AI empowered content caching in vehicular edge computing: opportunities and challenges

Muhammad Awais Javed* and Sherali Zeadally †
*COMSATS University Islamabad, Pakistan
†University of Kentucky, United States
{awais.javed@comsats.edu.pk, szeadally@uky.edu}

***Abstract*—Vehicular networks are an indispensable component of future autonomous and intelligent transport systems. Today, many vehicular networking applications are emerging, and therefore, efficient data computation, storage, and retrieval solutions are needed. Vehicular Edge Computing (VEC) is a promising technique that uses Road Side Units (RSUs) to act as edge servers for caching and task offloading purposes. We present a task-based architecture of content caching in VEC, where three major tasks are identified namely, content popularity prediction, content placement in the cache, and content retrieval from the cache. We present an overview of how Artificial Intelligence (AI) techniques such as regression and Deep Q-learning can improve the efficiency of these tasks. We also highlight related future research opportunities in the areas such as collaborative data sharing for improved caching, efficient sub-channel allocation for content retrieval in C-V2X, and secure caching.**

***Index Terms*—Internet of Vehicles, vehicular networks, artificial intelligence, caching, edge computing.**

I. Introduction

Internet of Vehicles (IoV) is a key technology that enables several applications for future traffic management. With reliable wireless communications among vehicles themselves along with the infrastructure Road Side Units (RSUs), autonomous driving and intelligent traffic congestion control can be realized [1]. Moreover, several infotainment and entertainment applications such as advertisements by companies, content sharing by service providers, and multimedia streaming can be developed using an IoV communication architecture.

As innovative vehicular applications continue to emerge, big data will be generated and it will need to be disseminated among various IoV nodes (vehicles and RSUs). Similarly, several tasks such as vehicle safety decisions, traffic route calculation, content downloading and so on will need to be performed by the IoV nodes. All these new tasks will need more computation and storage. Cloud computing is one possible solution for caching popular and urgent data, and for offloading computationally intensive tasks. Since cloud servers are located far away from the vehicles, storage and computation may not be efficient due to the long geographical distance which results in higher latency (to retrieve contents from the cloud cache or sending task offloading requests) and wastes communication resources (for long distance communication between the vehicles and the cloud) [2].

In this context, Vehicle Edge Computing (VEC) is a promising technique wherein RSUs act as edge servers providing computing and storage services closer to the vehicles. Using VEC, data can be processed quicker which is key for IoV applications and as a result decisions can be made faster and the latency can be reduced [3]. Higher processing delays can result in expiration of the data which can lead to wrong inferences. For instance, if a vehicle offloads a safety task to a server (by sending neighborhood data) which takes a long time to process the data, the safety data may no longer be relevant at that time and could issue the wrong instruction to the vehicle.

Other advantages of VEC include reduced communication bandwidth because data is disseminated to servers near to the vehicles for processing. It is critical to reserve a vehicular channel for application data sharing rather than using it for sharing computing and storage messages. Distributed edge servers can also provide reliable computation and storage as compared to central cloud computing. Moreover, edge servers along with cloud servers, provide a scalable solution when the number of vehicular applications increases. Based on the application type and delay requirements, data can be offloaded to either the edge or the cloud servers [4].

This work focuses on the caching service provided by VEC. Efficient caching and RSU storage management are major challenges in dynamic vehicular networks. Fast mobility and short-lived connectivity in vehicular network make it harder to rank the contents in terms of their popularity, and therefore, storage needs to be periodically updated with the relevant contents, and in this case, an optimal resource allocation strategy is needed to enable download of contents from the cache.

Although a technique such as matching theory [5] has been used for effective content caching in VEC, Artificial Intelligence (AI) techniques can collect and analyze different types of data to help make better informed decisions regarding caching. Data in vehicular networks has temporal and location variability. Hence, static algorithms may not work well in vehicular networks. Learning techniques can help gain useful insights into the data and use data caching to reduce application latency and improve application reliability.

Research contributions of this work

We summarize the research contributions of this work as follows:

- We present an overview of a task-based architecture for content caching in VEC.
- We describe how AI techniques such as regression, reinforcement learning and deep Q-learning can improve the

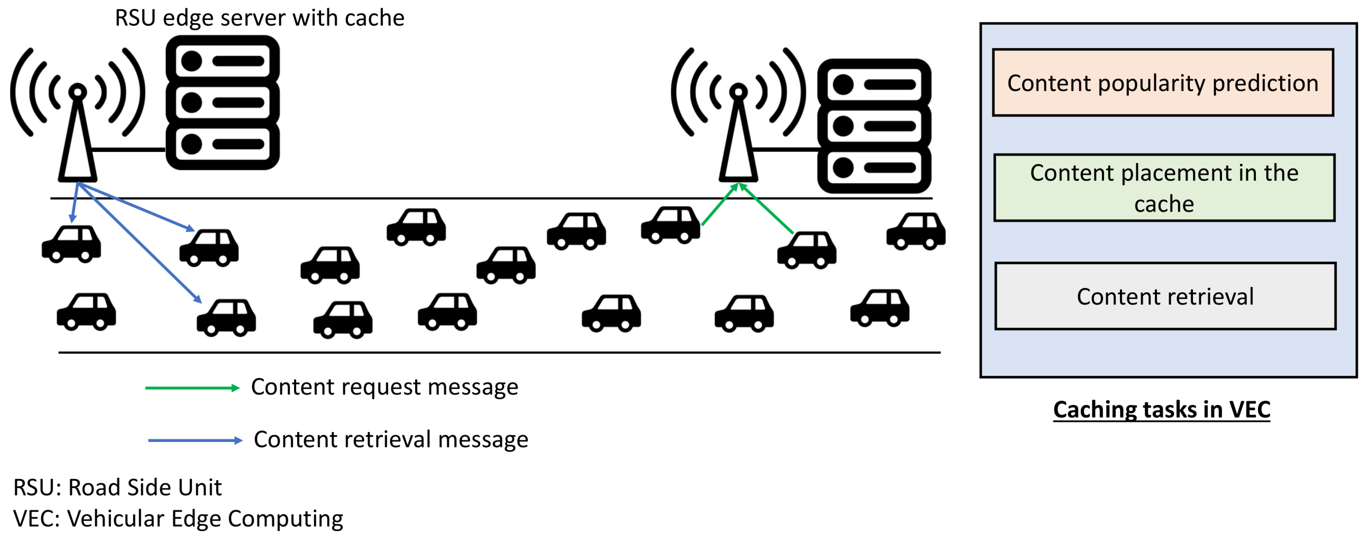

Fig. 1: Task-based caching in VEC architecture.

caching process in VEC.

- We discuss several open challenges and future opportunities for using AI for efficient caching in VEC.

II. Caching in Vehicular Edge Computing

Fig. 1 presents a vehicular edge computing architecture where vehicles communicate with RSUs for storage and computation services. Each RSU has a limited storage space for caching contents. Vehicle can send content requests to the RSUs using cellular V2X communications. If the content is in the cache of the RSU, it can send the content to the vehicles, otherwise, vehicles have to download the content from the cloud server.

Content caching in IoV places popular contents into the storage of the RSU. We divide content caching into three major tasks. The first task is the content popularity prediction where RSUs estimate and rank the contents in terms of their demand. As the RSUs have finite storage space and contents compete to be placed in the cache, an intelligent allocation is required. RSUs can evaluate the content popularity based on the requests generated by the vehicles or forecast it using learning techniques.

The second task for caching in VEC is the content placement in the cache. Once the data is ranked in terms of popularity, it can be placed in the RSUs storage blocks. However, decisions such as whether to cache the content or which RSU to choose for placing the content are critical. For this purpose, optimization algorithms such as knapsack and matching theory have traditionally been used [6]. However, the time-varying nature of vehicular data makes learning algorithms well-suited to perform this task.

Content retrieval from the cache is the last task for caching in VEC. Since most contents will be placed in the cache and vehicles will download them directly from the RSUs, such an approach puts a burden on the RSU to vehicles link. The communication resources need to be intelligently allocated because safety data is also shared over the same channel. Two major wireless technologies could be used to retrieve contents from the cache, namely, IEEE 802.11p and Cellular V2X (C-V2X) [1]. IEEE 802.11p is a variant of Wi-Fi technology and C-V2X uses 5G cellular communications.

Content retrieval can be a challenging task as the network is normally shared with many types of data traffic. Issues such as data rate reduction, higher packet inter-arrival times and packet losses may occur when the network load increases. Efficient resource allocation techniques are needed to address channel overloading. In IEEE 802.11p, a multi-channel operation may be used in this context, isolating safety and non-safety data and placing them on different channels. In this case, content retrieval can be assigned one particular non-safety channel. Other possible solutions include medium access control techniques which focus on providing delay guarantees, load balancing, and fairness. In contrast, C-V2X divides the resources into frequency and time units known as sub-channels. Here efficient scheduling techniques are needed which can provide low content retrieval times while not affecting the safety data transmissions. Traditional resource allocation techniques such as maximum throughput or fairness based scheduling may not work well in this scenario because they do not consider different types of data traffic. Therefore, learning techniques are useful in scheduling the data adaptively and intelligently.

Several types of IoV data can be placed into cache. This includes safety data such as a scenario where a group of vehicles detect an accident or emergency situation and data (such as information about the accident, image of the accident and video of the accident situation) can be stored in the nearby RSU cache. This data is then disseminated to the vehicles moving toward the accident place and also shared with the nearby RSUs. However, most of the safety data and safety tasks are stored and computed at the vehicles due to time delay requirements. Vehicles have to periodically share cooperative awareness messages within a time delay of 100 milliseconds. Based on these periodic messages, vehicles develop a Local Dynamic Map (LDM) and make driving decisions such as lane

TABLE I: Overview of learning techniques used for tasks in VEC

MEC Task	Problem	Learning technique used	Key idea
Content popularity prediction	Time varying content popularity prediction [7]	Autoregression	Forecast future content popularity based on past values
	Location-aware content popularity prediction [8]	Ridge regression	Consider location attributes while predicting content popularity. Add a penalty to overcome the noise in the content demand information and provide stable estimation
	Location-aware and preference-based content popularity prediction [9]	Logistic regression	Capture user preference and region-based content popularity
	Time varying content popularity prediction [10]	Regression using Constrained Non-Negative Least Squares (CNNLS) approach and Follow the Leader (FTL) online learning technique	Use online optimization and online learning to predict the content in a computationally efficient manner
Content placement	Reduce latency of content caching [3]	Heuristic Q-learning	Use Long Short-Term Memory (LSTM) network to predict mobility of vehicles. Q-learning with greedy search is used to optimally select caching strategy
	Efficient content placement and task computation [4]	Deep Q-learning with multi-timescale	Two Deep Q-learning models at two different time scales and mobility-aware reward estimation to reduce complexity when a large action space exists
	Collaborative data scheduling to reduce cost and meet delay constraints [11]	Deep Q-learning	Data scheduling decision based on whether to cache data at the RSU or other vehicles
	Joint content caching and computation for continuous action space [2]	Deep Deterministic Policy Gradient (DDPG)	Actor-critic algorithm that uses actor network for policy learning and critic network for policy evaluation
Content retrieval	Low content access latency and improve transmission fairness [12]	Q-learning	Use Q-learning for adaptive contention window based Medium Access Control (MAC) protocol
	Improve content throughput and reduce the packet loss [13]	Q-learning	Use network parameters to find optimal Uplink (UL)/Downlink (DL) ratio in Time Division Duplexing (TDD) based 5G vehicular networks
	Maximize resource utilization in network slicing based content retrieval [14]	Off-line Q-learning	Optimally allocate radio resources on each network slice

changing and applying sudden brakes.

Infotainment data will be a major part of cache storage as such data will be in demand and be requested by many vehicle users. Moreover, content providers will also be interested in hosting their data in the cache storage for user convenience. This data will include large multimedia files such as movies and advertisements. With the movement of vehicles, the location of caching such data may need to be changed periodically. One solution is to cache only certain chunks of infotainment data in a RSU and store the other chunks on the neighboring RSUs depending on vehicles speed and direction it is moving.

Another type of data that can be placed in the cache is the information related to the communication protocol parameters. Vehicular networks different types of protocols at the network, link, and physical layers. Some of these protocols have specific location and time dependent parameters or variables that need to be disseminated to all the vehicles for optimal operation of the protocols. For example, transmission power control protocols may require knowledge of current channel load or network density data. RSUs can store the data relevant to its geographical region, periodically update it and send it to all incoming vehicles so that transmission power control protocol can provide optimal throughput. Similarly, other physical layer protocols such as congestion control, data rate adaptation, and sub-channel selection can also cache their important control parameters.

Similarly, at the link layer, several medium access protocols rely on clustering techniques. This requires the selection of cluster size and cluster head vehicle. Other variables for link layer protocols may include information about time slots and sensing time. At the network layer, information about routing algorithms can be stored in the cache. So, instead of distributing the collection of protocol data (parameters and variables) by the vehicles, RSUs can better collect and share this information.

III. AI EMPOWERED CACHING IN VEC

In this section, we describe how AI can be helpful for caching in VEC. As described in the last section, caching in VEC can be divided into three major tasks. Table I presents how AI can be used to efficiently perform each of these tasks.

A. *Content popularity prediction*

Content popularity is a key metric for placing contents into the cache storage in edge servers. Content popularity directly relates to the content in demand which will be frequently requested for download by several vehicles. Placing popular content into the cache yields several advantages such as optimal use of limited cache storage, quick retrieval of content at the vehicles with a higher data rate, and reduced load on the communication channel between the RSU and the vehicle.

Content popularity is generally calculated based on the number of requests by the vehicles for a particular content. Moreover, content popularity is often assumed to follow the

TABLE II: States, actions and reward in reinforcement learning techniques used for content placement

RL technique	States	Actions	Reward
Heuristic Q-learning [3]	RSU cache occupation	Cache or not to Cache, selection of RSU based on LSTM network	Content retrieval time
Deep Q-learning [4]	Signal-to-Noise Ratio (SNR) between vehicle and RSU, contact frequency, and contact time between vehicles and RSUs	Cache or not to cache, number of coded packets to be cached, computation task to be offloaded or not	Cost of communication, computation and caching
Deep Q-learning [11]	Amount of data cached in the queue	Data scheduling	Data scheduling loss
Deep Deterministic Policy Gradient (DDPG) [2]	State of vehicle, computation resources, caching resources, bandwidth of server	Amount of caching and computation resources, bandwidth allocated by RSU to vehicles	Computing utility, caching utility, energy consumption cost

Zipf distribution [6]. A major challenge in modeling content popularity in IoV is that it varies with time and location. As vehicular network is highly dynamic with vehicles constantly moving at a high speed, evaluating content popularity in real time is a cumbersome task. Even within a time span of few minutes, the content popularity within the coverage range of an edge server (placed in the RSU) may change. Similarly, each edge server may have a different content popularity within its geographical range.

It is vital to accurately predict the content popularity for efficiently performing the caching in MEC based IoV. This prediction may use past history of requests for a particular content as well as the current demand for the content. Online learning techniques are mostly used for content popularity prediction because they do not require a training phase and are suitable in predicting the time-varying nature of content popularity. Moreover, online learning techniques require less data storage and are computationally less intensive as compared to offline learning.

Regression is a key learning technique that is efficiently used to predict the content popularity. For example, in [7], an autoregressive model is used to forecast the future content popularity based on past content demand. The popularity is based on the contents popularity history of the past 15 cycles. In [8], content popularity is predicted by taking into account the content and the location attributes. Using the content demand information in the previous time slot, a ridge regression algorithm is proposed to evaluate the content popularity in the current time slot. The rationale behind using ridge regression is to mitigate the random noise in the required information. This is achieved by adding a suitable penalty which results in a stable estimation.

In [9], the authors propose a logistic regression based algorithm to predict content popularity (value between 0 and 1) based on the location and the user preferences. Another online learning and content popularity prediction is proposed in [10] which predicts the popularity of contents at a given time based on past content popularities. The regression problem is solved using the Constrained Non-Negative Least Squares (CNNLS) approach. To reduce the computational complexity of the algorithm by the authors of [10], the authors further propose the use of an online learning technique such as Follow The Leader (FTL).

B. Content placement in the cache

Content placement in the cache storage is the second task to be performed for caching in VEC. As the amount of storage space available at the edge servers for caching is limited, it is critical to optimally utilize it. AI can play a vital role in intelligently placing contents in the cache by collecting, predicting, and processing data such as content popularity, content expiry time, vehicle mobility, cost of storage and link conditions.

Reinforcement Learning (RL) is a promising AI technique that has been frequently used in the literature to optimize content caching. RL is a branch of machine learning wherein an agent learns to take optimal actions when interacting with an environment. The agent improves its learning experience based on repeated observations of taking actions in a given state and computing rewards. The goal of RL technique is to find an optimal strategy that maximizes the reward. There are two types of RL, model-based and model-free. In model-based RL, an agent aims to use or learn the model (a model here refers to a reward function and state transition probabilities). On the other hand, in model free RL, an agent makes decision based on experience through repeated observations without learning the model. While model-based RL techniques are more efficient because agent strategies can be planned based on the model, developing an accurate model in a dynamic environment is a challenging task. Therefore, model-free RL techniques such as Q-learning, are more suitable for VEC. Table II presents states, actions, and rewards for different RL techniques in the literature.

In [3], the authors used a heuristic Q-learning technique with greedy search for content placement in the cache. Long-Short Term Memory (LSTM) based neural network is used to predict the mobility of vehicles from the traffic data. Here, the RSU cache occupation (number of units of cache that are occupied) is taken as the state, the decision to cache (if yes, at which RSU) or not to cache is the action. LSTM based mobility prediction is used to find the best action. Moreover, the reward used by the authors [3] is the time required by the vehicles to get the required content.

While simple RL techniques are useful when state space and action space are small, they fail to provide an optimal solution otherwise. Therefore, Deep RL techniques such Deep Q-Learning (DQL) are more useful for VEC. In DQL algorithms,

TABLE III: States, Actions and Reward in Reinforcement learning techniques used for content retrieval

RL technique	States	Actions	Rewards
Q-learning [12]	Contention Window (CW) value	Packet transmission	+1 if vehicle chooses most commonly used CW within first hop neighbors. Reduced reward if vehicle chooses a different CW
Q-learning [13]	Percentage of UL/DL data rate against channel capacity	UL/DL ratio	UL and DL data rates
Off-line reinforcement learning [14]	Single state	Slicing ratios	Resource utilization

the agent stores its experiences (i.e., states, action, rewards) in a replay memory which can be used to train a Deep Neural Network (DNN). As compared to Q-learning, DQL uses both past and current experiences to efficiently train the DNN. Moreover, DQL algorithms are more stable due to less frequent updates of the weights in the DNN.

The authors of [4] use Deep Q-learning technique to reduce the system cost (that includes the cost of communication, computation, and caching storage) in the presence of finite cache storage. The authors propose a deep Q-learning technique with two different models at two different time scales, a large time scale corresponding to several time slots whereas a small time scale for single time slot. In the proposed algorithm, states depend on SNR, contact frequency, and contact time between the vehicles and the edge servers. Actions are the caching decision, number of coded packets (using Fountain code) to be cached, and making the task offloading decision. The reward is evaluated based on the cost of communication, computation, and caching. In [11], the authors propose another low cost and delay tolerant technique based on deep Q-learning which makes decision on whether to cache data at the edge servers or at vehicles (used as cache storage). The states in the algorithm include the amount of data cached in the queue whereas the action is the data scheduling decision. They evaluate the reward function based on data scheduling loss (which includes the size of data that does not satisfy the delay constraints and cost of energy consumption to cache the data) which is minimized by the proposed deep Q-learning algorithm.

Deep Deterministic Policy Gradient (DDPG) is another model-free Deep RL technique useful for continuous action space. DDPG is an actor-critic algorithm, where the actor network learns the optimal policy by taking actions based on the given state of the agent. The critic network evaluates the performance of the action taken and provides feedback to the actor network. In [2], the authors propose a DDPG based joint caching and computing algorithm that works well for an action space with several continuous random variables such as the amount of caching resources, computation resources, and bandwidth allocated to vehicles. Here the state depends on the state of the vehicle (mobility parameters), available computation and caching resources, and bandwidth of the server. For reward, the authors proposed utility functions for computing, caching, and energy consumption. Specifically, the caching utility depends on the time taken by the server to cache the content, price paid by vehicle to request the content from the cache, and the popularity of the content.

C. Content retrieval

Content retrieval is the last major task for the successful operation of the caching application in VEC. Vehicles can benefit from caching contents in VEC only if the content is reliably and correctly received at the receiver. Challenges such as the design of MAC protocol, efficient resource allocation, and robust interference management should be properly addressed to optimize the content retrieval at the vehicles.

As discussed in Section II, IEEE 802.11p and C-V2X technologies will be used for content retrieval in VEC. For IEEE802.11p, AI can help in time slot allocations in Time Division Multiple Access (TDMA) based MAC protocols, selection of Contention Window (CW) for Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) based MAC protocols, multi-hop transmissions, and efficient cluster selection (such as the size of the cluster and the cluster head). In C-V2X, AI techniques can be useful for optimal resource block allocation, scheduling decisions, and relay selection.

In [12], the authors propose a Q-learning based MAC protocol for adaptive CW selection in IEEE 802.11p network. The goal of the technique is to reduce content access latency and improve transmission fairness among vehicles. Here, the CW value is taken as the state, packet transmission is the action, and the reward is chosen to select similar CW values within a neighborhood for fairness as Table III shows.

The work in [13] uses Q-learning to find the optimal ratio of time for UpLink (UL) transmissions and DownLink (DL) transmissions in a Time Division Duplex (TDD) based 5G vehicular network. This proposal improves the content throughput and reduces the packet loss. To maximize resource utilization in network slicing based vehicular networks, the authors of [14] propose an offline Q-learning technique. The uplink and downlink slicing ratios in each slice is defined as number of resources allocated to the uplink and the downlink respectively divided by the total number of resources. The resource utilization is computed based on the ratio of used resources to the total allocated resources. Here the actions are the slicing ratios and reward is the resource utilization. While online Q-learning uses real-time data for making caching decisions, offline Q-learning is applied on large amounts of stored data. A major disadvantage of offline Q-learning algorithm in context of content caching is that it needs to be re-evaluated every time the data changes, and it is therefore not suitable for dynamic vehicular networks.

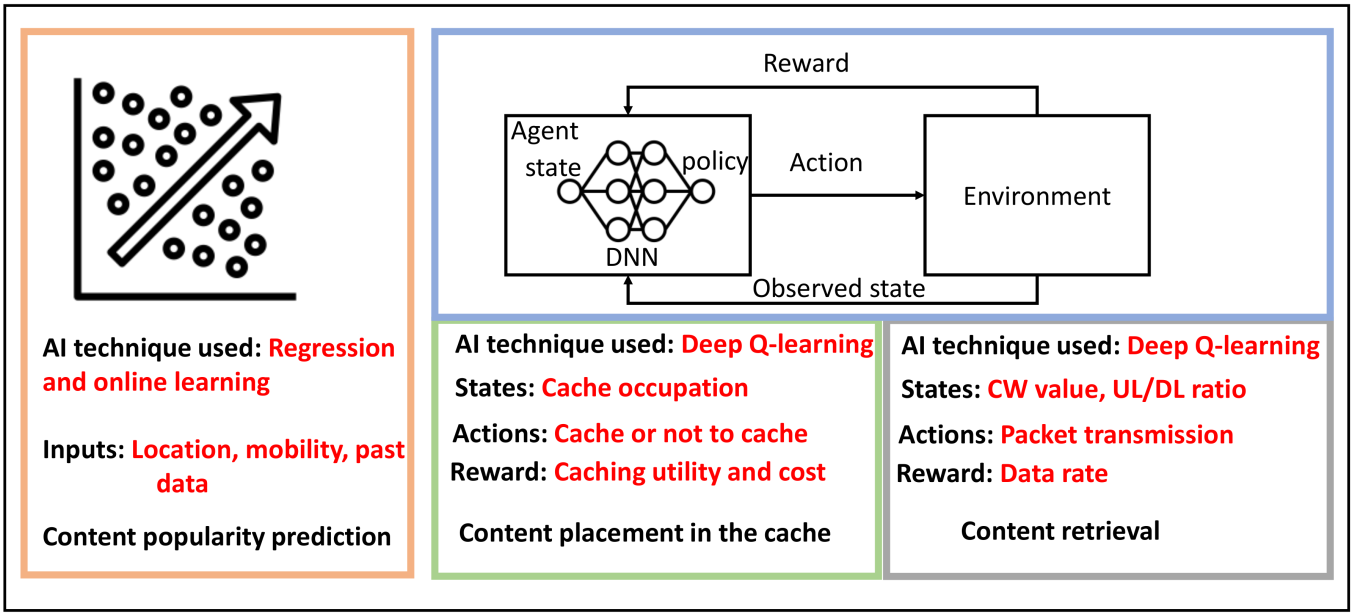

Fig. 2: AI techniques used in caching tasks in VEC.

D. Discussion

Fig. 2 presents a summary of AI techniques used in VEC based caching. For content popularity prediction, regression and online learning are the suitable AI techniques where inputs to the regression algorithm include location, mobility of vehicles, and past popularity data. For content placement and content retrieval, the optimal AI technique is deep Q-learning. Here states include CW value, UL/DL ratio and slicing ratio depending on the defined problem and scenario. Actions are making decisions about caching data in the RSU, caching data in the cloud or not to cache. For content retrieval, the action is to find optimal packet transmission parameters. Finally, reward is evaluated based on system utility and cost. The goal is to maximize the system utility (including caching utility, computing utility and communication utility) and reduce the system cost (including communication cost and energy consumption cost). In case of content retrieval, the reward is a function of data rate.

IV. FUTURE OPPORTUNITIES

In this section, we discuss some future opportunities and open challenges related to caching in VEC. Table IV presents a summary of these opportunities and challenges in.

A. Collaborative data sharing using Federated learning

Federated learning is a technique in which each distributed edge server runs machine learning models based on collected localized data and collaborate to train a robust centralized machine learning model [15]. It has advantages such as maintaining the data privacy of edge servers and it is also a global improved learning model that can make robust decisions. Federated learning can be useful for further improving the caching decisions based on collaborative data sharing of content popularities, vehicle mobility and the amount of free storage space in the cache. For a highly dynamic IoV network, such collaboration of machine learning models can help implement caching on the fly, where contents can be kept into the cache irrespective of vehicle mobility (by moving the content into the cache of a neighboring edge server if the vehicle moves away from the current edge server) and can be continuously downloaded. Intelligent handover schemes can be implemented using federated learning. The future challenges in this area include protocols for collaborative model sharing and making decisions based on different models trained on different datasets.

B. Improving the sensing-based semi-persistent scheduling protocol for content retrieval

In C-V2X, mode 4 is defined for autonomous resource allocation which is particularly useful for direct communication among the vehicles. The 3rd Generation Partnership Project (3GPP) has proposed a Sensing-Based Semi-Persistent Scheduling (SB-SPS) protocol for autonomous resource allocation [1]. From the perspective of content retrieval from edge servers, C-V2X mode 4 communications among vehicles can play an important role. Vehicles can act as relays to transmit the content from the edge servers to the vehicle requesting the content. A learning technique such as Q-learning can help improve the operation of the SB-SPS protocol. The sensing procedure in SB-SPS requires a selection window parameter which refers to the window of resource blocks from which to select a resource. Furthermore, it requires an estimation of available resources within the selection window and using that estimate, it selects a random available resource for transmitting the packet. Similarly, in the semi-persistent scheduling procedure, a reselection counter value is selected which corresponds to the number of packets that can be sent

TABLE IV: Future opportunities and open challenges

Future opportunity	Application	Open challenges
Collaborative data sharing using federated learning	Collaborative data sharing, development of robust machine learning model, intelligent handover	Protocols for model sharing, decision making is based on different models
Improving the sensing-based semi-persistent scheduling protocol for content retrieval	Vehicle to vehicle communication with autonomous resource allocation, content retrieval from RSUs using vehicles as relay nodes	Q-learning to select optimal selection window and reselection counter value
Secure caching	Protection against security threats such as jamming, man-in-the-middle and fake edge servers	Feature learning to analyze attacks, reputation-based content placement using deep Q-learning

by a vehicle consecutively. With Q-learning, the best action, in terms of these parameter values, can be selected to maximize the reward function (which depends on the packet success rate and packet delay).

C. Secure caching

Security is a key concern for caching in VEC. Malicious vehicles can send fake content requests to influence the content popularity prediction. For instance, they can send jamming signals to disrupt the communication link between edge servers and vehicles and broadcast itself as fake edge server. Feature learning techniques such as K-means clustering and support vector machine can be useful in detecting and analyzing the attacks attributes. For example, the frequency of fake content requests can be analyzed and the location of the malicious vehicle can be identified using the transmission power of the jamming signals. Similarly, a deep Q-learning technique can be used to make decisions on whether to place a certain content in the cache or not, and allocating resources for content retrieval based on the reputation of vehicles.

V. Conclusion

In this paper, we discuss how AI techniques can support efficient caching in VEC. We present a task-based architecture of caching in the context of IoV, which relies on three main tasks namely, content popularity prediction, content placement and content retrieval. We describe how various AI techniques such as regression and deep Q-learning can provide efficient solutions to perform these three tasks. Finally, we highlight a few future opportunities for efficient caching in VEC.

VI. Acknowledgments

We thank the anonymous reviewers for their valuable comments which helped us improve the quality and presentation of this paper.

References

[1] S. Zeadally, M. A. Javed, and E. B. Hamida, "Vehicular communications for its: Standardization and challenges," *IEEE Communications Standards Magazine*, vol. 4, no. 1, pp. 11–17, 2020.

[2] Y. Dai, D. Xu, S. Maharjan, G. Qiao, and Y. Zhang, "Artificial intelligence empowered edge computing and caching for internet of vehicles," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 12–18, 2019.

[3] L. Hou, L. Lei, K. Zheng, and X. Wang, "A q -learning-based proactive caching strategy for non-safety related services in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4512–4520, 2019.

[4] L. T. Tan and R. Q. Hu, "Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10 190–10 203, 2018.

[5] M. Rahim, M. A. Javed, A. N. Alvi, and M. Imran, "An efficient caching policy for content retrieval in autonomous connected vehicles," *Transportation Research Part A: Policy and Practice*, vol. 140, pp. 142 – 152, 2020.

[6] Z. Hu, Z. Zheng, T. Wang, L. Song, and X. Li, "Roadside unit caching: Auction-based storage allocation for multiple content providers," *IEEE Transactions on Wireless Communications*, vol. 16, no. 10, pp. 6321–6334, 2017.

[7] Y. Liu, T. Zhi, H. Xi, X. Duan, and H. Zhang, "A novel content popularity prediction algorithm based on auto regressive model in information-centric iot," *IEEE Access*, vol. 7, pp. 27 555–27 564, 2019.

[8] P. Yang, N. Zhang, S. Zhang, L. Yu, J. Zhang, and X. Shen, "Content popularity prediction towards location-aware mobile edge caching," *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 915–929, 2019.

[9] Y. Jiang, M. Ma, M. Bennis, F. Zheng, and X. You, "User preference learning-based edge caching for fog radio access network," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1268–1283, 2019.

[10] N. Garg, M. Sellathurai, V. Bhatia, B. N. Bharath, and T. Ratnarajah, "Online content popularity prediction and learning in wireless edge caching," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1087–1100, 2020.

[11] Q. Luo, C. Li, T. H. Luan, and W. Shi, "Collaborative data scheduling for vehicular edge computing via deep reinforcement learning," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[12] A. Pressas, Z. Sheng, F. Ali, and D. Tian, "A q-learning approach with collective contention estimation for bandwidth-efficient and fair access control in ieee 802.11p vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 9136–9150, 2019.

[13] Y. Zhou, F. Tang, Y. Kawamoto, and N. Kato, "Reinforcement learning-based radio resource control in 5g vehicular network," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 611–614, 2020.

[14] H. D. R. Albonda and J. Prez-Romero, "An efficient ran slicing strategy for a heterogeneous network with embb and v2x services," *IEEE Access*, vol. 7, pp. 44 771–44 782, 2019.

[15] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Network*, vol. 33, no. 5, pp. 156–165, 2019.