10911–10921. DOI: 10.1109/JIOT.2021.3051290.

[94] David Naseh, Swapnil Sadashiv Shinde, and Daniele Tarchi. "Network Sliced Distributed Learning-as-a-Service for Internet of Vehicles Applications in 6G Non-Terrestrial Network Scenarios." In: *Journal of Sensor and Actuator Networks* 13.1 (2024), p. 14.

[95] Manzoor Ahmed et al. "Vehicular Communication Network Enabled CAV Data Offloading: A Review." In: *IEEE Transactions on Intelligent Transportation Systems* 24.8 (2023), pp. 7869–7897. DOI: 10.1109/TITS.2023.3263643.

[96] Sedeng Danba et al. "Toward collaborative intelligence in IoV systems: Recent advances and open issues." In: *Sensors* 22.18 (2022), p. 6995.

[97] "UAV-based Internet of Vehicles: A systematic literature review." In: *Intelligent Systems with Applications* 18 (2023), p. 200226. DOI: https://doi.org/10.1016/j.iswa.2023.200226.

[98] Kai Jiang et al. "Asynchronous Federated and Reinforcement Learning for Mobility-Aware Edge Caching in IoV." In: *IEEE Internet of Things Journal* 11.9 (2024), pp. 15334–15347. DOI: 10.1109/JIOT.2023.3349255.

[99] Rafat Aghazadeh, Ali Shahidinejad, and Mostafa Ghobaei-Arani. "Proactive content caching in edge computing environment: A review." In: *Software: Practice and Experience* 53.3 (2023), pp. 811–855.

[100] Sindhu Padakandla. "A survey of reinforcement learning algorithms for dynamically varying environments." In: *ACM Computing Surveys (CSUR)* 54.6 (2021), pp. 1–25.

[101] Khadija Shaheen et al. "Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks." In: *Journal of Intelligent & Robotic Systems* 105.1 (2022), p. 9.

[102] SM Ahsan Kazmi et al. "Computing on wheels: A deep reinforcement learning-based approach." In: *IEEE Transactions on Intelligent Transportation Systems* 23.11 (2022), pp. 22535–22548.

[103] J. D. Co-Reyes et al. "Evolving reinforcement learning algorithms." In: *arXiv preprint arXiv:2101.03958* (2021).

[104] W. Jiang et al. "Multi-Agent Reinforcement Learning for Efficient Content Caching in Mobile D2D Networks." In: *IEEE Transactions on Wireless Communications* 18.3 (Mar. 2019), pp. 1610–1622. DOI: 10.1109/TWC.2019.2894403.

[105] Z. Yang et al. "Learning Automata Based Q-learning for Content Placement in Cooperative Caching." In: *IEEE Transactions on Communications* 68.6 (June 2020), pp. 3667–3680. DOI: 10.1109/TCOMM.2020.2982136.

[106] X. Fang et al. "Multi-agent Cooperative Alternating Q-learning Caching in D2D-enabled Cellular Networks." In: *IEEE Global Communications Conference (GLOBECOM)*. HI, USA, Dec. 2019, pp. 1–6. DOI: 10.1109/GLOBECOM38437.2019.9014053.

[107] Y. Qian et al. "Reinforcement Learning-based Optimal Computing and Caching in Mobile Edge Network." In: *IEEE Journal on Selected Areas in Communications* 38.10 (Oct. 2020), pp. 2343–2355. DOI: 10.1109/JSAC.2020.3000396.

[108] P. Liu et al. "Intelligent Mobile Edge Caching for Popular Contents in Vehicular Cloud Toward 6G." In: *IEEE Transactions on Vehicular Technology* 70.6 (June 2021), pp. 5265–5274. DOI: 10.1109/TVT.2021.3076304.

[109] W. Qi et al. "Extensive Edge Intelligence for Future Vehicular Networks in 6G." In: *IEEE Wireless Communications* 28.4 (Aug. 2021), pp. 128–135. DOI: 10.1109/MWC.001.2000393.

[110] J. Shi et al. "A Novel Deep Q-Learning-Based Air-Assisted Vehicular Caching Scheme for Safe Autonomous Driving." In: *IEEE Transactions on Intelligent Transportation Systems* 22.7 (July 2021), pp. 4348–4358. DOI: 10.1109/TITS.2020.3018720.

[111] Z. Zhang et al. "Smart Proactive Caching: Empower the Video Delivery for Autonomous Vehicles in ICN-Based Networks." In: *IEEE Transactions on Vehicular Technology* 69.7 (July 2020), pp. 7955–7965. DOI: 10.1109/TVT.2020.2994181.

[112] Y. Liu and B. Mao. "On a Novel Content Edge Caching Approach Based on Multi-Agent Federated Reinforcement Learning in Internet of Vehicles." In: *32nd Wireless and Optical Communications Conference (WOCC)*. Newark, NJ, USA, 2023, pp. 1–5. DOI: 10.1109/WOCC58016.2023.10139417.

[113] A. Ndikumana et al. "Deep Learning Based Caching for Self-Driving Cars in Multi-Access Edge Computing." In: *IEEE Transactions on Intelligent Transportation Systems* 22.5 (May 2021), pp. 2862–2877. DOI: 10.1109/TITS.2020.2976572.

[114] Z. Zhu et al. "Proactive Caching in Auto Driving Scene via Deep Reinforcement Learning." In: *11th International Conference on Wireless Communications and Signal Processing (WCSP)*. Xi'an, China, Oct. 2019, pp. 1–6. DOI: 10.1109/WCSP.2019.8928131.

[115] A. Ndikumana and C. S. Hong. "Self-Driving Car Meets Multi-Access Edge Computing for Deep Learning-Based Caching." In: *International Conference on Information Networking (ICOIN)*. Kuala Lumpur, Malaysia, Jan. 2019, pp. 49–54. DOI: 10.1109/ICOIN.2019.8718113.

[116] W. Jiang et al. "Learning-Based Cooperative Content Caching Policy for Mobile Edge Computing." In: *IEEE International Conference on Communications (ICC)*. Shanghai, China, May 2019, pp. 1–6. DOI: 10.1109/ICC.2019.8761121.

[117] C. Zhang et al. "Toward Edge-Assisted Video Content Intelligent Caching with Long Short-Term Memory Learning." In: *IEEE Access* 7 (2019), pp. 152832–152846. DOI: 10.1109/ACCESS.2019.2947067.

[118] Y. Ye, M. Xiao, and M. Skoglund. "Mobility-aware Content Preference Learning in Decentralized Caching Networks." In: *IEEE Transactions on Cognitive Communications and Networking* 6.1 (Mar. 2020), pp. 62–73. DOI: 10.1109/TCCN.2019.2937519.

[119] Y. Zhang et al. "Cooperative Edge Caching: A Multi-agent Deep Learning Based Approach." In: *IEEE Access* 8 (2020), pp. 133212–133224. DOI: 10.1109/ACCESS.2020.3010329.

[120] X. Xu, M. Tao, and C. Shen. "Collaborative Multi-agent Multi-armed Bandit Learning for Small-cell Caching." In: *IEEE Transactions on Wireless Communications* 19.4 (Apr. 2020), pp. 2570–2585. DOI: 10.1109/TWC.2020.2966599.

[121] Ruyan Wang et al. "Cooperative Caching Strategy With Content Request Prediction in Internet of Vehicles." In: *IEEE Internet of Things Journal* 8.11 (2021), pp. 8964–8975. DOI: 10.1109/JIOT.2021.3056084.

[122] X. Bi and L. Zhao. "Collaborative Caching Strategy for RL-Based Content Downloading Algorithm in Clustered Vehicular Networks." In: *IEEE Internet of Things Journal* 10.11 (June 2023), pp. 9585–9596. DOI: 10.1109/JIOT.2023.3235661.

[123] Gianni Barlacchi, Marco De Nadai, Roberto Larcher, et al. "A Multi-source Dataset of Urban Life in the City of Milan and the Province of Trentino." In: *Scientific Data* 2 (2015), p. 150055. DOI: 10.1038/sdata.2015.55. URL: https://doi.org/10.1038/sdata.2015.55.

[124] C. Zhang et al. "Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data." In: *IEEE Journal on Selected Areas in Communications* 37.6 (June 2019), pp. 1389–1401.

[125] C. Wang et al. "Deep learning-based intelligent dual connectivity for mobility management in dense network." In: *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. Chicago, IL, USA, Aug. 2018, pp. 1–5. DOI: 10.1109/VTCFall.2018.8690554.

[126] Degan Zhang et al. "New Multi-Hop Clustering Algorithm for Vehicular Ad Hoc Networks." In: *IEEE Transactions on Intelligent Transportation Systems* 20.4 (2019), pp. 1517–1530. DOI: 10.1109/TITS.2018.2853165.

[127] Y. Cui, Y. Liang, and R. Wang. "Resource Allocation Algorithm with Multi-Platform Intelligent Offloading in D2D-Enabled Vehicular Networks." In: *IEEE Access* 7 (2019), pp. 21246–21253. DOI: 10.1109/ACCESS.2018.2882000. URL: https://doi.org/10.1109/ACCESS.2018.2882000.

[128] S. Agrawal and N. Goyal. "Thompson Sampling for Contextual Bandits with Linear Payoffs." In: *CoRR* (2012). arXiv: 1209.3352 [cs.LG]. URL: https://arxiv.org/abs/1209.3352.

[129] Y. Chen et al. "Prefetch and Cache Replacement Based on Thompson Sampling for Satellite IoT Network." In: *ICC 2021 - IEEE International Conference on Communications*. Montreal, QC, Canada, 2021, pp. 1–6. DOI: 10.1109/ICC42927.2021.9500508. URL: https://doi.org/10.1109/ICC42927.2021.9500508.

[130] L. Zhao et al. "Intelligent Content Caching Strategy in Autonomous Driving Toward 6G." In: *IEEE Transactions on Intelligent Transportation Systems* 23.7 (July 2022), pp. 9786–9796. DOI: 10.1109/TITS.2021.3114199. URL: https://doi.org/10.1109/TITS.2021.3114199.