Mobility Aware Vehicular Cloud Systems based on Edge Computing

Shelena Soosay Nathan[1], Celestina Chinenye Ezekwudo[2]
[1]Universiti Tun Hussein Onn Malaysia, Johor, Malaysia.
[2]Faculty of Education, Veritas University, Abuja, Nigeria.

Article Info

Article history:

Received Apr 9, 2025
Revised May 12, 2025
Accepted Jun 4, 2025

Keywords:

Edge Computing
Vehicle – cloud-based Systems
Autonomous Drive
Remote Sensing
Wireless networks

ABSTRACT

Remote sensing from a single integrated system in autonomous vehicles often leads to false alerts and deadlock conditions, posing major challenges to safety and functionality. To address this, cloud-based vehicle control systems have been explored, as they can synchronize data from multiple vehicles using distributed sensors. However, cloud systems face inherent limitations such as long-haul connectivity issues, increased latency, and packet loss due to interference. As a result, Mobile Edge Computing (MEC) has emerged as a promising solution in next-generation wireless networks, particularly in 5G environments. This paper proposes an edge computing (EC) based method for vehicle charging, aligned with a broad, data-driven development model. In this framework, mobility-aware edge servers interact with nearby vehicles, providing real-time information on the availability of charging stations (CSs), collecting dynamic data from moving vehicles, and implementing decentralized big data processing. This method reduces dependency on the cloud, enabling faster response times and more efficient data handling. If the access schedule between edge servers and the cloud is optimized, with cloud reliance reduced by 50%, the system maintains the same reliability while benefiting from continuous edge-based monitoring. This innovative model improves scalability, enhances energy efficiency, and supports the seamless operation of future intelligent transportation ecosystems.

Corresponding Author:

Shelena Soosay Nathan,
Universiti Tun Hussein Onn Malaysia, Johor, Malaysia.

1. INTRODUCTION

Mobile edge computing (MEC) [1], as only an Internet of Vehicles (IoV) enhancing technologies, offers possible approaches to share computing capability between vehicles, in response to other open services. Different reports are currently based on the development of autonomous vehicles [2]. They design their commuting route efficiently and monitor itself as per the sensors, like camera, GPS, etc. While these self-contained monitoring systems are satisfied, they have challenges as well. It also suggested a cloud-based automated driving system [1] to identify these difficulties, which regularly analyze sensors, data, including locations, speeds, etc. from several vehicular traffic and regulate individuals from the cloud. It would not only eliminate deadlock, thus it protects cars from expensive sensors. The cloud-based solution, as such, provides several Internet-induced issues. A few of the key concerns for reliable regulation are just to adjust for the latency. In comparison, the Internet unexpectedly has great bandwidth due to network interference which interacts with the vehicle's [3] effective remote control.

Mobile-Edge Computing (MEC), as being among the mobile service developments, has drawn significant attention [5]. MEC implements computing nodes at the edge of networks, related to as edge servers. This is responded to as an E-UTRAN module [4], if the edge server is situated at the edge of the cell network, i.e. at the access point. It refers to the Packet Data Network access point if it is at the network edge. The edge node will transmit information from low-latency automobiles.

While it is desirable from a networking viewpoint to use the edge server, the processing capabilities of the edge servers are required to be reduced. This article, suggests an Edge Computing based vehicle monitoring techniques that, along with the cloud-based automobile monitoring system, requires the edge server as compute nodes, as shown in Figure 1. They have automobile controls on the cloud and edge servers and are related to the cloud processor and the edge control system. The edge controller is capable of achieving secure vehicle power against changes in the network.

The proposed framework does have an automated vehicle interface switching device between edge and cloud providers to manage the computing load between edges and private cloud, exchanging behavioral characteristics between edge and cloud servers to avoid unreliable power by devices. A sensor detects bandwidth between edge and cloud services and transfers control around them as follows because variance is also the most system-level parameter to decrease in the prior analysis and social. The authorized user gets various sensors from automobiles to the private network if the bandwidth is below the threshold range, and the vehicle system is transferred to the cloud server; then, the edge server ends routing the sensor data and uses on these various sensors.

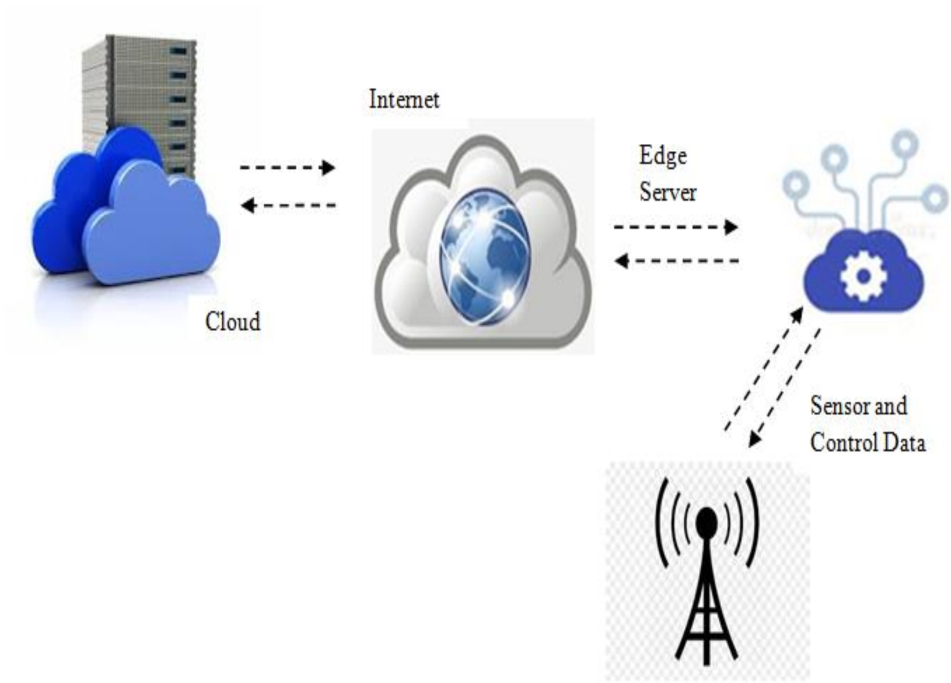

Figure 1. Edge Computing based Vehicular System

Figure 1 explains the basic structure of the edge computing. Our proposed work, to the results related, is the one to suggest synchronized vehicle regulation for both the cloud and the MEC. While MEC solutions for vehicles are being suggested in [5], they were reduced to recreational activities and hazard alerts and were not thoroughly investigated in terms of vehicle regulation. While cloud-based automated driving systems [6] are being suggested, they presume valuable controllers and use the cloud for edge devices [7]. This article uses inter-cars to execute the model and test the reliability of traffic management. It determines the monitoring, proportion, and volume of traffic to match the computing load between edge and cloud servers. The key objectives are greater than the amount of this issue. Initially, as regulated from the cloud with a duration of 150ms, multi-cars differ from the path by more than 0.1 m from the 50 percent of the total path that may trigger traffic congestion.

However, a security flaw in VFCN distributed systems poses a significant network difficulty. Users' decisions on the reliability, security, integrity, and authenticity of services offered by different service providers are reflected in VFCN's trust-aware resources. Authenticity pertains to the veracity of identification assertions made by fog and cloud node-based services.

As a result, it is difficult to guarantee safe multi-side offloading and scheduling for e-transport applications in the VFCN. A promising option is blockchain-enabled technology, which is made up of an expanding collection of data points called blocks. Through the use of cryptographic hash values, these blocks connect. Every block contains the period, transaction data, and a cryptographic fingerprint of the one before it. To solve the security and trust concerns, VFCN can make use of these fundamental blockchain features.

Moreover, our proposed device transfers power between the edge and the cloud servers dynamically as per the nodes in the network while reducing the reliability of the moving path. So if the amount of virtual machine monitoring phase to cloud improving the efficiency is reduced to 50 percent, it will enhance the same reliability as edge controller constant monitoring. Finally, the trade-off between the high reactivity to the transition in network architecture and the volume of traffic between edge and cloud services is established. This article is stated as follows. In section 2 discusses the related works of this paper. The framework suggested and the architecture is discussed in Section 3. The proposed system is evaluated in section 4. Section 5 concludes this paper.

2. RELATED WORKS

In recent years, owing to its continuous support for computation-intensive systems and interruption-sensitive applications, MEC also gained considerable publicity. The idea of edge computing and a few research studies were presented in [8], varying from cloud distributing to smart device and region, and also an interactive edge to emerge the edge computing theory. In the area of edge computing, they have also identified many issues and solutions. The major application instances and comparison structures for the relevant MEC have been defined in section 3. The researchers have described the new principles of MEC technology integration into mobile networks and examined the recent advances in the standardization of MEC. In [9], the authors presented a detailed review of recent MEC studies, in particular, the participants will complete communication and computing resources. The researchers have described concerns pertaining to MEC analysis and possible research paths, such as the implementation of the MEC method, optimization, control of mobility, utilization of natural resources, and confidentiality. In [10], the study evaluated a large-scale MEC network's bandwidth efficiency. The MEC framework is developed using mathematics with differentiated features of wireless connectivity and computation to research the costs and benefits between these parameters and restrictions. The latency is analyzed depending on the condition by the implementation of stochastic geometry theory, queuing, and concurrent computation. ETSI considers the MEC platform as an integral aspect of centralized connectivity and big data. The aim is to build support for the implementation structure that allows developers, service providers, and third-party edge processing high concentration to be integrated [11].

The implementation phase is a vital MEC technique and has a critical effect on the performance of the operation and the customer interface. In [12], in a cloudlet-based mobile cloud environment, the developers discussed the effect of edge computing of loading decision making across numerous smart applications. The article [13] proposed an optimized architecture for device transferring and disturbance control in heterogeneous wireless edge computing mobile communication systems. But instead, in this structure, the computation offloading interpretation, existing resource network distribution, and MEC computing resource provisioning concerns are proposed. In order to accurately handle computation offloading and web configuring techniques with edge computing in wireless communications systems, the article [14] developed the determination on sensor networks, resource distribution, and data routing technique as an optimization method, considering the overall system profitability.

Offloading systems are developed to enhance energy costs, delay efficiency, or expense for knowledge used in several current works [15]-[18]. In [15], the article analyzes the distribution of resources for a multi-user MECO scheme focused on cognitive radio period-division, multicarrier modulation dynamic spectrum-division, allowing for all instances of indefinite and the finite computing power of the cloud. It illustrates that the optimum resource distribution strategy must have a threshold-based mechanism to minimize the weighted amount of mobile energy usage. In [16], the authors approached the issue of computing in a mixed fog/cloud environment by properly selecting the choices of loading and allocating computing energy, transmission power, and radio bandwidth, as maintaining user parity and maximum tolerable latency. In [17], separate from the foregoing studies, the authors explored the issue of minimizing joint energy use, latency, and payment expense for the MDs in an interconnected fog computing network.

In [18], the authors explored the topic of shared allocation of radio and computational resources to maximize device efficiency and boost customer satisfaction. In view of the distributional characteristics of the IoT system, a balancing methodology is explored to achieve a secure fit between network devices as a semi-distributive solution method. In recent years, integrate MEC and vehicular systems have also been a testing hotspot. A MEC-based multi-path communication resource equalization challenge to overcome was suggested by Research [19]. The approach is implemented by the integration of edge devices and cloud computing data centers to the edge computing system. By evaluating the response behavior of various kinds of vehicles, different kinds of goals are to achieve are automatically distributed to every edge node, and then the fog nodes of every edge node are efficiently distributed to the related VM to minimize the execution time of the vehicle model.

In [18], the study suggests an issue for sensor communications where MEC and cloud computing combines for shared computing offloading. By integrating the estimation of offload decisions and calculating resource distribution to optimize device availability, this function can be considered as restricted optimal. In addition, to handle the issue, a CCORAO system is recommended, which involves machine discharge multiplayer and distribution of resources. To minimize device complexity without losing efficiency, the DCORA method was introduced in the solution. In [19], by finding the optimal connectivity and processing capabilities in the MEC allowed vehicle network, the study identifies the computational overload reduction issue. The researchers initially transformed the non-convex issue into an analogous issue. Then the researchers separated the issue of equivalence into different parts. Moreover, to achieve the optimal approach, this article indicates a low-complexity algorithm. Numerical findings indicate that only a lot of processing downtime can be saved by this approach.

3. PROPOSED WORKS

This section, also suggests an infrastructure-based vehicle system model that has an automated vehicle controller transferring approach between edge and cloud nodes that would save the multitude of functions use of edge devices and prevent performance issues. Figure 2 presents the architecture for the design proposed. The key to providing includes controls for edges and clouds. Both cloud and edge controls have vehicle interfaces with the same features, and the vehicles are controlled by each of these. The edge controller controls and transfers power between edge and cloud servers and monitors network activity.

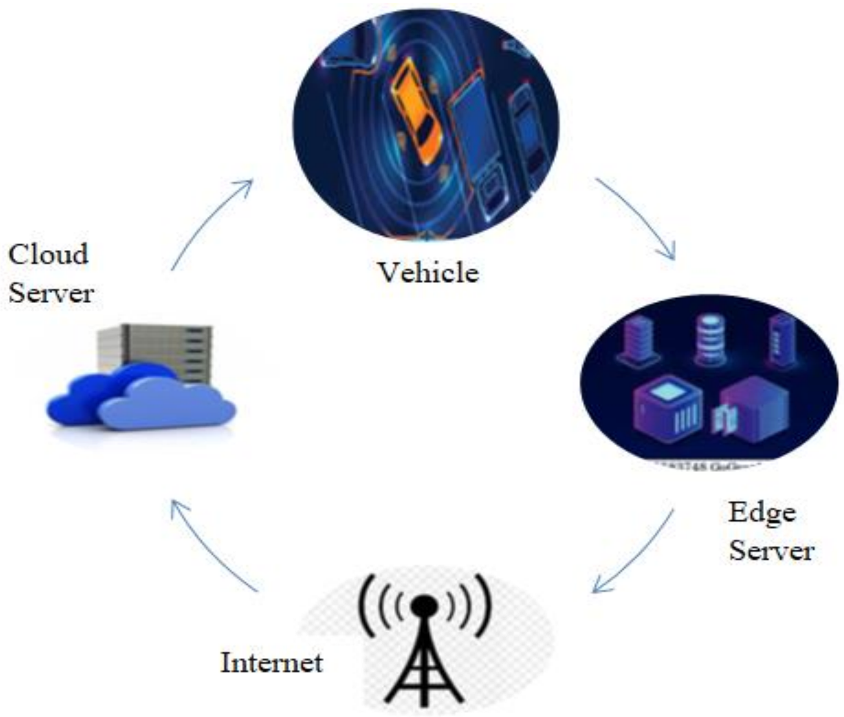

Figure 2. Vehicular Edge Computing

Figure 2 depicts the architecture of vehicle edge computing. Autonomous cars transmit vehicle trajectory network packets continually with vehicle ID, location, velocity, and angle of attack obtain Vehicle Control Data packets with Vehicle ID, vehicle speed, accelerating quality, and actuator importance, and fix the grip, engine and trigger in compliance with the Vehicle transmissions. If the vehicles are operated by the cloud processor, the edge controller converts Vehicle transmissions from the vehicle to the cloud and Vehicle parameters of the vehicle from the cloud. Vehicle signals are received by the cloud controller and Vehicle packets are sent. A broad variety of Vehicle transactions could be aggregated by the cloud controller and a number of remotely controlled device computing tools can be used. Moreover, it raises numerous network issues induced by the Internet, like delay, node mobility, network capacity, etc.

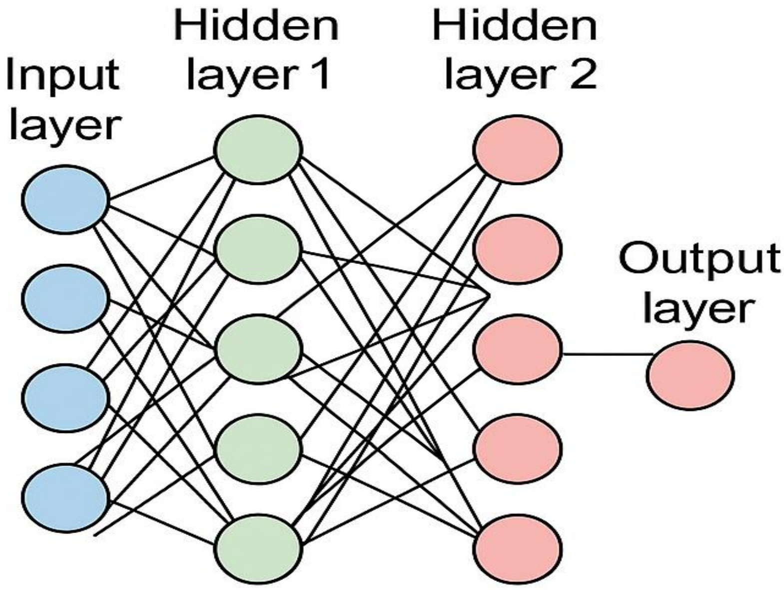

Figure 3. Neural network architecture diagram

Figure 3 shows a basic feedforward neural network with:

- **Input Layer (blue):** Takes in the initial data (e.g., sensor readings from vehicles).
- **Hidden Layers 1 & 2 (green and red):** Process the data using weighted connections and activation functions.
- **Output Layer (red):** Produces the final decision/output (e.g., control signals to autonomous vehicles).

The next node decrypts the data, recalculates the hash on the data, and the one that came before determines Ik whenever an authorization task is scheduled to another node. If the two hash values are the same, this ensures the safety of the method of migration and certifies the accuracy of the data. Additionally, the existing server recognizes Il and encrypts the data when the new node estimates the hash on the data. The secure hash of the preceding block is also inserted into the current permanent block by the to node. As demonstrated by Algorithm 1, in the event of a security breach, the digital ledger can assist in identifying the affected VFCN with scenarios.

- Decrypt the task information.
- Recalculate the hash using the data from the decrypted task. I1 is recognized by Si and the preceding node.
- Verify the accuracy of the data and the safety of the migration process by comparing it to the hash value contained in the task block. If the two values are the same.
- Additionally, node 2 estimates the digital signature on data once more, using the most recent server catalogs I2 this time.
- The node adds the hash from the prior block to the currently unchanging block in order to create a blockchain.

Algorithm 1: Task Offloading for Mobility Using Blockchain Input: {a } A,{a,va,l x a,k } K,i } N}, BC[]}

One start,
two foreach (i ~ a } A as N &k ~ K).
Perform 3 if (l0 a { Dg/2 & {a == true),
followed by 4 {a,i =1;
5 Use SHA256 encryption for tasks with security annotations.
N i=1 SCi SHA256 adheres to all procedures as detailed in the bullets;
Hash(Si, Ia) 6 BC[{a, Si] };
7. Determine the system expenses using equation (19);

9 Optimize P1;
10 if (1 − `{ a)
then 11 compute system costs using equation (20);
8 BC[{a, Si] } hash(Si, Ik);
13 BC[{a, Si] } hash(Si, Il);
12 BC[1 − `{ a, Si, k] } k;
14 Make P1 better;
15 if (k ~ K = 0),
then 16 k = 0 indicates a crashed node and k = 1 indicates a stable node?
17 Use CPBFTM, or the Consensus Practical Byzantine Fault Tolerant Method;
18 (K1, k2); 19 End-Loop;

Step 1: Transfer task data from the sender applications to the node receiver. To guarantee that the duties of data acquired from the original sender have been completed and that only the sender's information has been required, the receiver node had not been altered.
Step 2: By offloading the information provided from the programs, the sender gathers it for offline transmission. The node sender computes the data into a 256-little number using the SHA256 hash method.
Step 3: After that, the sender node compresses the 256-bit number into a digital signature by signing a private key with it. The contents of the responsibilities, the digitally signed document, and the public key node are now sent from the sender to the recipient node (keep in mind that you cannot use the public key to determine the associated private key, therefore sharing is safe). Any computing node with the private key must publish the data to the public shared key, and the receiving node must verify that the data delivered to the tasks has not been changed.
Step 1: Using the acquired public key, the receiver's node unravels the 256-bit digital authentication number. Step 3 above the sender's is "reversed" when the public key is applied to the digital signed document.
Step 2: The reception device then compares the SHA256 encode to the received data to produce a 256-bit integer; this procedure is identical to that of the transmitter's steps 1 and 2.
Step 3: The recipient then verifies that the two 256-bit numbers are the same. If incorrect, the data has been altered or a public key that does not match the sender's private key has been provided. The receiver should be notified that this information is available to use if it is authentic.

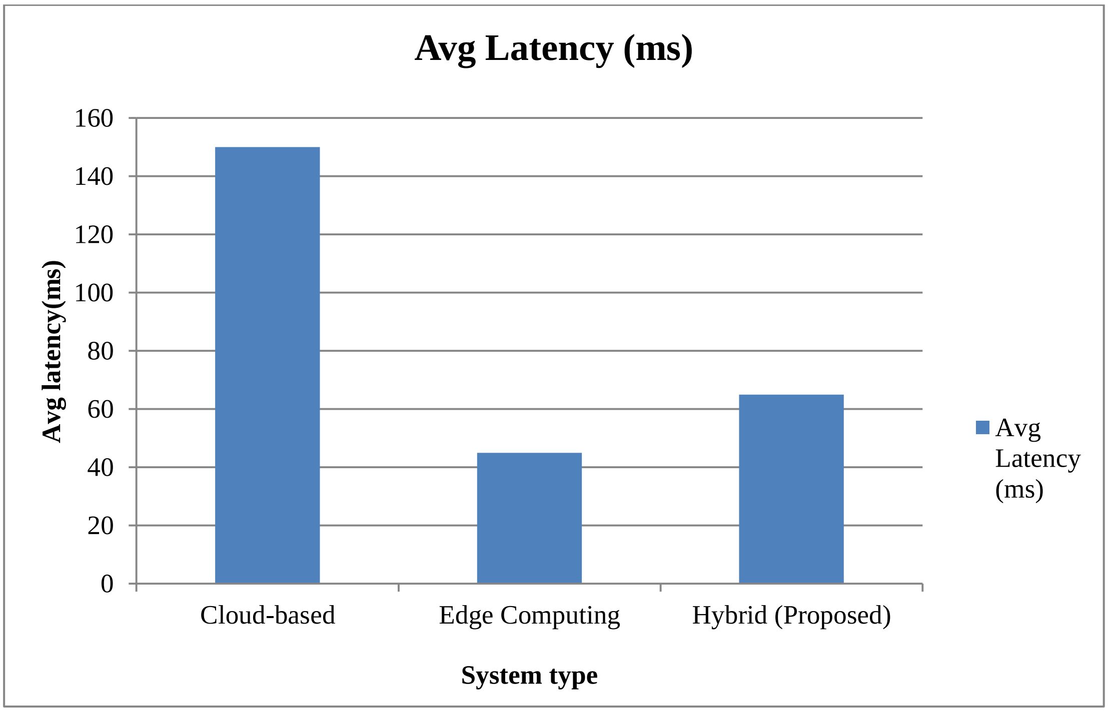

Figure 4. Latency Comparison Between Cloud and Edge Systems

Figure 4 replects the comparison of cloud and edge systems. If the vehicle is operated by the edge control system, the edge controller accepts VSD data and gives away cloud-free VCD files. It presumes that the edge server is installed on the access network and that vehicle control with minimal computing resources has to be processed by the edge server. On another side, without network problems created by the Internet, the edge server will perform vehicle power. A suggested device proposes a mobile 5G network. The connected network is one of a broadband network linking the Internet to vehicles. In 5G mobile growth, there is the conflict that it should implement edge servers. As per the research papers of 5G cellular networks, the radio component of end-to-end bandwidth is estimated at below milliseconds. Through implementing MEC servers on the vehicle networks, VEC's system capacity was being greatly increased. As such, VEC includes dynamic and hierarchical RSUs, vehicles, and MEC servers. Hence it is necessary and difficult to design VEC 's structure. It illustrates certain architectural concepts under is important to maintain VEC's sufficient and reliable service.

- Scalability refers to the software's ability to function, to improvements in vehicles and edge computing applications. On the first side, a significant number of vehicles are also required for entry or departure in VEC, so the device must be able to establish easily and satisfactorily. On the other side, VEC must also be capable of promoting program installation and software updates. Scalability for all applications is a significant aspect.
- A significant aspect of these structures is cognitive enhancers. Low energy consumption and low latency are included in efficiency enhancements for the VEC. The MEC server's electricity usage in the automobile consists of the usage of electricity, like fuel. Higher usage of electricity results in decreased energy consumption. The VEC should, thus, give greater attention to minimizing the consumption of electricity. In addition, the delay has a significant effect on the experience of the consumer and therefore should be recognized.
- Implementation of mobility is essential to service efficiency in the network. The vehicles are going quickly in VEC, and the communication network is changing significantly. The relation between the RSU and the vehicles is, thus, not secure. Offloading the vehicle is vulnerable to computing. Systematic protection is also required for the mobility of automobiles.

Users model the VEC system on the basis of project specifications, as shown in Figure 1, that includes a number of servers, vehicles, and RSUs from MEC. MEC servers provide resources for cloud computing systems. They build frameworks for edge computing that might access data for edge computing and perform computing activities. At VEC, there are different sorts of edge computing servers: Vehicle MEC servers and Relay nodes MEC servers. Other than the earlier, certain software, including model software or navigation programs, could be built on such servers. Because of the vehicle's reduced computational capacity, certain computing activities created by programs on the vehicles' supporting MEC servers and RSUs for analysis will have to be added. The vehicle which involves computational activities to those other sites for processing is called the vehicle introduced by the initiative. The RSUs provide a wide variety of facilities for the others, but the MEC servers have enough energy. They should include vehicles in the region with edge computing facilities, and also perform the computing activities that the vehicle's discharge. RSUs can assign the computational activities to the data center in the network infrastructure for execution while all the MEC processors in the scope are active.

The vehicles will interact with someone via the wireless communication in VEC to build a cluster and co-operate on mathematical computations. Both vehicles and clusters could be controlled beyond scope by the SDN-based centralized communication axis in the RSUs. Most precisely, the resource condition details of the MEC servers on vehicles could be queried and the servers managed via the data plane application. The communication research paper focuses on the situation of the vehicles is specified by the unified communication plan. In order to share knowledge and computational activities in a timely fashion, the operating collective vehicles will then communicate with each other. VEC also improves job execution by collective data processing and decreases the processing period of the task-initiated engine.

Features of every operating system platform of the others built framework can be seen, as said in the suggested VEC framework, in Figure 2. As described below, the VEC framework depends on three functionalities, namely the operation and device plane, the VEC network, and the network infrastructure plane. The programming activities of the mission-initiated automobile could be performed quickly in that framework, and separate programs and the functional areas of resources and networks may easily grow.

4. VEHICULAR EDGE COMPUTING

The edge cloud computing environment guarantees the regular and efficient system of job management, resource utilization, and planning, such as the RSU network virtualization system and vehicle edge computing system [20]. The edge computing paradigm wants to integrate related system components as an organizational system based to fulfill the platform's system feature specifications, like scheduling algorithms, channel control, routing and propagation, coordination, accessibility management, and data caching. The resources management system helps the edge computing services to call the managed network services to fulfill the specifications of the computing task; Channel surveillance may obtain transmitted signal at any moment, aimed at facilitating conflicting data between operation-initiated vehicles, collective vehicles, and RSUs due to its higher-speed travel of the project-initiated vehicle; Functional scheduling, work allocation, mobility assistance, and other tasks among the vehicles are introduced by the networking and routing, decentralized communication and data transfer subsystem; data storage will efficiently guarantee edge computing system data protection. In order to accomplish the appropriate functions, users might choose to insert the related specialized cellular if needed. Table 1 shows clearly the Paramters that can be used in experimental section.

Table 1. Vehicle Computing Parameters

Parameter	**Value**
CPU Speed	1.2 GHz
Energy Consumption (per g-cycle)	600 mJ
Task Input Volume	0.3 – 0.5 GB
Transmission Bandwidth	20 MHz
Decision Time	1 minute

5. EVALUATION RESULTS

In this article, it performs several experimental analyses to show the value of our suggested VEC framework and the offloading structure for collaborative activities. The general criteria for the simulator are described as follows. In a multinational group, the amount of vehicles is 2. And the stack status and channel state numbers were set to 2 and 1, respectively. The queue state integer is, whereas, set to 5. The minimum quantity of Computational resources per task is chosen systematically for the computation sequence diagram in the scope g-cycles, whereas the input feature, the volume is primary particles between [0:3; 0:5] GB. In addition, it believes that the vehicle's computing capacities are 1.2 GHz, and hence the absorbed energy is set to 600 mJ/g-cycle for the CPU cycle. In addition, the frequency, the transmitting capacity between the vehicles, and the acoustic intensity are calculated collectively to 20 MHz. The decision-making time is fixed at 1 minute.

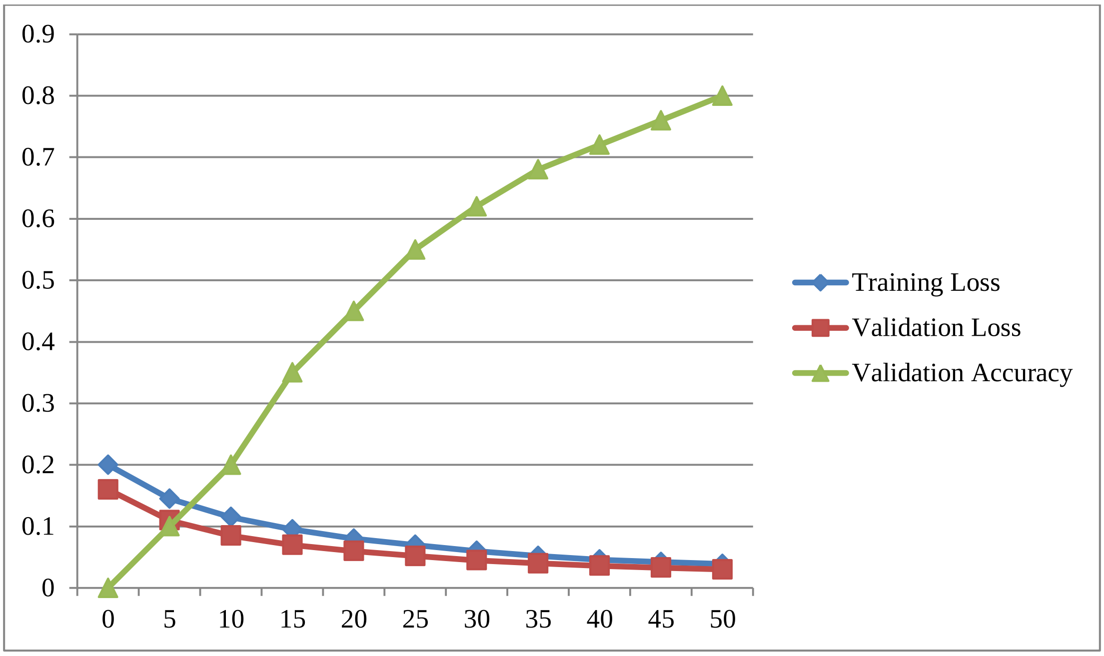

Figure 5. Neural Network Training Metrics over Epochs

Figure 5 explains the neural network training metrics over the epochs that could perform well.

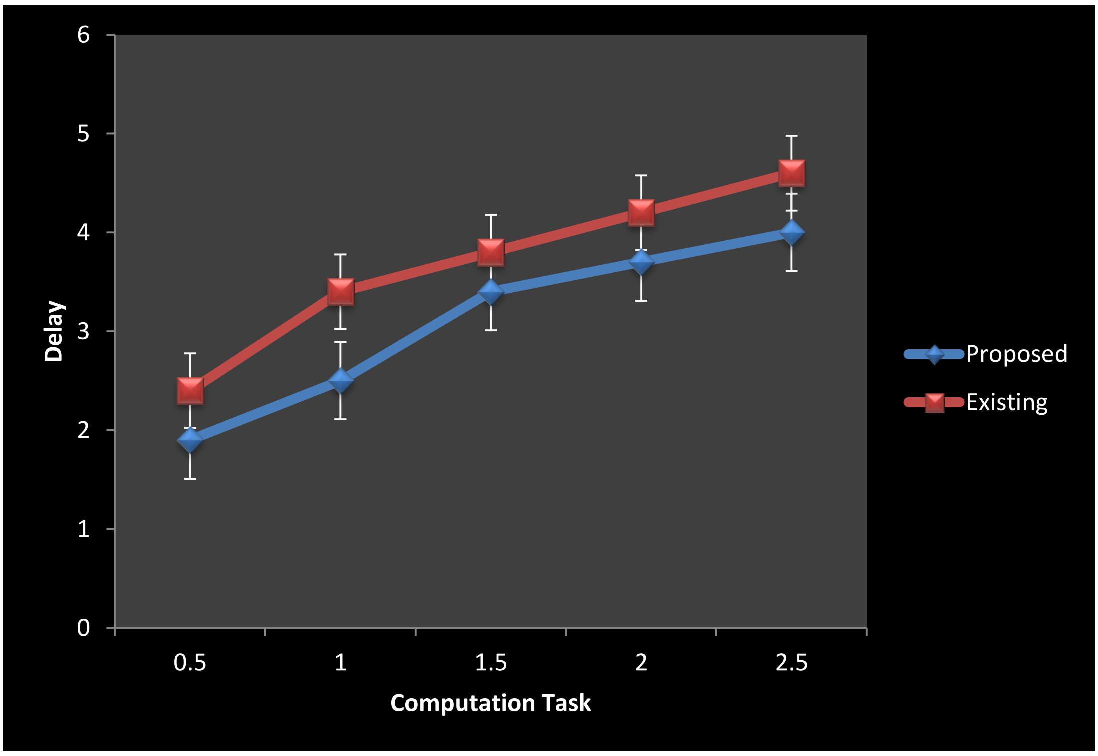

Figure 6. Performance Analysis of the average Computation Task.

The early R variable and the traditional inverters are defined in 0, simultaneously, for the R learning-based technique of loading. The training frequency is adjusted at 0:7 and the aggressive algorithm's data point is 0:2. In addition, after 250 learning instances, the output, outcomes of the implemented R learning-based methodology of initialization are obtained, including one that includes 2500 phases, i.e. 2500 decision cycles.

It equates our method to two sub-systems, i.e., a static integrative framework and a localized non - cooperative system, to help validate the effectiveness of our proposed model. In separate simulation settings, where the mean processes per/mission and the task arrival time have differed, it evaluates the efficiency of several offloading systems.

Figure 6, in various average calculations per task, it validates the proposed offloading system. These will note that as the average calculations per task raise, the waiting time of the systems proposed improves. This is due to the belief that every task's expanded total simulations per task would lead to a high computational time. In comparison, the collaborative framework latency is relatively lower than that of the non-collaborative system, and their suggested collaboration framework results higher than the coordinated dynamic system.

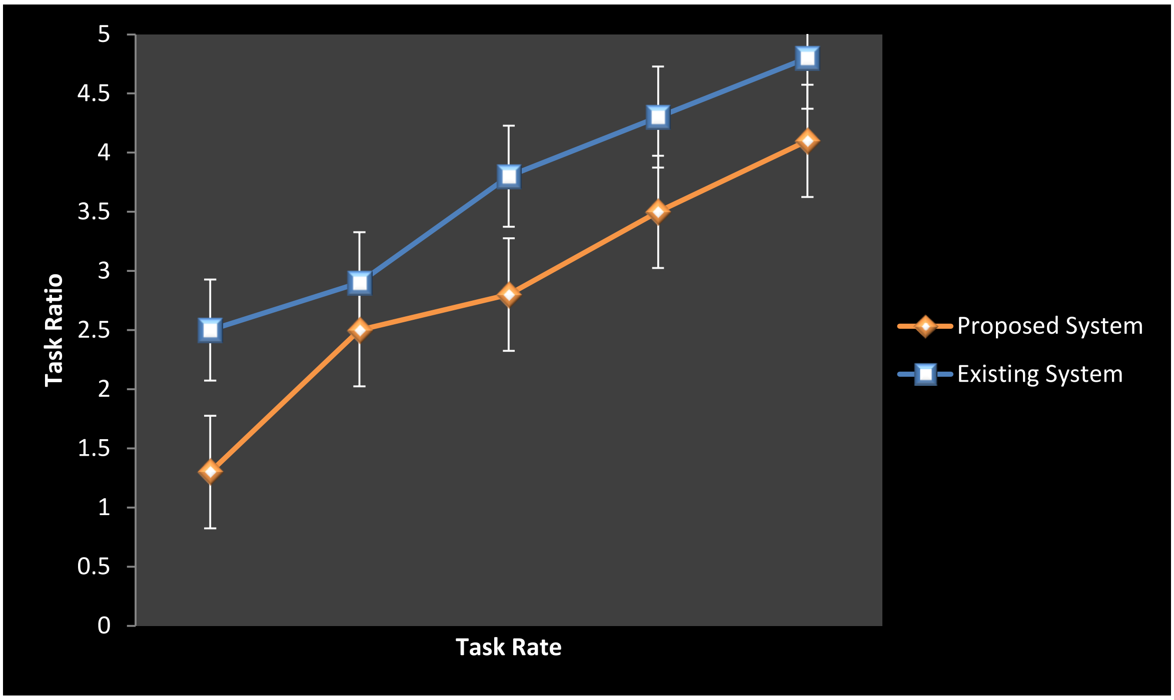

Figure 7. Evaluation of Task Ratio and Rate

Figure 7 gives the relation between the failure ratio of tasks and the packet size of tasks. The work failure ratio would rise if many new tasks appear due to the absence of edge computing network node queue space. Due to higher response in order rate of speed, further forwarded activities would occur, which might be eliminated from the queue of the edge compute node. In general, a minimal task failure ratio could be obtained by the proposed method as it might assess the proper decisions for global applications as well as offloading.

Table 2. Edge vs Cloud Offloading Outcomes

Metric	**Cloud**	**Edge**	**Proposed Hybrid**
Task Success Rate (%)	85	93	**97**
Energy Use (kJ)	12.5	10.2	**9.8**
Avg Delay (ms)	150	45	**65**

Table 2 Summarizes key benefits of the hybrid model in efficiency, speed, and reliability.

6. CONCLUSION

In this paper, it presented the VEC collaboration framework that would both boost the vehicle network's mission processing power and resource usage. It initially addressed the architectural concepts of VEC, which is the basis of device service. Then, it presented VEC's particular framework and functional outcome. After this implemented the VEC communication method in-depth, and the Edge computing technology-based implementation proposal was issued. The challenges were then defined and described, primarily involving the creation of edge coalitions, collective schemes, and control of mobility. It may also solve several other research questions required for the effective functioning of the system. Numerical findings have indicated that our proposed framework will produce better efficiency in a processing time delay and task failure, proportion, which facilitates computational task coordination between vehicles.

REFERENCES

[1] Feng, J., Liu, Z., Wu, C., & Ji, Y. (2018). Mobile edge computing for the Internet of vehicles: Offloading framework and job scheduling. IEEE vehicular technology magazine, 14(1), 28-36.

[2] Sasaki, K., Suzuki, N., Makido, S., & Nakao, A. (2016, September). The vehicle control system is coordinated between cloud and mobile edge computing. In 2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE) (pp. 1122-1127). IEEE.

[3] Yuya Suzuki, Kengo Sasaki, Kenya Sato, & Hiroaki Takada. (2015). Low-delay processing method for stream processing LDM oriented toward cloud-based autonomous driving. Embedded System Symposium 2015 Proceedings, 2015, 84-92.

[4] Patel, M., Naughton, B., Chan, C., Sprecher, N., Abeta, S., & Neal, A. (2014). Mobile-edge computing introductory technical white paper. White paper, mobile-edge computing (MEC) industry initiative, 1089-7801.

[5] Hu, Y. C., Patel, M., Sabella, D., Sprecher, N., & Young, V. (2015). Mobile edge computing—A key technology towards 5G. ETSI white paper, 11(11), 1-16.

[6] Kumar, S., Shi, L., Ahmed, N., Gil, S., Katabi, D., & Rus, D. (2012). Carspeak: a content-centric network for autonomous driving. ACM SIGCOMM Computer Communication Review, 42 ss(4), 259-270.

[7] Gerla, M., Lee, E. K., Pau, G., & Lee, U. (2014, March). Internet of vehicles: From an intelligent grid to autonomous cars and vehicular clouds. In the 2014 IEEE world forum on the internet of things (WF-IoT) (pp. 241-246). IEEE.

[8] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of things journal, 3(5), 637-646.

[9] Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials, 19(4), 2322-2358.

[10] Ko, S. W., Han, K., & Huang, K. (2018). Wireless networks for mobile edge computing: Spatial modeling and latency analysis. IEEE Transactions on Wireless Communications, 17(8), 5225-5240.

[11] Hu, Y. C., Patel, M., Sabella, D., Sprecher, N., & Young, V. (2015). Mobile edge computing—A key technology towards 5G. ETSI white paper, 11(11), 1-16.

[12] Cao, H., & Cai, J. (2017). Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: A game-theoretic machine learning approach. IEEE Transactions on Vehicular Technology, 67(1), 752-764.

[13] Wang, C., Yu, F. R., Liang, C., Chen, Q., & Tang, L. (2017). Joint computation offloading and interference management in wireless cellular networks with mobile edge computing. IEEE Transactions on Vehicular Technology, 66(8), 7432-7445.

[14] Wang, C., Liang, C., Yu, F. R., Chen, Q., & Tang, L. (2017). Computation offloading and resource allocation in wireless cellular networks with mobile edge computing. IEEE Transactions on Wireless Communications, 16(8), 4924-4938.

[15] You, C., Huang, K., Chae, H., & Kim, B. H. (2016). Energy-efficient resource allocation for mobile-edge computation offloading. IEEE Transactions on Wireless Communications, 16(3), 1397-1411.

[16] Du, J., Zhao, L., Feng, J., & Chu, X. (2018). Computation offloading and resource allocation in mixed fog/cloud computing systems with a min-max fairness guarantee. IEEE Transactions on Communications, 66(4), 1594-1608.

[17] Liu, Yuan, Ke Xiong, Yu Zhang, Li Zhou, Fuhong Lin, and Tong Liu. "Multi-objective optimization of fog computing assisted wireless powered networks: Joint energy and time minimization." Electronics 8, no. 2 (2018): 137.

[18] Gu, Y., Chang, Z., Pan, M., Song, L., & Han, Z. (2018). Joint radio and computational resource allocation in IoT fog computing. IEEE Transactions on Vehicular Technology, 67(8), 7475-7484.

[19] Alsheikh, M. A., Hoang, D. T., Niyato, D., Tan, H. P., & Lin, S. (2015). Markov decision processes with applications in wireless sensor networks: A survey. IEEE Communications Surveys & Tutorials, 17(3), 1239-1267.

[20] Xie, R., Tang, Q., Wang, Q., Liu, X., Yu, F. R., & Huang, T. (2019). Collaborative vehicular edge computing networks: Architecture design and research challenges. IEEE Access, 7, 178942-178952.

[21] Li, Y., Yang, C., Chen, X., & Liu, Y. (2024). Mobility and dependency-aware task offloading for intelligent assisted driving in vehicular edge computing networks. Vehicular Communications, 45, 100720.

[22] Zhou, J., Yang, Q., Zhao, L., Dai, H., & Xiao, F. (2024). Mobility-aware computation offloading in satellite edge computing networks. IEEE Transactions on Mobile Computing, 23(10), 9135-9149.

[23] Li, J., Guo, S., Liang, W., Wang, J., Chen, Q., Xu, W., ... & Jia, X. (2024). Mobility-aware utility maximization in digital twin-enabled serverless edge computing. IEEE Transactions on Computers.

[24] Abbasi, M. H. A., Arshed, J. U., Ahmad, I., Afzal, M., Ali, H., & Hussain, G. (2024). A Mobility Prediction Based Adaptive Task Migration in Mobile Edge Computing. VFAST Transactions on Software Engineering, 12(2), 46-55.

[25] Chen, L., Du, J., & Zhu, X. (2024). Mobility-Aware Task Offloading and Resource Allocation in UAV-Assisted Vehicular Edge Computing Networks. Drones (2504-446X), 8(11).

[26] Nahar, A., Mondal, K. K., Das, D., & Buyya, R. (2024). Clouds on the road: A software-defined fog computing framework for intelligent resource management in vehicular ad-hoc networks. IEEE Transactions on Mobile Computing.

[27] Sang, Y., Wei, J., Zhang, Z., & Wang, B. (2024). A mobility-aware task scheduling by hybrid pso and ga for mobile edge computing. Cluster Computing, 27(6), 7439-7454.