Edge Computing for Real-Time Decision Making in Autonomous Driving: Review of Challenges, Solutions, and Future Trends

Jihong XIE, Xiang ZHOU, Lu CHENG*
School of Automotive, Wuhan Technical College of Communications, Wuhan 430065, China

***Abstract*—In the coming half-century, autonomous vehicles will share the roads alongside manually operated automobiles, leading to ongoing interactions between the two categories of vehicles. The advancement of autonomous driving systems has raised the importance of real-time decision-making abilities. Edge computing plays a crucial role in satisfying this requirement by bringing computation and data processing closer to the source, reducing delay, and enhancing the overall efficiency of autonomous vehicles. This paper explores the core principles of edge computing, emphasizing its capability to handle data close to its origin. The study focuses on the issues of network reliability, safety, scalability, and resource management. It offers insights into strategies and technology that effectively handle these challenges. Case studies demonstrate practical implementations and highlight the real-world benefits of edge computing in enhancing decision-making processes for autonomous vehicles. Furthermore, the study outlines upcoming trends and examines emerging technologies such as artificial intelligence, 5G connectivity, and innovative edge computing architectures.**

Keywords—Edge computing; autonomous driving; real-time decision-making; reliability; resource management

I. Introduction

A. Context

As transportation technology advances, several car manufacturers plan to deliver new cars to the market. The vehicles include several technologies, such as Electric Vehicles (EVs) [1], Autonomous Vehicles (AVs) [2], and Connected Vehicles (CVs) [3]. Autonomous driving technology has gained significant attention in the transportation sector, attracting considerable interest in the academic community [4]. Recent research suggests that autonomous driving systems have a role in regulating speed and making decisions, which may impact traffic safety and effectiveness. CVs are automobiles equipped with different communication technologies to interact with the driver, cloud (V2C), roadside infrastructure (V2I), and other vehicles (V2V) [5].

The U.S. Department of Transportation's National Highway Traffic Safety Administration (NHTSA) defines driverless or fully automated cars as vehicles lacking direct driver input while potentially controlling critical safety functions like braking, throttle, and steering [6]. This categorization aligns with the framework established by the Society of Automotive Engineers (SAE), which outlines six degrees of automation for autonomous vehicles, ranging from level 0 with no automated driving assistance to level 5 with complete automation [7].

B. Problem Statement

Real-time decision-making skills are a fundamental necessity for the safe and effective integration of self-driving cars on our roads in the ever-changing field of autonomous driving [8, 9]. AVs depend on intricate systems of sensors and algorithms to understand and react to changing surroundings, unlike vehicles operated by humans [10]. The core of real-time decision-making is promptly analyzing a continuous data flow to enable immediate reactions to unexpected situations, possible dangers, and uncertain road conditions [11]. This skill is crucial when quick reactions may make the difference between a collision and preventing a calamity. Real-time decision-making enhances the flexibility of AVs in navigating difficult traffic circumstances. As the self-driving system advances, quick and reliable decision-making becomes crucial. Ensuring safety is crucial for vehicle passengers and other road users and for enhancing the efficiency and performance of autonomous systems on our roads [12].

C. Challenges

The fast progress of autonomous driving, driven by advancements in artificial intelligence and related machine learning technologies, offers substantial enhancements in road safety, traffic congestion alleviation, pollution mitigation, and a general rise in human well-being [13, 14]. AVs rely on extensive sensor data to interpret complex surroundings, derive valuable insights, and navigate constantly changing conditions [15]. Evolving technologies demand immediate responsiveness and efficient activity processing, including end-to-end decision-making, sensing, radar analysis, and visual object recognition [16]. On-board computing systems face challenges in meeting growing processing requirements while adhering to power supply and device space limitations. One of the significant challenges in edge computing for autonomous driving is network reliability. AVs must maintain consistent and robust connectivity to ensure continuous data flow between vehicles and edge servers, which is crucial for real-time decision-making. Interruptions in network connectivity can lead to delays and potentially hazardous situations.

Safety is another critical concern, as the systems must be resilient to cyber-attacks and ensure data integrity and privacy [17]. Ensuring that data is secure and that the system can withstand malicious attacks is paramount for the safe operation of AVs. Scalability poses a challenge as the number of autonomous vehicles increases. The infrastructure must be capable of handling a growing volume of data and processing

*Corresponding Author.

demands without compromising performance. Efficiently managing these resources is essential to prevent bottlenecks and ensure smooth operations. Resource management is also a vital aspect, encompassing the allocation of computational power, storage, and bandwidth. Optimal resource management ensures that the edge computing infrastructure can support the intensive computational tasks required by AVs while balancing power consumption and device constraints.

D. Motivation

Mobile Edge Computing (MEC) refers to a computing model that allows individuals, including automobiles and devices, to receive computing and storage capabilities via Edge Computing Servers (ECSs) situated at base stations near users [18]. This system effectively decreases the duration of network connections by moving traffic from central data centers to edge data centers [19]. MEC allows AVs to enhance energy efficiency and computation by assigning workloads to ECSs [20]. The surge in cars on roadways requires MEC to analyze a large quantity of received data, potentially resulting in network congestion. The rise of 5G wireless communications boosts the use of MEC thanks to high-speed 5G networks, which instantaneously enable quick data transfer between clients and edge servers [21]. MEC's strong processing power and minimal network delay make it suitable for developers and subscribers who want to outsource demanding computing tasks to ECSs. Combined with 5G, MEC efficiently facilitates automated driving by fulfilling intensive computation requirements and stringent response time constraints [22].

AVs are powered by edge computing systems that integrate real-time functions like navigation, sensing, planning, and command. This means that all the processing is done on the edge, eliminating the need for a traditional cloud computing system and reducing latency. This is essential for the real-time decision-making capabilities of AVs. Vehicles are connected to ECSs and finally to the cloud over current cellular networks. Furthermore, vehicles can connect to Road Side Units (RSUs) over cellular networks or Dedicated Short-Range Communications (DSRCs). Moreover, DSRC technologies enable V2V communication between vehicles.

E. Contributions

This study highlights the critical role of edge computing in facilitating timely decisions in autonomous driving. The main contribution is a thorough examination of the core ideas of edge computing and their direct application to the complex problems of AVs. The study comprehensively analyzes the network's reliability, security, scalability, and resource management, identifies significant challenges, and proposes strategic solutions and technological advancements. The study also presents successful applications of edge computing using detailed case studies and highlights concrete improvements to the decision-making processes for autonomous cars in practical situations. Future research areas and forecast trends are examined, focusing on emerging technologies such as artificial intelligence, 5G connectivity, and revolutionary edge computing architectures. The research focuses primarily on connecting theoretical principles with actual applications to guide the continuous development and integration of edge computing to improve the safety and effectiveness of autonomous driving systems.

F. Structure of the Paper

Remaining portions of the paper are organized as follows. Section II explores the origins and foundations of both edge computing and autonomous driving, offering essential context in order to comprehend how they cross. Section III provides a comprehensive analysis of several solutions and technologies pertaining to edge computing in autonomous driving. This section also emphasizes the importance of these solutions in tackling difficulties and enhancing system performance. Section IV delineates prospective avenues for future study in this domain, highlighting areas that need more investigation and innovation. Section V presents a summary of the main results of the study and points out the significance of edge computing in the advancement of autonomous driving technologies.

II. Backgrounds

A. Edge Computing

Cloud computing involves offering centralized and virtualized processing, storage, services, and application resources over the Internet [23]. This strategy separates services from the foundational infrastructure, removes initial expenses, and streamlines IT infrastructure management [24]. Cloud computing faces challenges in providing real-time autonomous driving services given its centralized nature and fixed locations far from clients. Autonomous driving demands minimal latency, consistent timing, enough bandwidth, and mobile services. Cloud computing-based paradigms have been suggested to fulfill the requirements of autonomous driving. Spanning Cloud Computing (SCC) is a strategy that distributes applications throughout many cloud data centers to enhance performance, availability, and cost-effectiveness [25].

Although SCC has advantages, it also brings up additional tasks like resource management and orchestration, necessitating security measures to safeguard decentralized resources. The Cloud-based Content Delivery Network (CCDN) utilizes regionally dispersed and pay-as-you-go cloud platforms to provide content as a service [26]. CCDNs like Amazon CloudFront, Google CDN, and Azure CDN enhance the Quality of Experience (QoE) by duplicating material near the user [27]. While SCCs and CCDNs have benefits, they do not fulfill the stringent demands of autonomous driving. To overcome these challenges, cloud-based services should be moved closer to the data source to enable local data processing, analysis, and filtering. Edge computing is the extension of computing capabilities from traditional cloud models to the edge of the network. Edge computing optimizes infrastructure efficiency by attaining low latency, decreasing backhaul load, facilitating mobility services, and enhancing service robustness [28].

Edge computing is a decentralized infrastructure resource paradigm where essential resources are strategically located near data sources or the edge. This method avoids transferring data to a central cloud, which decreases latency, jitter, and network core load [29]. It also enhances security by keeping data in on-premises infrastructure [30]. Researchers are still debating the definition and precise position of the edge. Some consider the edge IoT-connected devices with restricted resource capabilities that handle gathering data processing [31]. Others view the edge as a structure that transfers processing duties to data sources [32]. We believe the application delivery environment determines the network's edge and positioning. The edge functions as a rational limit that variations in data users and suppliers may alter. In connected automobiles, a car acts as an edge, handling data collecting and processing inside. Within multi-access edge computing, the Radio Access Network (RAN) functions as an edge computing unit that controls the processing of user data. In a CCDN, a server acts as an edge and handles user requests [33].

B. *Autonomous Driving*

The exponential rise in population has caused a surge in the number of automobiles, placing substantial strain on current transportation infrastructure, such as parking facilities and gas stations. The ongoing growth of vehicle traffic significantly contributes to transportation concerns, particularly environmental pollution, traffic accidents, and congestion [34]. Researchers are now working on building AVs to address and maybe eliminate the problems posed by human drivers. As shown in Fig. 1, AVs are intelligent agents equipped with various sensors attached to vehicles to sense their surroundings. AVs rely on sensors like radars and cameras instead of human drivers who depend on their senses, like sight and hearing. The efficacy of AVs relies significantly on the quality of their sensors. Sophisticated perception algorithms may not operate well if the sensor data is unreliable.

Fig. 1. AVs and sensor technology for intelligent transportation.

Vehicle sensors can be classified into two classes: proprioceptive sensors that measure the vehicle itself and exteroceptive sensors that measure the vehicle's surroundings. Cameras are commonly used in autonomous driving as passive sensors that gather light and capture precise information about a situation. Factors like resolution and area of view are critical factors in a camera's quality. The resolution, determined by the number of pixels composing the image, directly influences its quality. The horizontal and vertical angular visibility range governs the camera's area of vision. High dynamic range is essential for autonomous cars, particularly when navigating different illumination situations at night. Stereo cameras, comprised of two cameras with overlapping fields of view, provide a disparity map to evaluate depth in individual pixels.

LIDAR is a crucial sensor that emits laser beams to determine the depth of objects by analyzing reflected light and time of flight. LIDAR creates a 3D point cloud map and details the scene's geometry. Key LIDAR measurements consist of the number of sources, points gathered per second, and field of view.

RADAR sensors, used before LIDAR, effectively identify massive objects and are especially beneficial in challenging weather situations. RADAR sensors are defined by their detection range, field of view, and accuracy in measuring position and speed. Ultrasonic sensors, which use sound waves to determine distance, are crucial in situations like parking when vehicles must manoeuvre close to other automobiles.

Global navigation systems, especially GPS or Galileo, act as proprioceptive sensor, enabling the calculation of a vehicle's location, speed, and sometimes direction. The Inertial Measurement Unit (IMU) measures angular rotation rate and accelerations. The wheel odometer, a proprioceptive sensor, monitors the wheel's rotation speed to determine the speed and change in direction of the vehicle. Following an environmental evaluation, AVs perform object recognition, planning, decision-making, speed control, and driving independently without human involvement.

Autonomous cars, including both non-motorized and motorized vehicles, use sensors to detect and categorize things in the surroundings. It is essential to monitor the detected things and observe mobile objects on the road. Moreover, the vehicle requires sensor data for its precise positioning. Perceptual and localization data are integrated and transformed into a unified 3D coordinate system before being sent to the planning module. The planning component generates a continuous chain of directed paths from the starting point to the destination following motion, behavior, and route planning. The planning unit creates route information for vehicle self-regulation to adjust to various vehicle actions. The control commands are relayed to the braking, acceleration, and steering wheel elements of the vehicle's operating system for implementation. Fig. 2 depicts the software architecture of the whole system.

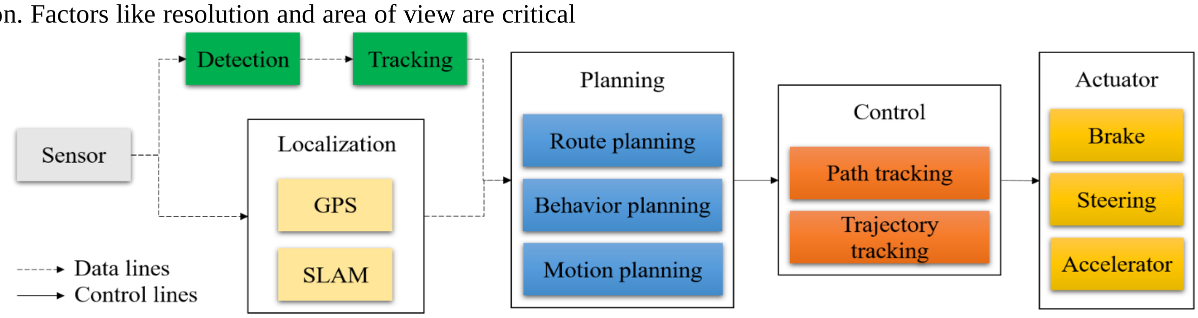

Fig. 2. General autonomous driving system.

Perception is a major problem in autonomous driving, but planning and control are also key challenges. Planning involves determining ideal routes and making decisions depending on the observed environment, which may be complicated under dynamic and unexpected traffic situations. Planning requires dealing with various road conditions, unanticipated actions of other road users, and guaranteeing both safety and efficiency in navigation. Control encompasses the implementation of planned activities by accurately managing vehicle characteristics, including acceleration, braking, and steering, in order to guarantee a seamless and secure operation. Developing reliable and effective planning and control algorithms is crucial for AVs to function safely and efficiently in a variety of conditions. These issues highlight the intricate nature of creating completely self-driving systems that can operate smoothly in real-life situations, requiring sophisticated algorithms and sensor technology to handle the many aspects of autonomous driving tasks.

III. Solutions and Technologies

This section delves into a diverse array of innovative solutions aimed at addressing the intricate challenges within autonomous driving through cutting-edge technologies and methodologies. Table I provides an insightful comparison of these solutions.

TABLE I. AN OVERVIEW OF AUTONOMOUS DRIVING SOLUTIONS

References	Objective	Methodology	Challenges	Findings / Results
[35]	Real-time autonomous robotic and vehicle applications	Runtime layer abstraction, heterogeneity-aware scheduling, and vehicle cloudlet coordinator	Energy efficiency and real-time processing	Deployment on Nvidia Jetson TX1 with power consumption under 11W
[36]	Resource offloading for autonomous driving functions	3-layer protocol: autonomous vehicle, network edge, and cloud computing	Latency reduction and resource optimization	Evaluation of optimization methods in various scenarios
[37]	Faster processing of autonomous driving tasks through dynamic offloading	ILP formulation and lookup table for real-time applications	Offline-online tradeoff and network state fluctuations	Significant enhancement in system performance
[38]	Efficient task scheduling for autonomous driving	Task assignment based on time limitations	Urgency and vulnerability consideration	Efficient scheduling of tasks and accommodating critical tasks
[39]	Leveraging computing resources within each vehicle	Optimization problem based on vehicle mobility	Vehicle mobility and breaks in connection	Enhanced reaction time by 34% compared to other techniques
[40]	Intelligent networking architecture using MEC	Dynamic driving model for each road segment	Generalization challenges and changing environmental conditions	Improved driving model for each road segment
[41]	Predicting QoS for autonomous driving services	Balancing exploration and exploitation, and maximizing discounted future reward	Conventional prediction model limitations	Attains highest performance under Autoware benchmark settings
[42]	Lyapunov optimization for task offloading in autonomous driving	Optimal target server selection based on system stability	System stability and time considerations	Steady queue backlog and efficient task processing
[43]	Allocating computing resources to reduce vehicle travel distance	Whittle index calculation using DRL approach	Delay in receiving computing results and changing vehicle mobility	Efficiently provides computational outcomes to cars
[44]	Using game theory for mutually beneficial outcomes in AVs	FPGA-accelerated calculating process	Combinatorial calculations and Nash equilibrium	2.4 times performance increase compared to CPU
[45]	Edge computing-based lanes scheduling system	Sesa and SVLSA centralized management lane scheduling approaches	Efficient crossing navigation, guaranteeing designated cars cross intersections	Outperforms other strategies in common lane-changing situations
[46]	Using deep learning for infotainment caching in AVs	Block-wise majorization-minimization approach for optimization	Caching choices based on passenger trains and reduction in content download delays	Prediction accuracy of 97.8%, effective time reduction
[47]	Container-based architecture for autonomous driving	Utility-focused greedy algorithm for offloading scheduling	Privacy preservation and resource segregation	Great practicality and isolation, millisecond edge relief

Tang, et al. [35] developed LoPECS, a low-power edge computing framework for real-time autonomous robotic and vehicle applications using cost-effective embedded technologies. A heterogeneity-sensitive runtime structure was created to optimize the use of the vehicle's diverse computing resources for autonomous driving applications. A vehicle edge coordinator was also developed to transfer vehicle tasks to edge cloudlets efficiently. These components were effectively integrated into the LoPECS system. The system was deployed on the Nvidia Jetson TX1, demonstrating its efficiency with a power consumption of under 11W. At the application tier, LoPECS provides obstacle detection, localization, voice recognition, and other features to provide secure, effective, and real-time driving behavior. The QoE Oriented Service Classification categorizes autonomous driving services based on real-time needs and energy costs. The real-time OS is a minimalistic operating system that efficiently oversees several services and facilitates their communication with little additional processing. The LoPECS runtime layer abstracts diverse computing resources and utilizes a heterogeneity-aware scheduling method to allocate tasks on heterogeneous hardware platforms. The vehicle cloudlet coordinator shifts tasks to the cloud in real-time to maximize energy economy, considering vehicle movement and cloud availability.

Ibn-Khedher, et al. [36] designed an end-to-end communication architecture that allows computationally demanding autonomous driving functions, such as Autopilot, to be allocated to distributed resources on edge computing infrastructure. This architecture aims to enhance the performance of autonomous driving vehicles by reducing latency and ensuring reliability. The architecture outlines an Advanced Autonomous Driving (AAD) connectivity protocol for AVs, edge computing servers, and the centralized cloud. An Integer Linear Programming (ILP) technique is used to create a mathematical model for resource offloading of the autopilot chain at the network edge. A Deep Reinforcement Learning (DRL) method is suggested for high-density Internet of Autonomous Vehicle (IoAV) networks. The AAD protocol has three primary layers or modules. The autonomous vehicle layer has to outsource autopilot service chains because of limited local resources. The distributed network edge layer serves as an intermediary connecting OBU cars to the cloud. The system comprises distributed edge servers that provide communication between vehicles or the virtualized OBUs. The task involves processing and evaluating externally provided Virtual Network Functions (VNFs) based on vehicle specifications and the edge servers' resources. The cloud computing tier functions as a cloud autopilot and handles non-real-time edge autopilot VNFs.

Cui, et al. [37] introduced a new method to transfer computationally demanding autonomous driving tasks to on-road equipment and the cloud for faster processing. The method integrates an ILP model for optimizing the planning approach offline with a quick heuristic technique for online adjustments. The suggested method is validated using both artificial task diagrams and practical implementations. The proposed approach contains an offline ILP solution and a rapid heuristic for online adjustment. Two factors justify this hybrid approach. While the ILP approach may provide ideal outcomes, the time needed to achieve a result is substantial and unsuitable for online changes. Conversely, the network state might be unstable, causing fluctuations in the bandwidth between OBU, edge, and cloud. It is necessary to modify the approach dynamically to accommodate the fluctuating network conditions.

An ILP formulation is needed for the offline phase to determine the best offloading method, which involves segmentation, scheduling, and allocation for the Directed Acyclic Graph (DAG) across all three platforms based on the provided topology characteristics. Optimal methods for potential network variables are computed and kept in a lookup table for real-time applications. The runtime is not affected by the time needed for ILP computation. The ODA scheduling method utilizes an offline scheduling methodology based on the network status to construct a new offloading strategy in a greedy manner. When the present network condition aligns with entries in the lookup table, the corresponding plan from the table is selected. Otherwise, the approach most similar to the network state will serve as the starting point for the greedy algorithm. This technique uses the smallest Euclidean distance between two network sets as a decision criterion. When the Euclidean distance is equal, the mobile network with the shortest distance is chosen. This procedure is triggered at runtime whenever there is a network status update.

Dai, et al. [38] suggested a task scheduling technique that accounts for the specific features of autonomous driving tasks. The system selects appropriate edge computing servers by using an enhanced early deadline first strategy, which involves task migration via replacement and recombination. The experimental findings indicate that the system can efficiently schedule a greater number of tasks as the task quantity grows, successfully accommodating critical tasks. The scheduling algorithm aims to allocate tasks efficiently by considering the urgent nature and vulnerability of autonomous driving duties, allowing for the execution of a greater number of tasks. The assignment of tasks to edge computing servers is established in accordance with the time limitations of the assignment. Upon arrival of a new task, BFRS estimates its deadline and then sequentially tries the alternative methods, such as the Task Replacement Strategy (TRS) and the Direct Execution Strategy (DES).

DES first verifies whether the task can be handled directly on the local edge server. If not, it fits whether the task can be performed directly on the other edge computing servers. It selects the most suitable edge computing server based on the Earliest Start Time (EST) or the Shortest Sufficiently Free Interval (SSFI) selected. If DES fails to address the problem, TRS first examines whether the task can be accomplished by replacing it with fewer computationally intensive tasks on the local edge computing server. Otherwise, other edge servers are iteratively checked and ordered according to free intervals. Lastly, if TRS cannot resolve the problem, BFRS returns that none of the edge computing servers can perform the recently arrived task.

Vehicular Edge Computing (VEC) is becoming more popular because it can decrease latency and alleviate the burden on backhaul networks. To address the rising computing needs of expanding vehicle applications, such as autonomous driving, sufficient computing resources within each vehicle can be vital for task execution in a VEC scenario, leading to enhanced user experience. However, the increased mobility of the vehicles makes this process difficult and might cause breaks in connection, leading to delays in current task processing. Liu, et al. [39] developed a task-shifting technique that utilizes multi-hop vehicle computation resources in VEC, according to the study of vehicle movement. An optimization problem is created with the objective of minimizing the combined weighted total of execution time and computing effort of all functions in a vehicle. A strategy using semi-definite relaxation with an adaptive adjustment technique is suggested to address the optimization issue and determine the unloading options. The simulation results demonstrate that the suggested offloading method may notably enhance reaction time by an average of 34% when compared to other techniques, such as local processing and random offloading.

AI-driven AVs may use a variety of machine learning methods to construct a sophisticated self-driving structure. One AV intelligence is insufficient to handle constantly changing driving conditions. Current neural network design and training techniques have challenges in generalizing driving models to varied contexts due to sampling inefficiency and the curse of dimensionality. Robust computational resources and extensive data may be used to train an effective driving model without the need for real-time operation. Nevertheless, the driving model derived offline may not be successful in some instances. Wu, et al. [40] proposed an intelligence networking architecture connecting AVs using multi-access edge computing and end-to-end learning for demonstration goals. This framework separates the trip and collects data individually for each road section. multi-access edge computing networks generate and update a dynamic driving model for each road segment in almost real-time to account for changing conditions. Segmenting the route decreases the requirement for generalization since it allows a single model to focus on adapting to a particular section. The simulation results demonstrate that the solutions provide an enhanced driving model for each road segment to more effectively adjust to environmental variations compared to the current method.

Xiong, et al. [41] proposed a learning method to predict the Quality of Service (QoS) of services in a multi-dimensional setting. They also created a reliable service delivery method that needs minimal hyperparameter adjustments and a small number of trials to learn multilayer neural network policies. This method can balance exploration and exploitation by modifying hyperparameters using maximum entropy gain learning. They demonstrated that this method attains the highest level of performance under Autoware benchmark settings. QoS prediction involves completing missing values. The conventional prediction model does not consider multi-faceted factors. Hence, they aim to use undisclosed aspects of the multi-dimensional environment to anticipate QoS. The method offers an independent service provisioning strategy that relies on QoS prediction. A neural network model is designed to accomplish this objective and is accountable for carrying out the actions defined by the model. The model, based on QoS, decides on the action to take and then receives a reward.

Jang, et al. [42] developed a new task offloading approach that utilizes Lyapunov optimization to ensure system stability and reduce task processing latency. A real-time monitoring mechanism is constructed to optimally use distributed computing resources in an autonomous driving setting. An analysis of computational complexity and memory access rate is conducted to demonstrate the features of deep learning applications for the task offloading technique. Lyapunov and Lagrangian optimization addresses the balance between system stability and user needs. The process of outsourcing complicated applications involves real-time monitoring, workload analysis, and optimum decision-making. The offloading algorithm selects the optimal target server based on system stability and time for processing the required task.

Li, et al. [43] studied the allocation of computing resources for real-time tasks in autonomous driving. In the presented scenario, AVs consistently capture the surroundings, transmit sensor data to an edge server for analysis, and receive processed findings from the server. A vehicle has a delay in receiving calculating results due to motion and processing latency, resulting in a distance covered between storing sensor input and receiving the results. The objective is to develop an edge separation planning strategy to reduce the distance travelled by vehicles. The method involves establishing the sequence of processing based on the mobility of each vehicle and the computational capacity of the edge server. They developed a Restless Multi-Arm Bandit (RMAB) issue, created a stochastic scheduling strategy based on the Whittle index, and calculated the index using a DRL approach. The suggested planning scheme circumvents the lengthy policy exploration often seen in DRL planning methods and efficiently produces judgments with little complexity.

Du, et al. [44] investigated the potential of using game theory in decision-making to create a mutually beneficial outcome for AVs. The Lemke-Howson method in game theory is a well-known combinatorial technique used to compute a Nash equilibrium of a bimatrix game. They developed the Lemke-Howson algorithm using FPGA to expedite the calculating process. The host side creates test data, transmits it to the FPGA card, and collects the outcomes. The host and the FPGA interact over the PCIe port using the AXI4-Stream protocol. The FPGA incorporates the Lemke-Howson accelerator and the DMA subsystem for the PCIe-IP core. The accelerator can fully execute the algorithm's functions. The IP core's DMA mode is set up to transfer data bidirectionally between host memory and FPGA using a PCIe bridge. The driver on the host enables sending instructions to link the host to the FPGA and initiate data transmission. The Lemke-Howson accelerator was applied to a KCU116 board, resulting in a performance increase of around 2.4 times compared to operating on a CPU.

Xia, et al. [45] proposed a new Edge Computing-based Lanes Scheduling System (ECLSS) model to analyze lane assignment for vehicles at junctions using real-time edge devices. Multiple edge computing devices are deployed at junctions to gather data from cars and road conditions via short-range wired or wireless transfers. Two centralized management lane scheduling approaches, the Search for Efficient Switching Algorithm (SESA) and the Special Vehicles Lane Switching Algorithm (SVLSA), are developed, given the strong computational power and real-time transmission performance of edge devices. These edge computing-driven autonomous driving techniques focus on efficiently navigating crossings and guaranteeing that designated cars may cross intersections within a certain timeframe. Comprehensive simulations were carried out, showing that the suggested methods outperform other strategies in common lane-changing situations.

Ndikumana, et al. [46] suggested using infotainment caching in autonomous vehicles. This system would make caching choices by analyzing passenger traits using deep learning. Initially, deep learning models are proposed to forecast the material that should be stored in multi-access edge computing servers in self-driving cars and vehicles near self-driving cars linked to roadside units. Secondly, a communication strategy is introduced for accessing stored infotainment material. Thirdly, a caching mechanism is offered for stored material. A cached content computation model is presented that may be delivered in various forms and quality according to the level of demand. An optimization problem is introduced that integrates the suggested models to reduce content download delays. The issue is solved using a block-wise majorization-minimization approach.

Tang, et al. [47] proposed a container-based edge-offloading architecture for autonomous driving. It constructs an offloading decision module, an offloading scheduler module, and an edge offloading middleware using lightweight virtualization. It offers abstraction and control of the execution environment at the level of containers at the edge. Thus, it enables the preservation of privacy and the segregation of resources from limitations during autonomous driving operations. They codified the mapping challenge of many applications on multiple edge nodes into a multi-dimensional knapsack challenge with his utility-preferred offloading scheduling approach. A utility-focused greedy algorithm was presented for the immediate resolution. The proposed system consists of three agents: AVs, edge servers, and a node coordinator. Service middleware for offloading is installed on edge servers to handle service offloading dispatch promptly. The middleware utilizes containers to separate the operational environment, application data, and hardware resources in order to guarantee the security of external applications. The middleware can modify the allocation of container resources based on the requirements of external applications to optimize performance. The node coordinator manages several edge servers within an appropriate range.

IV. Result and Discussion

In this section, we present and analyze the findings from our review and case studies of edge computing in autonomous driving. The results highlight the effectiveness of current solutions in addressing key challenges such as network reliability, safety, scalability, and resource management. Our review indicates that edge computing significantly enhances network reliability by reducing latency and ensuring robust connectivity. Studies have shown that deploying edge servers closer to the data source minimizes the risk of communication delays and interruptions, which is crucial for real-time decision-making in autonomous vehicles.

Edge computing contributes to improved safety by enabling faster and more secure data processing. The localized processing power of edge servers allows for immediate response to potential hazards, reducing the likelihood of accidents. Additionally, security measures integrated into edge computing frameworks protect against cyber-attacks, ensuring data integrity and privacy. The scalability of edge computing solutions is demonstrated through various case studies where the infrastructure efficiently handled increasing data volumes and processing demands. The flexibility of edge computing allows for seamless integration with advanced technologies such as 5G, further enhancing its capacity to support a growing number of autonomous vehicles.

Effective resource management is a key advantage of edge computing. By offloading computational tasks to edge servers, autonomous vehicles can optimize their onboard resources, leading to better energy efficiency and performance. Our analysis shows that edge computing frameworks can dynamically allocate resources based on real-time needs, ensuring optimal utilization. The rapid development of autonomous driving, coupled with the transformative influence of edge computing, opens up opportunities for interesting future research directions. This section discusses challenges, open issues, and future directions in adopting edge computing in autonomous vehicles.

- Integration with advanced AI techniques: Exploring enhanced synergy between edge computing and AI algorithms presents an exciting prospect. Integrating advanced AI techniques, such as machine learning and deep neural networks, with edge computing can bolster the decision-making capabilities of autonomous vehicles, making them more adept at navigating complex scenarios [48, 49].
- Optimal 5G integration for communication: The burgeoning landscape of 5G connectivity holds promise for enhancing communication between AVs and edge computing infrastructure. Investigating the optimal integration of 5G networks to facilitate low-latency, high-bandwidth data exchange is imperative for maximizing the potential of edge computing in real-time decision-making.
- Innovative edge computing architectures: The design and development of innovative edge computing architectures represent a current research focus. These architectures are envisioned to handle diverse data sources efficiently, prioritize safety-critical tasks, and dynamically scale to accommodate the evolving complexity of autonomous systems. This research aims to provide a robust foundation for the seamless operation of edge computing in the intricate landscape of autonomous driving.
- Addressing ethical implications: The rise of autonomous decision-making at the edge brings forth ethical considerations that demand careful examination. Interdisciplinary research actively manages ethical implications, including establishing accountability mechanisms, ensuring transparency, and formulating ethical frameworks that guide the responsible deployment of autonomous driving technologies.
- Ensuring security and privacy: As edge computing plays a pivotal role in processing sensitive data, ongoing investigations are directed toward implementing robust security measures. This research is crucial for safeguarding data processed at the edge, countering potential vulnerabilities, and ensuring users' privacy within autonomous driving systems.
- Human-machine interaction enhancement: Prioritizing the seamless integration of AVs into mixed traffic environments, research efforts are dedicated to improving human-machine interaction through edge computing solutions. This encompasses developing interfaces and communication strategies that enhance understanding and cooperation between AVs and human drivers or pedestrians.
- Environmental impact assessment: Researchers are actively conducting environmental impact assessments to evaluate the ecological footprint of edge computing in autonomous driving. This includes considerations of energy consumption, sustainability, and the implementation of eco-friendly optimizations to minimize the environmental impact of these emerging technologies.

- Standardization and interoperability: A key focus in ongoing research is the development of industry standards for edge computing in autonomous driving. This effort aims to ensure interoperability among diverse systems, fostering a cohesive and standardized approach that enhances the compatibility and seamless integration of autonomous driving technologies.
- Distributed edge computing models: Expanding beyond conventional edge computing models, current research explores distributed edge computing architectures. This approach involves decentralizing computing resources across interconnected edge devices, contributing to enhanced scalability, reduced latency, and improved fault tolerance in autonomous driving environments.
- Edge-to-edge collaboration: Investigating edge-to-edge collaboration is a burgeoning area of interest. Researchers are exploring how different edge devices within the autonomous ecosystem can collaborate effectively. This collaborative approach aims to distribute computational loads efficiently, optimize resource utilization, and improve overall system performance.
- Context-aware edge processing: Research is underway to develop context-aware edge processing capabilities to enhance the contextual understanding of autonomous vehicles. This involves tailoring edge computing algorithms to consider specific environmental factors, traffic conditions, and other contextual nuances, thereby refining decision-making processes in real time.
- Edge-cloud integration: Addressing the balance between edge and cloud computing, ongoing research focuses on effective integration strategies. This involves leveraging the strengths of both edge and cloud computing to optimize resource utilization, scalability, and data processing efficiency in autonomous driving scenarios.
- Edge analytics for predictive maintenance: Expanding the role of edge computing, researchers are exploring its application in predictive maintenance for autonomous vehicles. By implementing edge analytics, the system can proactively identify and address potential hardware or software issues, thereby enhancing the reliability and longevity of autonomous driving systems.
- Adaptive edge resource allocation: To ensure efficient resource management, research is dedicated to developing adaptive edge resource allocation mechanisms. This involves dynamically allocating computing resources based on the varying demands of different tasks, contributing to improved overall system performance and responsiveness.
- User-centric edge services: With a focus on user experience, current research explores the development of user-centric edge services. This entails tailoring edge computing capabilities to meet users' specific needs and preferences, creating a more personalized and adaptive autonomous driving experience.
- Edge-based anomaly detection: Enhancing the security posture of autonomous systems, researchers are investigating edge-based anomaly detection techniques. The system can identify abnormal patterns or behaviors by analyzing data at the edge, enabling rapid responses to potential cybersecurity threats and ensuring the integrity of autonomous driving operations.

V. Conclusion

This paper has provided a comprehensive exploration of the pivotal role played by edge computing in advancing real-time decision-making capabilities within the domain of autonomous driving. The examination of core principles, challenges, and solutions underscored the significance of bringing computation and data processing closer to the source, mitigating latency, and enhancing overall system efficiency. The case studies presented exemplified successful implementations of edge computing, demonstrating tangible improvements in decision-making processes for AVs across various scenarios. The outlined future research directions shed light on the evolving landscape, emphasizing the integration of advanced AI techniques, optimal 5G connectivity, innovative edge computing architectures, and ethical considerations. As autonomous driving continues to evolve, it is evident that edge computing will remain a linchpin, shaping the trajectory of intelligent, adaptive, and safe self-driving vehicles on our roads. The interplay between edge computing and emerging technologies, coupled with ongoing interdisciplinary collaborations, sets the stage for a future where autonomous systems not only navigate complex environments seamlessly but also prioritize safety, security, and ethical considerations in their decision-making processes. This synthesis of theoretical insights and practical applications underscores the transformative potential of edge computing in defining the future of autonomous driving.

Acknowledgment

This work was supported by project of Research on Thermal Failure Mechanism of Lithium Battery under Multi Physical Parameter Model (No. B2022593).

References

[1] Pramuanjaroenkij and S. Kakaç, "The fuel cell electric vehicles: The highlight review," International Journal of Hydrogen Energy, vol. 48, no. 25, pp. 9401-9425, 2023.

[2] X. Zhao, Y. Fang, H. Min, X. Wu, W. Wang, and R. Teixeira, "Potential sources of sensor data anomalies for autonomous vehicles: An overview from road vehicle safety perspective," Expert Systems with Applications, p. 121358, 2023.

[3] Y. Xue, L. Wang, B. Yu, and S. Cui, "A two-lane car-following model for connected vehicles under connected traffic environment," IEEE Transactions on Intelligent Transportation Systems, 2024.

[4] M. Bargahi, H. Barati, and A. Yazici, "Relationship between Criticality and Travel Time Reliability in Transportation Networks," in 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), 2023: IEEE, pp. 2479-2484, doi: https://doi.org/10.1109/ITSC57777.2023.10421885.

[5] Z. Mahmood, "Connected vehicles in the IoV: Concepts, technologies and architectures," in Connected vehicles in the internet of things: concepts, technologies and frameworks for the IoV: Springer, 2020, pp. 3-18.

[6] A. Chougule, V. Chamola, A. Sam, F. R. Yu, and B. Sikdar, "A Comprehensive review on limitations of autonomous driving and its impact on accidents and collisions," IEEE Open Journal of Vehicular Technology, 2023.

[7] D. Hopkins and T. Schwanen, "Talking about automated vehicles: What do levels of automation do?," Technology in Society, vol. 64, p. 101488, 2021.

[8] A. A. Anvigh, Y. Khavan, and B. Pourghebleh, "Transforming Vehicular Networks: How 6G can Revolutionize Intelligent Transportation?," Science, Engineering and Technology, vol. 4, no. 1, 2024.

[9] S. Jaferian and M. Rezvani, "Export New Product Success: The Impact of Market and Technology Orientation," International Journal of Management, Accounting & Economics, vol. 1, no. 5, 2014.

[10] K. Pal, P. Yadav, and N. Katal, "RoadSegNet: a deep learning framework for autonomous urban road detection," Journal of Engineering and Applied Science, vol. 69, no. 1, pp. 1-21, 2022.

[11] W. Wang, T. Qie, C. Yang, W. Liu, C. Xiang, and K. Huang, "An intelligent lane-changing behavior prediction and decision-making strategy for an autonomous vehicle," IEEE transactions on industrial electronics, vol. 69, no. 3, pp. 2927-2937, 2021.

[12] T. N. Hoang, P. P. Hong, N. N. Vinh, N. T. Nguyen, K. H. Nguyen, and L.-D. Quach, "An Improved Lane-Keeping Controller for Autonomous Vehicles Leveraging an Integrated CNN-LSTM Approach," International Journal of Advanced Computer Science and Applications, vol. 14, no. 7, 2023.

[13] S. Olugbade, S. Ojo, A. L. Imoize, J. Isabona, and M. O. Alaba, "A review of artificial intelligence and machine learning for incident detectors in road transport systems," Mathematical and Computational Applications, vol. 27, no. 5, p. 77, 2022.

[14] S. R. Abdul Samad et al., "Analysis of the performance impact of fine-tuned machine learning model for phishing URL detection," Electronics, vol. 12, no. 7, p. 1642, 2023.

[15] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," Sensors, vol. 21, no. 6, p. 2140, 2021.

[16] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues," Array, vol. 10, p. 100057, 2021.

[17] M. Zarei, O. Zarei, M. Karimi, M. R. Skandari, M. Haghighatjoo, and M. W. Khordehbinan, "The application of multi-criteria decision analysis in gaining a premier sort of stability in airplane safety," in Safety and Reliability, 2024: Taylor & Francis, pp. 1-16, doi: https://doi.org/10.1080/09617353.2024.2303699.

[18] B. Pourghebleh and N. Jafari Navimipour, "Towards efficient data collection mechanisms in the vehicular ad hoc networks," International Journal of Communication Systems, vol. 32, no. 5, p. e3893, 2019.

[19] C. Caiazza, S. Giordano, V. Luconi, and A. Vecchio, "Edge computing vs centralized cloud: Impact of communication latency on the energy consumption of LTE terminal nodes," Computer Communications, vol. 194, pp. 213-225, 2022.

[20] L. P. Qian, Y. Wu, N. Yu, D. Wang, F. Jiang, and W. Jia, "Energy-efficient multi-access mobile edge computing with secrecy provisioning," IEEE Transactions on Mobile Computing, vol. 22, no. 1, pp. 237-252, 2021.

[21] Y. Liu, M. Peng, G. Shou, Y. Chen, and S. Chen, "Toward edge intelligence: Multiaccess edge computing for 5G and Internet of Things," IEEE Internet of Things Journal, vol. 7, no. 8, pp. 6722-6747, 2020.

[22] A. Vaishnavi, E. Naresh, and G. Vijay, "Performance Evaluation of Mobile Edge Computing using 5G Networks," in 2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 2021: IEEE, pp. 1-6.

[23] B. Pourghebleh, A. A. Anvigh, A. R. Ramtin, and B. Mohammadi, "The importance of nature-inspired meta-heuristic algorithms for solving virtual machine consolidation problem in cloud environments," Cluster Computing, pp. 1-24, 2021.

[24] V. Hayyolalam, B. Pourghebleh, M. R. Chehrehzad, and A. A. Pourhaji Kazem, "Single‐objective service composition methods in cloud manufacturing systems: Recent techniques, classification, and future trends," Concurrency and Computation: Practice and Experience, vol. 34, no. 5, p. e6698, 2022.

[25] M. M. Islam, F. Ramezani, H. Y. Lu, and M. Naderpour, "Optimal placement of applications in the fog environment: A systematic literature review," Journal of Parallel and Distributed Computing, vol. 174, pp. 46-69, 2023.

[26] T. Kurokawa and N. Hayashibara, "Content placement using Cuckoo search in Cloud-based Content delivery networks," Internet of Things, vol. 16, p. 100430, 2021.

[27] Z. Jin, L. Pan, and S. Liu, "Randomized online edge service renting: Extending cloud-based CDN to edge environments," Knowledge-Based Systems, vol. 257, p. 109957, 2022.

[28] J. Pan and J. McElhannon, "Future edge cloud and edge computing for internet of things applications," IEEE Internet of Things Journal, vol. 5, no. 1, pp. 439-449, 2017.

[29] Y. Mansouri and M. A. Babar, "A review of edge computing: Features and resource virtualization," Journal of Parallel and Distributed Computing, vol. 150, pp. 155-183, 2021.

[30] G. Carvalho, B. Cabral, V. Pereira, and J. Bernardino, "Edge computing: current trends, research challenges and future directions," Computing, vol. 103, pp. 993-1023, 2021.

[31] A. M. Ghosh and K. Grolinger, "Edge-cloud computing for Internet of Things data analytics: Embedding intelligence in the edge with deep learning," IEEE Transactions on Industrial Informatics, vol. 17, no. 3, pp. 2191-2200, 2020.

[32] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," IEEE access, vol. 8, pp. 85714-85728, 2020.

[33] B. Liang, M. A. Gregory, and S. Li, "Multi-access Edge Computing fundamentals, services, enablers and challenges: A complete survey," Journal of Network and Computer Applications, vol. 199, p. 103308, 2022.

[34] J. Lu, B. Li, H. Li, and A. Al-Barakani, "Expansion of city scale, traffic modes, traffic congestion, and air pollution," Cities, vol. 108, p. 102974, 2021.

[35] J. Tang, S. Liu, L. Liu, B. Yu, and W. Shi, "LoPECS: A low-power edge computing system for real-time autonomous driving services," IEEE Access, vol. 8, pp. 30467-30479, 2020.

[36] H. Ibn-Khedher et al., "Edge computing assisted autonomous driving using artificial intelligence," in 2021 International Wireless Communications and Mobile Computing (IWCMC), 2021: IEEE, pp. 254-259.

[37] M. Cui, S. Zhong, B. Li, X. Chen, and K. Huang, "Offloading autonomous driving services via edge computing," IEEE Internet of Things Journal, vol. 7, no. 10, pp. 10535-10547, 2020.

[38] H. Dai, X. Zeng, Z. Yu, and T. Wang, "A scheduling algorithm for autonomous driving tasks on mobile edge computing servers," Journal of Systems Architecture, vol. 94, pp. 14-23, 2019.

[39] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan, and A. Taherkordi, "Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks," IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 2, pp. 2169-2182, 2022.

[40] M. Wu, F. R. Yu, and P. X. Liu, "Intelligence networking for autonomous driving in beyond 5G networks with multi-access edge computing," IEEE Transactions on Vehicular Technology, vol. 71, no. 6, pp. 5853-5866, 2022.

[41] W. Xiong et al., "A self-adaptive approach to service deployment under mobile edge computing for autonomous driving," Engineering Applications of Artificial Intelligence, vol. 81, pp. 397-407, 2019.

[42] J. Jang, K. Tulkinbekov, and D.-H. Kim, "Task offloading of deep learning services for autonomous driving in mobile edge computing," Electronics, vol. 12, no. 15, p. 3223, 2023.

[43] M. Li, J. Gao, L. Zhao, and X. Shen, "Adaptive computing scheduling for edge-assisted autonomous driving," IEEE Transactions on Vehicular Technology, vol. 70, no. 6, pp. 5318-5331, 2021.

[44] S. Du, T. Huang, J. Hou, S. Song, and Y. Song, "FPGA based acceleration of game theory algorithm in edge computing for autonomous driving," Journal of Systems Architecture, vol. 93, pp. 33-39, 2019.

[45] C. Xia, X. Jin, L. Kong, C. Xu, and P. Zeng, "Lane scheduling around crossroads for edge computing based autonomous driving," Journal of Systems Architecture, vol. 95, pp. 1-8, 2019.

[46] A. Ndikumana, N. H. Tran, K. T. Kim, and C. S. Hong, "Deep learning based caching for self-driving cars in multi-access edge computing," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 5, pp. 2862-2877, 2020.

[47] J. Tang, R. Yu, S. Liu, and J.-L. Gaudiot, "A container based edge offloading framework for autonomous driving," ieee Access, vol. 8, pp. 33713-33726, 2020.

[48] W. Anupong et al., "Deep learning algorithms were used to generate photovoltaic renewable energy in saline water analysis via an oxidation process," Water Reuse, vol. 13, no. 1, pp. 68-81, 2023.

[49] S. P. Rajput et al., "Using machine learning architecture to optimize and model the treatment process for saline water level analysis," Water Reuse, vol. 13, no. 1, pp. 51-67, 2023.