***Journal of Artificial Intelligence and Technology*, 2022, 2, 132-143**
https://doi.org/10.37965/jait.2022.0120

REVIEW ARTICLE

Review on Offloading of Vehicle Edge Computing

Mingwei Wang,[1] Hualing Yi,[3] Feng Jiang,[1] Ling Lin,[2] and Min Gao[3]
[1]Chongqing Business Vocational College, Chongqing, China
[2]Chongqing Chang'an Automobile Co., Ltd, Chongqing, China
[3]Chongqing University, Chongqing, China

(Received 07 June 2022; Revised 26 July 2022; Accepted 27 July 2022; Published online 22 August 2022)

Abstract: Vehicle Edge Computing (VEC) is a new technology that can extend computing and storage functions to the edge of the Internet of Things systems. For limited computing power and delay-sensitive mobile applications on the Internet of Vehicles (IoV), it is important to offload computing tasks to the end of the VEC network. Still, high mobility data security and privacy resource management and the randomness of IoV brought about new problems to the offloading of VEC. To this end, this study focuses on the offloading of computing tasks in VEC. We survey principal offloading schemes and methods in the VEC field and classify the current offloading of computing tasks into different categories. We also discuss the prospect of VEC. This survey could give a reference for researchers to find and understand the essential characteristics of VEC, which helps choose the optimal solutions for the offloading of computing tasks in VEC.

Key words: computing offloading; internet of vehicle; vehicle edge computing

I. INTRODUCTION

With the continuous development of information technologies such as science and technology, wireless network communication technology, and artificial intelligence (AI), vehicles have become more and more interconnected in the era of the interconnection of all things, while the applications for vehicles are also more intelligent and diversified at the same time. The rapid evolution of Internet of Vehicles (IoV) has led to the emergence of various delay-sensitive vehicle applications such as automatic driving systems and high-precision maps. Automotive applications usually need to handle large amounts of data. Thus, these applications have brought severe challenges to vehicles with limited computing resources [1]. To solve such problems, various computer load strategy and communication infrastructure have been proposed. They use the Intelligent Video Surveillance communication capabilities to transfer computing aggregation tasks to the cloud center of the computing applications. However, the long communication transmission distance between the vehicle and the cloud center leads to a long response time and unstable communication links [2]. The emergence of cloud computing technology satisfies the computing power requirements of resource-intensive applications to some certain extent and also brings the problem of propagation delay to applications. Mobile edge computing is a new network paradigm whose computing could be completed at the network edge of the terminals, which can effectively solve the delay problem of delay-sensitive applications [3]. Vehicle Edge Computing (VEC) is a mobile edge computing IoV application. It provides solutions to enhance computing and improve delay-sensitive onboard applications. As shown in Fig. 1, the vehicle improves transportation efficiency and reduces traffic accident rate and energy loss through information interaction with roadside units (RSUs) and cloud centers [4].

Computing offload can perform computing-intensive and delay-sensitive tasks performed in an edge server, and it significantly reduces task processing delay and energy consumption. At the same time, computational offloading can not only utilize less core network bandwidth to ensure data transmission rate but also alleviate network congestion, thereby reducing the risk of network congestion. Computing offloading is the key of VEC. However, vehicles in the IoV environment are with high mobility and dynamic network topology, so the computing offloading in VEC faces great challenges. In addition, each vehicular terminal in VEC, which is different with other common mobile terminals, could be a task vehicle for service request and a service vehicle for task execution at the same time. Therefore, computing offloading in VEC is an important research direction [5,6]. With extensive investigation and research on VEC, we analyze and give a summary for computing offloading on VEC.

The rest of the paper is organized as follows. Section II shows the outline of VEC. In Section III, application scenarios of VEC are introduced. In Section IV, an existing study on computational loading for VEC is summarized. In Section V, we discuss the challenges and future tasks of computing offloading. The full text is summarized in Section VI.

II. THE OVERVIEW OF VEC

In the VEC network, every vehicle itself has some capabilities to finish communication, computing, and resource storage, while the RSUs located near the vehicle act as an edge server for timely data collection, processing, and storage. Due to the limitation of own capacity, vehicles can offload computationally intensive and delay-sensitive tasks to edge servers, shortening response times and reducing the burden of backhaul networks.

Corresponding author: Hualing Yi (e-mail: yihualing@cqu.edu.cn)

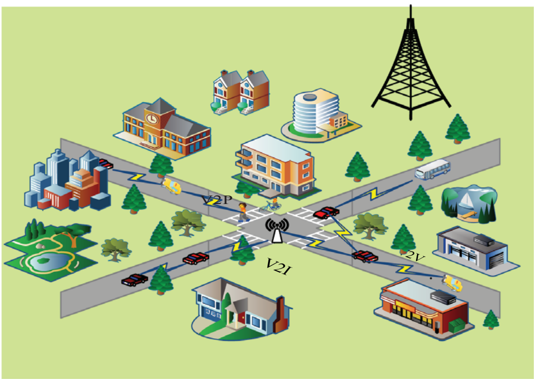

Fig. 1. VEC network.

A. THE ARCHITECTURE OF THE VEC NETWORK

The network structure of VEC is shown in Fig. 2. Generally, the architecture includes three layers, that is, vehicle terminal as user layer, RSUs as multi-access edge computing (MEC) layer, and cloud server as cloud layer.

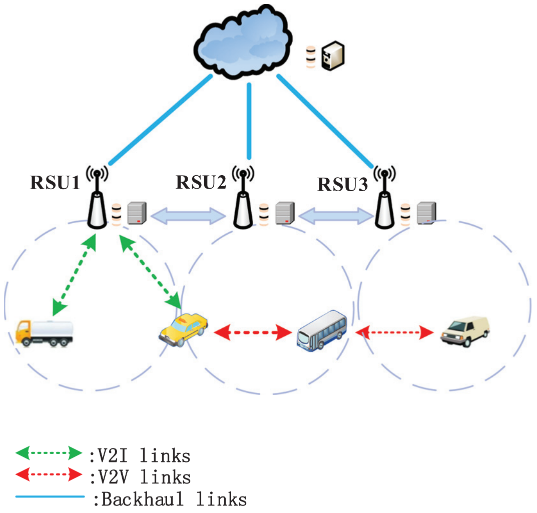

Fig. 2. The architecture of VEC.

B. COMPUTING OFFLOADING

The computing offloading in VEC is that the vehicles upload the computing tasks to edge servers for processing to reduce the consumption of time and energy for vehicles. As shown in Fig. 3, computing offloading includes offloading policy, resource allocation, and offloading system implementation. The offloading process is as follows.

(1) Request service. The task vehicle sends a service request to the infrastructure within its communication range (e. g. base station, RSU).

(2) Upload tasks. The task vehicle sends the computing tasks to edge servers through V2X (Vehicle to Everything) communication.

(3) Execute tasks. According to resource allocation policy, the edge server determines to offload tasks somewhere to finish computing tasks.

(4) Return results. The edge server sends computing results back to the task vehicles.

The computing offloading could reduce network overloading and communication delay. Content requesters can obtain content directly from edge service nodes without accessing core networks, thus reducing end-to-end delay and improving network bandwidth utilization efficiency [7].

C. OFFLOADING POLICY. The offloading policy is to determine if the vehicle will offload and how much to offload. There are three types of results that are usually offloading policy VEC, local execution, complete offloading, and partial offloading policies. The results for the task vehicle are determined by the energy delay of the vehicle and the time delay of the computing task. The goal of offloading policy could be summarized as several aspects, reducing delay, reducing energy consumption, balancing delay, and energy consumption.

(1) Local execution. The computing task is completed by the vehicle itself.

(2) Complete offloading. The computing task is offloaded and processed by MEC.

(3) Partial offloading. Part of the computing task is processed locally, and the others are offloaded to the MEC server for processing.

The delay, energy cost, and delivery of task vehicles in different offloading policies are listed, as shown in Fig. 4.

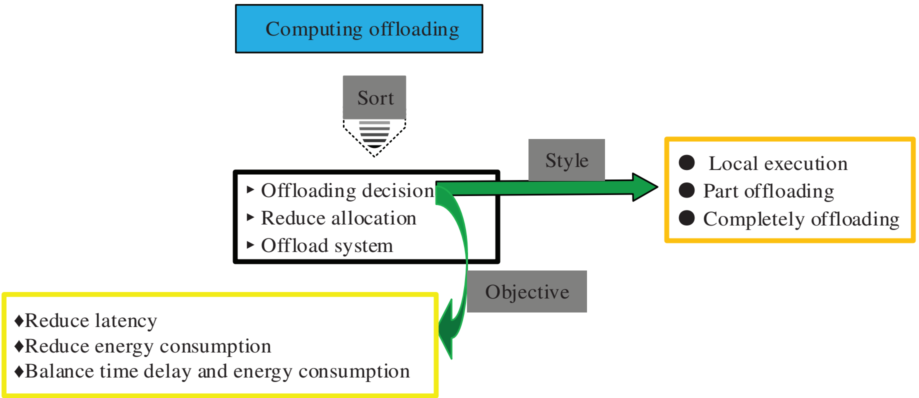

Fig. 3. The computing offloading.

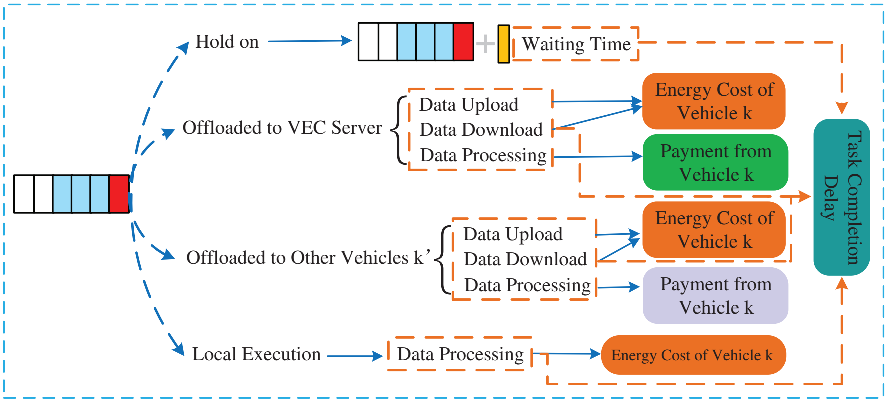

Fig. 4. Delay, energy cost, and delivery of different offloading policy.

The offload policy needs to consider the factor of computing delay, as the delay could affect the quality of user experience. In addition, the problem of energy consumption needs to be considered. If the device's energy consumption is too high, the battery of the mobile device terminals will run out.

D. OFFLOADING MODES

A vehicle in VEC could be a task vehicle, service vehicle, or relay node at one time. Therefore, the computing offloading modes of VEC are various. The main computing offloading modes are as follows.

Vehicle to infrastructure offloading (V2I). The onboard equipment communicates with the RSU to offload the computing task directly to the infrastructure.

Vehicle-to vehicle-offloading (V2V). Carry out the communication between vehicles through the onboard terminals and offload the task directly to the adjacent service vehicles through the remaining computing resource.

Vehicle to vehicle to infrastructure offload (V2V-I). Through the communication between multiple vehicles, the target vehicle offloads the computing offloading task to the infrastructure through other vehicles.

Vehicle to vehicle-to-vehicle offloading (V2V-V). Through the communication between multiple vehicles, the target vehicle offloads the computing offloading task to the service vehicle through other vehicles.

III. THE APPLICATION SCENARIOS OF VEC

IoV is usually applied in security, efficiency, and entertainment. For limited computing and energy resources, mobile vehicles tend to offload the computing-intensive tasks to edge servers. Traditional methods still work well for entertainment applications that are insensitive to delay and reliability. For example, it could greatly improve computing power for vehicle entertainment applications by offending computing-centric tasks to remote cloud centers and returning results from cloud centers. When computing tasks are very sensitive to delay and reliability, such as emergency obstacle avoidance, path planning, road state recognition, vehicles are usually far away from the cloud center, thus resulting in high network delay between the core network and backbone network. The emergence of VEC enables various applications to be realized. Several typical VEC application scenarios are shown in Fig. 5.

A. ROAD SAFETY

VEC is applied to road safety (such as emergency obstacle avoidance, path planning, road state recognition) for edge servers. Its role is to perform real-time analysis and processing of data from vehicles and sensors deployed along the road, and the edge servers could give danger warnings when they find insecurity factors. Thus, the surrounding vehicles could take proper braking, lane changing, and turning to avoid possible hazards.

B. ENTERTAINMENT

The benefit of VEC's flexible computing and storage resource is that intelligent vehicles make the driver's driving operation much simpler. Drivers can be distracted attention from driving on entertainment activities like games, music, and videos [8]. Through the content cache by the cooperation between vehicles and edge servers, vehicles' applications obtain the content

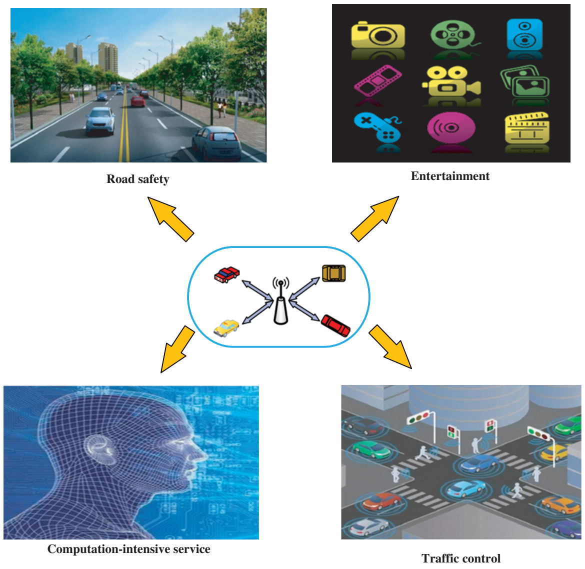

Fig. 5. VEC application scenarios.

resource without the help of a remote cloud, reducing the delay and improving the user experience.

C. TRAFFIC CONTROL

In VEC, the edge server covers a certain communication area. Each vehicle sends its current state (position speed) and information collected around the vehicle (weather conditions and road conditions) to the relevant server. By receiving information uploaded by vehicles from the VEC network, servers in VEC could get the local traffic conditions in time and then control the traffic flow to help avoid traffic congestion [9].

D. COMPUTING-INTENSIVE SERVICES

Computing offloading of VEC can be used to provide services for computing-intensive applications, such as augmented reality and facial recognition. Vehicles usually cannot meet computing requirements for their applications due to their limited resource. Therefore, they reduce latency and improve user experience by offloading computing tasks to edge servers.

IV. COMPUTING OFFLOADING IN VEC

The existing computing offloading research usually focuses on optimizing delay and energy. The computing offloading in VEC puts various solutions and methods for different objectives. The computing offloading in VEC could be divided into two categories, as shown in Fig. 6.

A. USER-SIDE OPTIMIZATION

1) DELAY MINIMIZATION. The development of vehicle applications has improved the user experience a lot. However, these applications are usually real-time, complex, and intensive [10]. Taking too much time to process application tasks might reduce the effectiveness of data, thus even leading to traffic accidents. VEC delay includes transmission delay, computing delay, and communication delay. Computing delay has an offloading delay, queuing delay, and processing delay. Therefore, minimizing the latency of VEC applications is of great significance. Zhao et al. proposed an edge cache and computing management scheme to optimize service caching, request scheduling, and resource allocation strategies. The results show that this method achieves suboptimal delay performance [11]. Dai et al. proposed a hierarchical edge cloud computing architecture to minimize energy consumption and task execution delay of mobile terminals by utilizing computational offloading algorithms and resource allocation mechanisms [12].

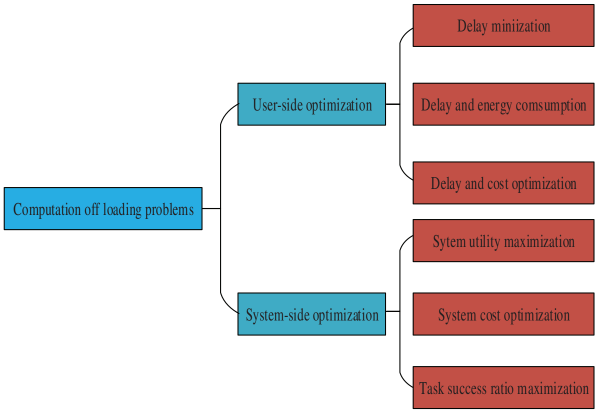
Computation off loading problems
User-side optimization
System-side optimization
Delay miniization
Delay and energy comsumption
Delay and cost optimization
Sytem utility maximization
System cost optimization
Task success ratio maximization

Fig. 6. Categories of computing offloading in VEC.

2) DELAY AND ENERGY CONSUMPTION OPTIMIZATION. Computing-intensive applications with strict time constraints bring challenges to VEC, usually resulting in higher energy consumption. When the energy of the vehicle itself is low, it is difficult to meet the needs of onboard applications. Therefore, energy consumption needs to be considered with time constraints, which means that the VEC offloading needs to balance and minimize the delay and energy consumption to obtain larger channel gain and lower local calculation energy consumption. Sun et al. proposed a task offloading algorithm based on adaptive learning. The proposed algorithm understands the delay performance of adjacent vehicles in calculating offloading and minimizes the mean load delay and energy consumption for each task vehicle [13]. Zhan et al. studied the computing offloading scheduling problem in VEC scenarios, modeled it through a well-designed Markov decision process, and designed the latest near-end policy optimization algorithm based on DRL [14].

3) DELAY AND COST OPTIMIZATION. Another goal of VEC is to reduce offloading, which includes communication costs and computing costs. Taking cellular network communication as an example, vehicles may need to pay for transmitting and receiving data to a cooperative distributed computing architecture using vehicles as a computing resource for VEC is constructed. Qiao et al. proposed a cooperative task offloading mechanism and transmission mechanism to reduce the delay and energy consumption of the system [15]. Zhao et al. proposed a cooperative computing offloading and resource allocation optimization scheme and designed distributed computing offloading and resource allocation algorithm. This scheme maximizes system utility (task processing delay and computational resource cost) by optimizing offloading strategy and resource allocation. The offloading strategy is optimized by game theory [16]. Salman Raza et al. proposed a motion-aware partial task offloading algorithm to minimize the total offloading cost. This method considers the cost of required communication and computing resources, divides the task into three parts, and further determines the task allocation ratio of the three parts according to the available resources of vehicle. The simulation results show that this method could not only reduce communication cost of nearby vehicles but also relieve the burden on VEC servers, especially when deployed in dense urban environments [17].

B. SYSTEM SERVER OPTIMIZATION

1) SYSTEM UTILITY MAXIMIZATION. In the computing offloading for VEC, the system resource needs to be balanced. There are no overloaded devices and no idle devices, ideally. The purpose is to use the system utility of VEC more efficiently. According to specific scenarios, Liu et al. proposed an offloading strategy and resource allocation scheme for vehicle edge servers and fixed edge servers. Considering the randomness and uncertainty of vehicle communication, VEC network's total utility maximization problem is reconstructed into a semi-Markov process, and a reinforcement learning method and DRL method based on Q-Learning are proposed to find the best strategy for computing offloading and resource allocation [18]. Dai et al. proposed an AI-enhanced VEC and cache scheme. The proposed architecture cooperates edge computing, intelligently caches resources, realizes cross-layer offloading, multi-point caching, and delivery, and improves system utilities using DRL and DDPG algorithms [19].

2) SYSTEM COST MINIMIZATION. Tan et al. developed a joint optimal resource allocation framework for communication, caching, and computing in VEC networks. Aiming at the optimization problem of resource allocation, they propose a multi-time scale frame-based algorithm based on deep reinforcement learning with the aim of optimizing resource allocation with limited storage capacity, computational resources, and severe delay constraints of vehicles and RSUs [20]. Zhang et al. formulated a joint cache and computing allocation strategy for minimizing system caching. A mobile recognition active cache technology is proposed in which core ideas are to obtain and store video content in the cache of the base station, and a KNN algorithm is used to find the optimal joint view set to maximize the total reward and minimize the cost of the system [21].

3) MISSION SUCCESS RATE MAXIMIZATION. Qiao et al. focused on cooperative edge caching and auxiliary caching and proposed a cooperative edge caching scheme to optimize content placement and delivery. The joint optimization problem is a two-time scale Markov decision process [22]. Liang et al. employ model resource sharing and deep Q-network (DQN)-based methods as multi-agent reinforcement learning vehicles in network spectrum sharing. The V2V spectrum and power allocation scheme are designed to improve payload delivery reliability of the V2V link and realize periodic safety critical message sharing [23].

V. CHALLENGES FOR VEC

The environment of task offloading for VEC is dynamic and random. Its network topology, wireless channel state, and computing workload are changing rapidly. These uncertainties bring additional challenges to VEC. We list some main challenges and their relationships in VEC, as shown in Fig. 7.

A. High mobility

In the scene of IoV, the high mobility of vehicles and the frequent switching of information lead to the dynamic change of its network topology, and the communication link is easy to be disconnected, which affects the communication quality. Therefore, developing a mobile model is imperative, which is used to judge its mobility mode accurately according to a different environment in practical applications, perceive and learn the information of popular vehicles better, and improve communication reliability and resource utilization. Junior et al. constructed a mobile offloading system (MOSys) that supports seamless offloading operations when users move between wireless networks, which can effectively reduce offload response time. The mobility management application of MOSys has proved to have good energy efficiency [24]. Boukerche proposes a mobility-oriented prediction search protocol for the offloading scheme in VEC. The protocol uses position prediction to estimate the location of the user and achieves a more efficient search for delay and retrieval rate [25].

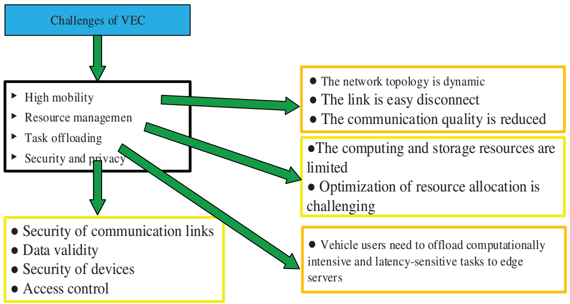
Challenges of VEC
High mobility
Resource managemen
Task offloading
Security and privacy
● The network topology is dynamic
● The link is easy disconnect
● The communication quality is reduced
●The computing and storage resources are limited
● Optimization of resource allocation is challenging
● Vehicle users need to offload computationally intensive and latency-sensitive tasks to edge servers
● Security of communication links
● Data validity
● Security of devices
● Access control

Fig. 7. Challenges in VEC.

B. RESOURCE MANAGEMENT

The rapid channel change in high mobility vehicle environment makes it difficult to collect accurate instantaneous channel state information at base stations for centralized resource management. Edge servers have limited computing and storage resources, and this limitation brings challenges to resource allocation. The cognitive engine is introduced into the IoV to realize more intelligent vehicle deployment and resource allocation. The recognition engine is introduced to IoV to achieve more intelligent vehicle deployment and resource allocation. Cognitive engines could recognize the content requirements of users and match content providers and requesters to improve cache hit rates. The cognitive engine recommends RSU or vehicle-related content requirements to perceive content requirements based on machine learning or deep learning (DL) methods and provide content. This content acquisition method connects different vehicles on the one hand but brings the problem of unreliability on the other hand.

To solve the problem of resource management, Lin et al. found that social edge computing (vsec) can improve the quality of experience and service. To further enhance the performance of the vsec, network utilities are modeled and maximized by optimizing the management of available network resources [26]. Zhang et al. designed an effective wholesale scheme to maximize the total expected profit of mobile edge servers and proposed a fast convergence real-time repurchase scheme that minimizes the repurchasing cost considering the reserved computing resources [27]. Li et al. proposed an intelligent resource scheduling strategy to meet real-time requirements in smart manufacturing with edge computing support. The proposed strategy proved excellent real-time satisfaction and energy consumption performance of computational services in intelligent manufacturing by edge computing [28]. Wang et al. proposed a mechanism for configuring and automatically scaling edge node resources and developed an edge node resource management framework. The results verified the potential of this method in improving transmission efficiency and low delay [29]. Wang et al. proposed a task scheduling strategy suitable for an edge computing scenario. Load balancing of the network is measured according to the load distribution. By combining with the reinforcement learning method, the computational task continuously matches the appropriate edge server and satisfies the resource differentiation needs of the node task. The proposed strategy effectively reduces the load between the edge server and core traffic network and reduces the task processing delay [30]. You et al. proposed an energy-saving resource management strategy for an asynchronous mobile edge computing offloading system. An energy-saving offloading controller was designed for the asynchronous mobile edge computing offloading system, eliminating the overhead required for network synchronization and reducing the offloading and computing delay [31]. Peng et al. proposed a multi-agent deep deterministic policy gradient (MADDPG) . Resource allocation on MEC servers is formulated as a distributed optimization problem to maximize the number of offload tasks while satisfying the demand for heterogeneous quality of service. The MADDPG model is trained offline, while the car association and resource allocation decisions are made faster by online running stages. Simulation results show that the proposed resource management scheme based on MADDPG improves resource utilization and achieves a high degree of delay in heterogeneous service quality [32]. The summary of resource management based on MEC is shown in Appendix A.

C. TASKS OFFLOADING

Most research on task offloading in VEC networks rarely considered the load balance of computing resources on edge servers. To support centralized network management and various communication technologies of fiber-optic radio (FiWi) technology, Zhang et al. introduce FiWi to enhance VEC network. They proposed a load balancing task offloading scheme based on a software-defined network in FiWi-enhanced VEC networks to minimize the processing delay of vehicle computation tasks. Theoretical verification shows that the scheme reduces the processing delay by using the computing resources of the edge server [33]. Zhan et al. proposed a computing offloading game formula to solve the dynamic computing offloading decision-making problems among edge computing users in the dynamic environment [34]. Tang et al. propose a cost-effective offloading method with a deadline in the MEC environment to provide an efficient offloading service in the edge computing environment [35]. Zhang et al. proposed a technique based on meta-reinforcement learning to solve problems of computing offloading in MEC [36]. A summary of computing offloading based on MEC is shown in Appendix B.

Many researchers have carried out relevant research on task offloading. Li et al. proposed a heuristic algorithm with low complexity to obtain the offloading strategy and ensure load balance among multiple MECs simultaneously. Then, the Lagrange dual decomposition method was used to solve the computational resource allocation sub-problem [37]. Ding et al. proposed a cache-based NOMA downlink heterogeneous network architecture, formulated a messaging user coordination strategy, and allocated users' power reasonably. The MEC system, which built a new offloading system model based on D2D communication, used utility function optimization and game theory to solve the computing diversion problem [38]. Ke et al. propose local computing offloading and resource allocation scheme for distributed reinforcement learning. The arrival rate and the signal-to-noise ratio of the workload are calculated according to the time-varying channel state. Distributed reinforcement learning can achieve optimal resource allocation and privacy protection without exploring and learning prior knowledge [39]. Ding et al. proposed a new MEC offloading system based on D2D communication offloading system model. The parameter is replaced with the function optimization using the utility function. Game theory was used to solve the problem of calculation conversion. This scheme has good convergence, improves data transmission rate, and reduces the energy consumption of the task [40]. Chen et al. propose a strategic computation load algorithm based on a DQN without understanding prior knowledge of network dynamics, and we study an optimal strategy. By the additive structure of the utility function, the Q-function decomposition technique is combined with the double DQN, and a new learning algorithm to solve the conversion of the random calculation is obtained. The Markov decision process was used to solve the problem of optimal computing diversion strategy, and the offloading decision was made according to tasks queue state, energy queue state, and channel quality between mobile users and base stations [41]. Chen et al. proposed a task layout algorithm to solve the problem of task placement and resource allocation. The proposed scheme improved the offloading efficiency, reduced the task duration, and reduced energy consumption [42]. Su et al. proposed the layered computing offloading strategy and developed the layered computing offloading framework in the network virtualization scenario by using the advantages of MEC. The first step was to propose an incentive mechanism, which was realized by considering individual rationality and incentive compatibility constraints. After obtaining the CPU contribution of edge computing nodes, the second step is to consider computing resource allocation between MEC and computing service subscribers. They have developed an iterative matching algorithm that can be used to model the problem with an external Bayesian matching algorithm and converge to always stable results. This strategy improved the computing power of nodes and allocated resource reasonably [43].

D. PRIVACY PROTECTION AND SECURITY

In VEC, the high mobility of vehicles makes it difficult to solve the reliability problem between nodes. How to ensure the high security of servers and solve the complex computing offloading problem in VEC is still an urgent problem to be studied. Due to the particularity of IoV service, vehicles need to broadcast their location information and exchange data with each other. Under this complex network condition, it is easier for hackers, malicious users, and malicious wireless nodes to mix in. Thus, the security and privacy of the edge network environment are greatly threatened [44]. The vehicle broadcast information is collected by the attacker, and it steals the privacy information by analyzing the collected data. Furthermore, attackers could even track the owner through the driving trajectory, which brings serious security problems. Privacy types could mainly classify the research work in this area. The classification of privacy protection is shown in Fig. 8.

Recently, some researchers have focused on integrating blockchain into IoV to provide the latter with the security and privacy required for operation. It is recommended that using blockchain-based access control disable the cloud illegally to improve the load loading safety of computing. Boukerche et al. proposed a reliable access control mechanism to effectively detect and prevent the illegal loading of devices. The purpose is to verify vehicle identity, offload tasks, and manage offload data to ensure system security and privacy [45]. Liu et al. proposed a group authentication scheme of vehicle blockchain authorization with decentralized identity based on secret sharing and dynamic agent mechanism, which realized the

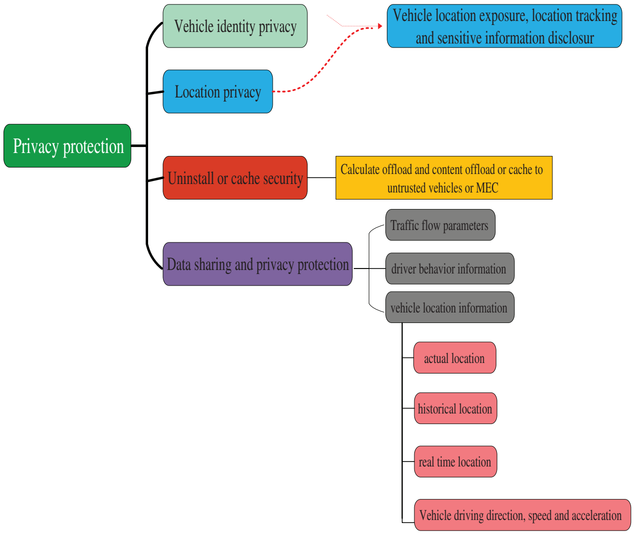
Privacy protection
Vehicle identity privacy
Location privacy
Vehicle location exposure, location tracking and sensitive information disclosur
Uninstall or cache security
Calculate offload and content offload or cache to untrusted vehicles or MEC
Data sharing and privacy protection
Traffic flow parameters
driver behavior information
vehicle location information
actual location
historical location
real time location
Vehicle driving direction, speed and acceleration

Fig. 8. Types of privacy protection.

cooperative privacy protection of vehicles and reduced the communication overhead and computing cost [46]. Qian et al. propose a privacy-aware service placement method to solve the problem of privacy-aware service placement in edge cloud systems. The service allocation on the mobile edge cloud is modeled as a 0–1 problem. Next, we propose a hybrid service placement algorithm combining the concentrated greedy algorithm and the distributed coupling learning [47]. Shen et al. established a privacy game model to optimize the targets of defenders/attackers and solved the problems of mutual privacy protection and task allocation [48]. Ma et al. proposed a new method to protect edge computing systems from illegal access based on biometric features [49].

VI. OUTLOOK

With the rapid development of IoV, although many researchers have conducted in-depth research on the VEC network, many technical problems have been solved in calculating the VEC's offloading. We list some research hotspots and trends in computing offloading in the future.

A. VEC BASED ON BLOCKCHAIN

Integrating blockchain and computing offloading of VEC could realize the data transmission and sharing mechanism. Blockchain can protect the cached content and data from malicious node attacks or denial of service attacks in the dynamic network environment under VEC. MEC provides the blockchain with computing resources and consensus intelligent algorithms to ensure efficient data processing and transmission by edge nodes. By adopting blockchain on IoV, security threats such as interruption, single node failure, and availability attacks could be solved. Blockchain builds a decentralized platform, establishes a trust mechanism, protects vehicle privacy, and reduces communication delay. Yang et al. use the integration of blockchain and edge computing and their distributed management and distribution services to meet future network and system safety, scalability, and performance requirements [50]. Ma et al. propose a blockchain framework for developing a unified scheme to realize reliable cooperation between edge servers in the MEC environment, secure trust among servers, and realize real-time cooperation and task sharing [51]. Liu et al. propose a blockchain model which could support cross-domain authentication. This model overcomes the shortcomings of identity-based encryption (IBE), which is not suitable for large-scale cross-domain architecture, allowing communication across different IBE and public key infrastructure domains while ensuring the security of resource or service sharing in a distributed multi-domain network environment. Taking advantage of the distributed and immutable characteristics of blockchain, this model is highly scalable but without changing the internal trust structure of each authentication domain [52]. Guo et al. proposed to design an optimized practical Byzantine fault-tolerant consensus algorithm on basis of building a blockchain network to shape an alliance blockchain that stores authentication data and logs to ensure trusted authentication and achieve the traceability of terminal activities [53]. Rehman et al. proposed reputation-aware fine-grained joint learning based on blockchain to ensure credible cooperative training in MEC systems [54]. Therefore, the data processing interaction between blockchain and computing offloading is the focus of VEC in the future.

The architecture of VECN based on blockchain is shown in Fig. 9. The architecture is divided into three planes, user plane, data plane, and storage plane. The storage plane is composed of block headers, and its main function is to package transaction records. The data plane is composed of RSUs deployed on both sides of the

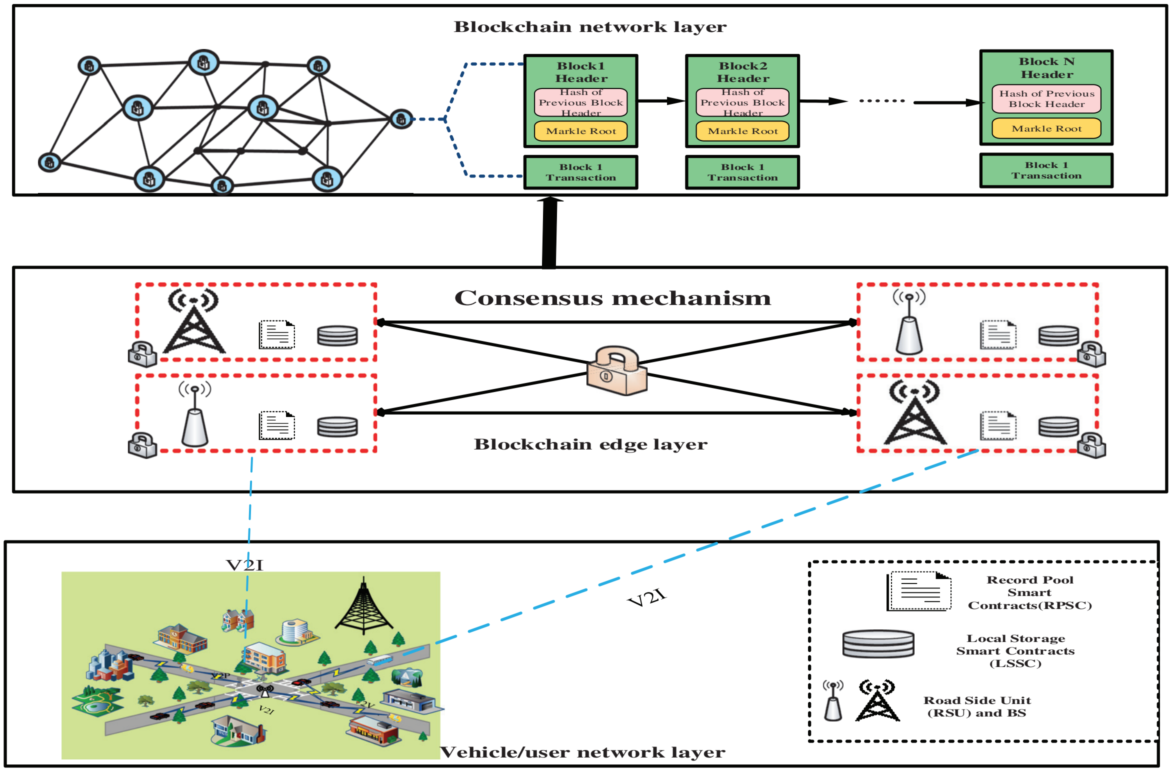
Blockchain network layer
Block1 Header
Hash of Previous Block Header
Markle Root
Block 1 Transaction
Block2 Header
Hash of Previous Block Header
Markle Root
Block 1 Transaction
Block N Header
Hash of Previous Block Header
Markle Root
Block 1 Transaction
Consensus mechanism
Blockchain edge layer
V2I
V2I
V2P
V2V
V2I
Vehicle/user network layer
Record Pool Smart Contracts(RPSC)
Local Storage Smart Contracts (LSSC)
Road Side Unit (RSU) and BS

Fig. 9. VECN architecture based on blockchain.

road and their affiliated MEC servers. The user plane refers to the vehicle users driving on the road.

B. VEC BASED ON EDGE INTELLIGENCE

Edge intelligence (EI) is AI in edge computing and developed by edge devices. EI refers to terminal intelligence that meets the essential needs of industry digitization from the perspective of agile connectivity, real-time business, data optimization, application intelligence, security, and privacy protection. Deng et al. divided EI into smart edge computing and AI for edge computing. The former is supported by AI technology, which provides a more optimized solution to the key problems in edge computing. The latter studied the whole process of AI model construction at the edge, model training, and reasoning, which provides a broader vision for this new interdisciplinary field [55]. Li et al. proposed a collaborative reasoning framework of DL models with device edge synergy. For low-delay EI, in the DNNS partition, it adaptively divides the DNN calculation between the mobile devices and edge servers based on the available bandwidth to make use of the processing capacity of edge servers and reduce the data transmission delay [56]. Salman Raza et al. proposed an edge computing framework based on Intel Up-Squared boards and Neural Compute Sticks that provides vision, voice, and real-time control services. Based on this framework, a guide dog robot prototype was proposed, which realizes the guide dog's obstacle avoidance, traffic sign recognition and following, and voice interaction with humans [57]. Zhou et al. proposed a method to realize intelligent collaborative edge computing in the Internet of Things to realize the complementary integration of AI and edge computing [58]. Xu et al. proposed a data-driven EI network anomaly detection method. Firstly, the main motivation of EI and its application in network anomaly detection were analyzed, and then, a network anomaly detection intelligent system based on edge computing was proposed. Finally, a data-driven network anomaly detection scheme was proposed. In this scheme, four stages were designed, which combined statistics and data-driven methods to train a learning model that can detect and identify network anomalies robustly [59]. Therefore, EI could reduce the delay and cost of data processing, store the original data in edge devices or terminal devices, protect private data, and promote the popularization and application of DL. Diversified services of DL extend the business value of edge computing. EI and in-depth learning will become an important direction for the future development of VEC.

C. VEC BASED ON FEDERATED LEARNING

As a distributed machine learning architecture, federated learning aims to protect the data security and data privacy for terminal devices and realize energy-efficient machine learning among computing nodes. On the one hand, edge computing helps federated learning save network resources, reduce training time, and enhance data privacy protection. On the other hand, terminal devices and edge nodes are limited by their delivery resource communication and computing processing capabilities due to the characteristics of their software and hardware. Federated learning could provide

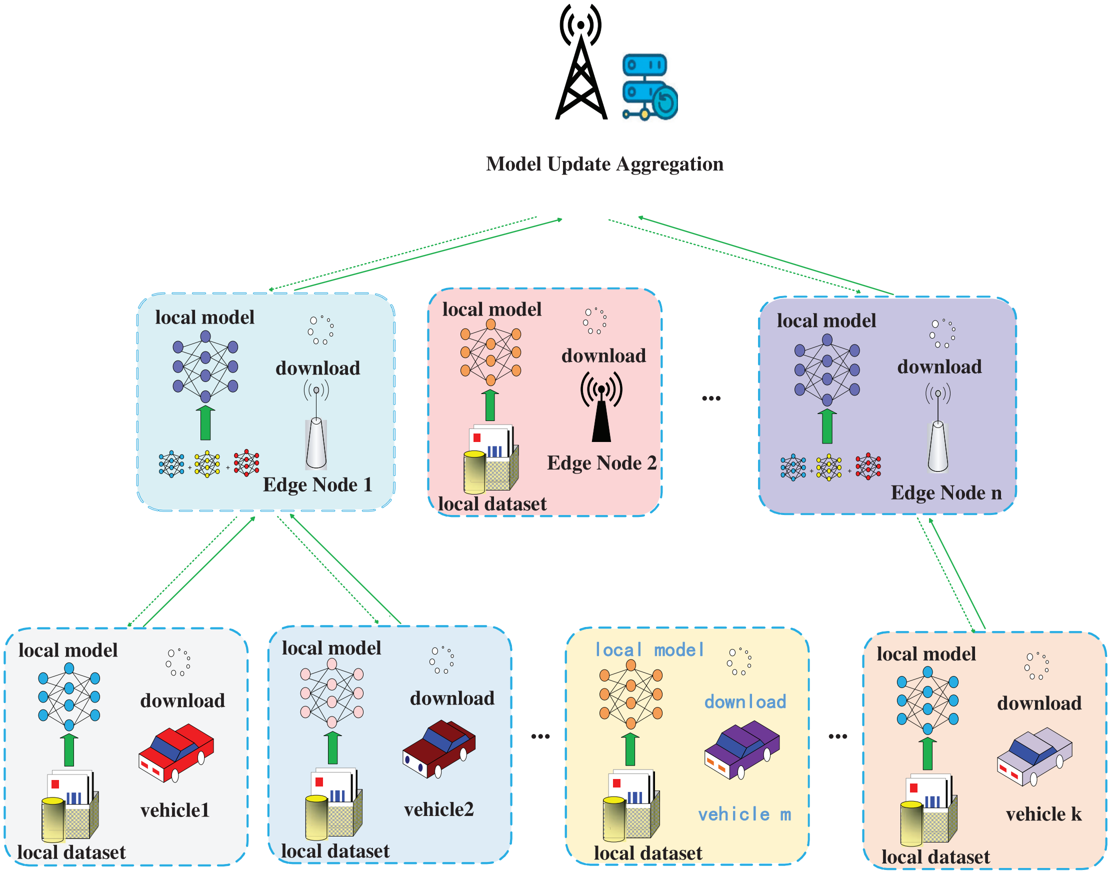

Fig. 10. Federated learning-oriented computing task offloading in VEC.

some effective solutions for edge computing systems. As shown in Fig. 10, the computing task offloading for federated learning in the VEC network includes information on the tested vehicle, edge nodes processing, and model aggregation. The vehicle information is downloaded to the local database, and the local database and edge nodes deliver and update the local model to form a local model aggregation system. In this process, multiple edge servers interact with multiple heterogeneous edge nodes, and each node stores its own user data. Zhou et al. proposed an efficient asynchronous federated learning mechanism for edge network computing, which reduced data transmission delay [60]. Ye et al. proposed an edge computing method based on federated learning, which reduced the cost of terminals participating in federated learning while protecting user privacy [61]. Zhang et al. proposed a mobile edge computing federated learning framework that integrates model partitioning technology and differential privacy, providing strong privacy guarantees while reducing the heavy computational cost of DNN training on edge devices [62].

There are often security problems when combing VEC offloading tasks and terminals. Blockchain could enable edge computing with encryption, non-tampering, and distributed characteristics, laying the foundation for edge computing to solve security problems. Lu et al. designed an adaptive block checking authorization federation learning framework for robust EI, a new digital dual wireless network model to reduce unreliable long-range communication between users and base stations by incorporating digital twins into wireless networks. Based on this model, data analysis could be performed directly on the digital twins at the base station, thus enhancing the security and privacy of the digital twin data [63]. Chu proposes a scalable blockchain and task offloading technique based on neural networks for mobile edge computing scenarios to solve the problems of high computing cost and large data propagation delay caused by traditional blockchain data offloading [64]. Shahbazi proposed a blockchain, edge computing, and machine learning to develop and promote intelligent manufacturing systems to solve the blockchain data offload, the blockchain of large computation, and data transmission delays [65].

Based on blockchain-enabled federated learning, VEC offloading undertakes terminal computing tasks, significantly reducing data interaction delay and network transmission overhead. Therefore, VEC offloading data are likely to be processed on the edge side.

D. VEC BASED ON DL

Ubiquitous sensors and intelligent devices from factories and communities generate large amounts of data. Growing computing power drives computing tasks and services from the cloud to edge networks. Due to the efficiency and latency of the power grid, the current cloud computing service architecture has hindered the vision of providing AI "for any organization and anyone, anytime, anywhere." It could be a possible solution that uses the resource of the edge network, which is close to the data source, to release DL services. As the representative technology of AI, DL can be integrated into the edge computing framework, constructed intelligent edges, and realized the dynamic and adaptive edge maintenance [66]. The integration of DL and computing offloading in VEC are mainly in the following aspects.

(1) The application of DL on edge. Construct the technical framework of VEC based on DL to provide intelligent services.

(2) DL reasoning in edges. Focus on DL's actual deployment and reasoning in VEC to meet various needs, such as accuracy and delay.

(3) The VEC offloading network architecture for DL. Focus on DL computing and how to adapt to the edge computing platform in terms of network architecture, hardware, and software.

(4) DL training on edge sides. Use distributed edge devices to train edge intelligent models under the constraints of resources and privacy.

(5) DL for optimizing edges. Maintain and manage aspects of the VEC offloading network with DL, such as edge cache, computing offloading, communication, network, and security protection.

Edge DL aims at seamless training between vehicles or edge nodes to solve the challenges of non-independent synchronization of training data, limited communication, unbalanced training contribution, privacy, and security in VEC. The integration of DL and VEC could be one of the development directions of DL training on edge sides in the future.

VII. CONCLUSIONS

Offloading computing-intensive and delay-sensitive tasks to edge servers is of great significance to vehicle terminals with increasing requirements for computing and storage resources, which is an important driving force for VEC development. In this paper, we conduct extensive research and literature analysis on the existing work of VEC and introduce VEC in detail, such as architecture, application scenarios, computing offloading, and recent studies. Then, we further expound the challenges of VEC and analyze research direction in future and development trends. In summary, the research on VEC is still in its infancy, and there are still many problems to be solved. We hope that the investigation from our work could provide a useful reference for further research on the computing offloading of VEC.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Mingwei Wang carried out the construction and writing of this paper. Hualing Yi edited and reviewed this paper. Feng Jiang and Ling Lin gave the bibliography analysis of this paper. Min Gao reviewed this paper. All authors had approved the final version.

Appendix A.Resource management based on MEC

References	Problems	Solution	Optimization objectives
[26]	Large consumption of core network bandwidth and service delay	A new edge computing strategy for vehicle social communication is proposed	Optimize the allocation of available resource and improve the experience and service quality
[27]	Computational resource management	A fast convergent real-time repurchase scheme for mobile edge server is proposed	Reduce energy consumption and calculation delay
[28]	Overall resource scheduling	This paper proposes a fuzzy-based micro-service computing resource expansion algorithm	Reduce energy consumption
[29]	Manage edge node issues	An edge node resource management framework is built, and a mechanism for configuring and automatically scaling edge node resource is proposed	Reduce delay and improve transmission rate
[30]	Dynamic task scheduling	A task scheduling strategy suitable for edge computing scenario is proposed	Obtain the optimal task scheduling results
[31]	Energy-saving resource management	An energy-saving resource management strategy for asynchronous mobile edge computing offload system is proposed	Minimize total mobile energy consumption
[31]	Resource optimization problem	A resource management scheme based on multi-agent depth deterministic strategy gradient is proposed	Improve resource utilization

Appendix B.Task offloading based on MEC

References	Problems	Solution	Optimization objectives
[37]	Multidimensional optimization of offloading strategy formulation, load balancing, and computing resource allocation	A low complexity heuristic algorithm is proposed to obtain the offloading strategy and ensure the load balance among multiple MECs. Lagrange dual decomposition method is used to solve the computational resource allocation sub-problem	Minimize the weighted sum of total delay and energy consumption of all devices in MEC network
[38]	Base station user matching and power allocation	A cache-based NOMA downlink heterogeneous network architecture is proposed, and the messaging user coordination strategy is formulated to reasonably allocates	Improve network cache hit rate and minimize energy consumption
[39]	Calculate privacy disclosure during uninstall	A wireless communication and computing model of partial computing offload and resource allocation considering time-varying channel state, bandwidth constraints, random arrival of workload, and privacy protection is established, and a scheme of local computing offload and resource allocation based on distributed reinforcement learning is proposed	Minimize the calculation and offloading delay and energy consumption, while protecting users' privacy
[40]	Task calculation offloading problem	A new MEC edge computing offload system based on D2D communication is proposed, and the problem of computing offload is solved by game theory	In a multi-user scenario, maximize the number of users who reach a beneficial state through task offloading
[41]	High offloading failure rate	This paper presents a policy calculation offloading algorithm based on double depth Q-network	The offloading failure rate and energy consumption of offloading are reduced
[42]	Low efficiency of task offloading	A task layout algorithm is proposed, and an effective task offloading scheme is given	Improve the efficiency of task offloading
[43]	Low utilization of node computing power	A layered computing offload strategy is proposed to realize the allocation of computing resource between edge computing nodes and computing service subscribers	Improve the computing power of nodes and allocate resource reasonably

REFERENCES

[1] H. Zhang et al., “An offloading mechanism based on software defined network and mobile edge computing in vehicular networks,” J. Electron. Inf. Technol., vol. 42, pp. 645–652, Mar. 2020.

[2] M. Muhammad, “Survey on existing authentication issues for cellular-assisted v2x communication,” Veh. Commun., vol. 12, pp. 50–65, Mar. 2018.

[3] A. Islam, A. Debnath, M. Ghose, and S. Chakraborty, “A survey on task offloading in multi-access edge computing,” J. Syst. Archit., vol. 118, pp. 1–16, Sep. 2021.

[4] Y Mao et al., “A survey on mobile edge computing, the communication perspective,” IEEE Commun. Surv. Tutor., vol. 19, no. 4, pp. 2322–2358, 2017.

[5] W. Haoxin et al., “Architectural design alternatives based on cloud/edge/fog computing for connected vehicles,” IEEE Commun. Surv. Tutor., vol. 22, pp. 2349–2377, Apr. 2020.

[6] R. Xie et al., “Collaborative vehicular edge computing networks, architecture design and research challenges,” IEEE Access, vol. 7, pp. 178942–178952, 2019.

[7] S. Ali, G. A. Mostafa, and S. Ali, “A survey on the computation offloading approaches in mobile edge computing, a machine learning-based perspective,” Comput. Netw., vol. 182, pp. 1–24, 2020.

[8] X. Chen et al., “Age of information aware radio resource management in vehicular networks, a proactive deep reinforcement learning perspective,” IEEE Trans. Wireless Commun., vol. 19, no. 4, pp. 2268–2281, Apr. 2020.

[9] S. Wei et al., “An integrated longitudinal and lateral vehicle following control system with radar and vehicle-to-vehicle communication,” IEEE Trans Veh. Technol., vol. 68, no. 2, pp. 1116–1127, Feb. 2019.

[10] S. Olariu, “A survey of vehicular cloud research, trends, applications and challenges,” IEEE Trans. Intell. Transp. Syst., vol. 21, pp. 2648–2663, Jun. 2020.

[11] J. Zhao et al., “Edge caching and computation management for real-time internet of vehicles: an online and distributed approach,” IEEE Trans. Intell. Transp. Syst., vol. 22, pp. 2183–2197, Apr. 2021.

[12] M. Dai et al., “A computation offloading and resource allocation mechanism based on minimizing devices energy consumption and system delay,” J. Electron. Inf. Technol., vol. 41, pp. 2684–2690, Nov. 2019.

[13] Z. Ning et al., “Deep reinforcement learning-based offloading scheduling for vehicular edge computing,” IEEE Internet Things J., vol. 7, pp. 5449–5465, Jun. 2020.

[14] X. Zheng et al., “Blockchain-based secure computation offloading in vehicular networks,” IEEE Trans. Intell. Transp. Syst., vol. 22, pp. 40736–7087, Jun. 2021.

[15] G. Qiao et al., “Collaborative task offloading in vehicular edge multi-access networks,” IEEE Commun. Mag., vol. 56, pp. 48–54, Aug. 2018.

[16] J. Zhao et al., “Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks,” IEEE Trans. Veh. Technol., vol. 68, no. 8, pp. 7944–7956, Aug. 2019.

[17] S. Raza et al., “An efficient task offloading scheme invehicular edge computing,” J. Cloud Comput., vol. 9, pp. 1–14, Jun. 2020.

[18] Y. Liu et al., “Deep reinforcement learning for offloading and resource allocation in VEC and networks,” IEEE Trans. Veh. Technol., vol. 68, no. 11, pp. 11158–11168, Nov. 2019.

[19] Y. Dai et al., “Artificial intelligence empowered edge computing and caching for internet of vehicles,” IEEE Wireless Commun., vol. 26, pp. 12–18, Mar. 2019.

[20] L. T. Tan and R. Q. Hu, “Mobility-aware edge caching and computing in vehicle networks, a deep reinforcement learning,” IEEE Trans. Veh. Technol., vol. 67, pp. 10190–10203, Nov. 2014.

[21] Z. Zhang et al., “Proactive caching for vehicular multi-view 3D video streaming via deep reinforcement learning,” IEEE Trans. Wireless Commun., vol. 18, pp. 2693–2706, May 2019.

[22] G. Qiao et al., “Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks,” IEEE Internet Things J., vol. 7, pp. 247–257, Jan. 2020.

[23] L. Liang, H. Ye, G. Y. Li, “Spectrum sharing in vehicular networks based on multi-agent reinforcement learning,” IEEE J. Sel. Areas Commun., vol. 37, pp. 2282–2292, Oct. 2019.

[24] J. Warley et al., “Supporting mobility-aware computational offloading in mobile cloud environment,” J. Netw. Comput. Appl., vol. 94, pp. 93–108, 2017.

[25] A. Boukerche, “An efficient mobility-oriented retrieval protocol for computation offloading in vehicular edge multi-access network,” IEEE Trans. Intell. Transp. Syst., vol. 21, pp. 2675–2688, Jun. 2020.

[26] F. Lin et al., “A novel utility based resource management scheme in vehicular social edge computing,” IEEE Access, vol. 6, pp. 66673–66684, 2018.

[27] Y. Zhang et al., “Efficient computation resource management in mobile Edge-Cloud computing,” IEEE Internet Things J., vol. 6, pp. 3455–3466, Feb. 2019.

[28] X. Li et al., “A hybrid computing solution and resource scheduling strategy for edge computing in smart manufacturing,” IEEE Trans. Ind. Inf., vol. 15, pp. 4225–4234, Jul. 2019.

[29] N. Wang et al., “ENORM, a framework for edge node resource management,” IEEE Trans. Serv. Comput., vol. 13, pp. 1086–109, Jul. 2020.

[30] R. Wang et al., “Social attribute aware task scheduling strategy in edge computing,” J. Electron. Inf. Technol., vol. 42, pp. 271–278, Jan. 2020.

[31] C. You et al., “Asynchronous mobile-edge computation offloading, pp. Energy-efficient resource management,” IEEE Trans. Wireless Commun., vol. 7, pp. 7590–7605, Nov. 2018.

[32] H. Peng and X. Shen, “Multi-agent reinforcement learning based resource management in MEC and UAV-assisted vehicular networks,” IEEE J. Sel. Areas Commun., vol. 39, no. 1, pp. 131–141, Jan. 2021.

[33] J. Zhang et al., “Task offloading in vehicular edge computing networks, a load-balancing solution,” IEEE Trans. Veh. Technol., vol. 69, pp. 2092–2104, Feb. 2020.

[34] Y. Zhan et al., “A deep reinforcement learning based offloading game in edge computing,” IEEE Trans. Comput., vol. 69, pp. 883–893, Jun. 2020.

[35] W. Tang et al., “An offloading method using decentralized P2P-enabled mobile edge servers in edge computing,” J. Syst. Archit., vol. 94, pp. 1–13, 2019.

[36] J. Wang et al., “Fast adaptive task offloading in edge computing based on meta reinforcement learning,” IEEE Trans. Parallel Distrib. Syst., vol. 32, no. 1, pp. 242–253, 2021.

[37] S. L. Li et al., “Task offloading load balancing and resource allocation in MEC networks,” IET Commun., vol. 14, pp. 1451–1458, Sep. 2020.

[38] H. Zhang et al., “Resource allocation in NOMA heterogeneous network based on MEC,” J. Commun., vol. 41, pp. 27–33, Apr. 2020.

[39] H. C. Ke et al., “Deep reinforcement learning-based computation offloading and resource allocation in security-aware mobile edge computing,” Wireless Netw., vol. 23, pp. 3357–3373, May 2021.

[40] X. Ding and W. Zhang, “Computing offload strategy of massive internet of things devices based on game theory in mobile edge computing,” Math. Probl. Eng., vol. 2021, pp. 1–12, 2021.

[41] X. Chen et al., "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," IEEE Internet Things J., vol. 6, pp. 4005–4017, Mar. 2019.

[42] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software-defined ultra-dense network," IEEE J. Sel. Areas Commun., vol. 36, pp. 587–597, Mar. 2018.

[43] C. Su et al., "Computation offloading in hierarchical multi-access edge computing based on contract theory and Bayesian matching game," IEEE Trans. Veh. Technol., vol. 69, pp. 13686–13701, Nov. 2020.

[44] Y. Huang et al. "Research on mobile edge computing and caching in massive wireless communication network," J. Commun., vol. 42, pp. 44–61, Apr. 2021.

[45] A. Boukerche, "Computation offloading and retrieval for vehicular edge computing, algorithms, models, and classification," ACM Comput. Surv., vol. 53, no. 4, pp. 1–35, 2020.

[46] H. Liu et al., "Blockchain empowered cooperative authentication with data traceability in vehicular edge computing," IEEE Trans. Veh. Technol., vol. 69, pp. 4221–4232, Apr. 2020.

[47] Y. Qian et al., "Privacy-aware service placement for mobile edge computing via federated learning," Inf. Sci., vol. 505, pp. 562–570, 2019.

[48] H. Shen et al., "P2TA, privacy-preserving task allocation for edge computing enhanced mobile crowd sensing," J. Syst. Archit., vol. 97, pp. 130–141, 2019.

[49] J. Wang et al., "Electromagnetic radiation based continuous authentication in edge computing enabled Internet of things," J. Syst. Archit., vol. 96, pp. 53–61, 2019.

[50] R. Yang et al., "Integrated blockchain and edge computing systems, a survey, some research issues and challenges," IEEE Commun. Surv. Tutor., vol. 21, pp. 1508–1532, 2019.

[51] M. Zhaofeng et al., "A blockchain-based trusted data management scheme in edge computing," IEEE Trans. Ind. Inf., vol. 16, no. 3, pp. 2013–2021, 2020.

[52] J. Liu et al., "Cross-heterogeneous domain authentication scheme based on blockchain," J. Artif. Intell. Technol., vol. 2, pp. 92–100, 2021.

[53] S. Guo et al., "Blockchain meets edge computing, a distributed and trusted authentication system," IEEE Trans. Ind. Inf., vol. 16, pp. 1972–1983, Mar. 2020.

[54] M. H. U. Rehman et al., "Towards blockchain-based reputation-aware federated learning," IEEE Commun. Lett., vol. 24, no. 6, pp. 183–188, Feb. 2020.

[55] S. Deng et al., "Edge intelligence, pp. the confluence of edge computing and artificial intelligence," IEEE Internet Things J., vol. 7, pp. 1–13, Aug. 2020.

[56] E. Li, Z. Zhou, X. Chen, "Edge intelligence, pp. on-demand deep learning model co-Inference with device-edge synergy," ACM Sig-comm., vol. 4, pp. 31–36, Dec. 2018.

[57] J. Zhu et al., "An edge computing platform of guide-dog robot for visually impaired," 2019 IEEE 14th Int. Symp. Autonom. Decentral. Syst., Institute of Electrical and Electronics Engineers (IEEE), pp. 1–7, Apr. 2019.

[58] X. Lu et al., "An asynchronous federated learning mechanism for edge network computing," Comput. Res. Dev., vol. 57, pp. 2571–2582, 2020.

[59] S. Xu, Y. Qian, R. Q. Hu, "Data-driven edge intelligence for robust network anomaly detection," IEEE Trans. Netw. Sci. Eng., vol. 7, pp. 148–1492, Mar. 2020.

[60] Z. Zhou et al., "Edge intelligence, paving the last mile of artificial intelligence with edge computing," With Edge Comput. Proc. IEEE, vol. 107, pp. 1738–1762, Aug. 2019.

[61] Y. Ye et al., "Edge fed, pp. optimized federated learning based on edge computing," IEEE Access, vol. 8, pp. 209191–209198, 2020.

[62] J. Zhang et al., "FedMEC, pp. improving the efficiency of differentially private federated learning via mobile edge computing," Mobile Netw. Appl., vol. 25, no. 6, pp. 2421–2433, 2020.

[63] Y. Lu et al., "Low-latency federated learning and blockchain for edge association in digital twin empowered 6G network," IEEE Trans. Ind. Inf., vol. 17, pp. 5098–5107. July 2021.

[64] C.-H. Chu, "Task offloading based on deep learning for block chain in mobile edge computing," Wireless Netw., vol. 13, pp. 17–127, 2021.

[65] Z. Shahbazi and Y.-C. Byun, "Improving transactional data system based on an edge computing–blockchain– machine learning integrated framework," Processes, vol. 9, no. 1, pp. 1–20, Jan. 2021.

[66] X. Wang et al., "Convergence of edge computing and deep learning, pp. a comprehensive survey," IEEE Commun. Surv. Tutor., vol. 22, pp. 869–904. Feb. 2020.