Eng. Sci., 2023, **26**, 980

Engineered Science

DOI: https://dx.doi.org/10.30919/es980

An Innovative Resource Management Framework using Logit-Boosted Machine Learning Algorithms for Vehicular Ad Hoc Networks (VANETs)

Wenye Zhang, Sergey Evgenievich Barykin,* Tatiana Viktorovna Kirillova, Irina Vasilievna Kapustina, Nikita Sergeevich Lukashevich and Andrey Zaytsev

Abstract

The article presents the results of a study of the physicochemical properties, mineral and elemental composition of the natural mud of Moiyldy, Tuzkala, Arasan, Shoshkaly, the eastern coast of Lake Alakol salt lakes of the North-Eastern region of Kazakhstan. Research on the composition of mud is one of the current trends carried out by scientists from different countries to identify compounds that cause their positive therapeutic effects. A review of the available data on the physical and chemical composition of natural muds in Kazakhstan showed the lack of systematic studies of the composition and structure of peloids in the North-Eastern part of the country, which are popular for their healing properties. The surface morphology, mineral composition, and physico-chemical parameters of natural mud were studied for the first time. The interpretation of the obtained data was carried out by taking into account previous studies of peloids from other regions of Kazakhstan, and a physico-chemical assessment of its therapeutic effect was given. According to the results obtained, the studied peloids are mostly salt-saturated muds with a pH range of 8.66 - 9.69. Peloids are characterized by a high content of sulfate ions and ammonium ions, which together have anti-inflammatory and wound-healing effects. The mineral composition is mainly represented by fine-grained quartz and albite. All samples are characterized by a high content of bismuth compared to its clarke in the Earth's Crust, which indicates the need for monitoring control of the composition and studying the possible effects of using these native muds.

Keywords: Vehicular ad-hoc networks; Machine learning; Logit-Booster; Resource predictor; Performance evaluator.

Received: 12 August 2023; Revised: 28 September 2023; Accepted: 03 October 2023.

Article type: Research article.

1. Introduction

Vehicular Ad Hoc Networks (VANETs) are a type of wireless ad hoc network that enable communication among vehicles and infrastructure nodes (*e.g.*, traffic lights, road signs) in a dynamic and decentralized manner. VANETs have the potential to enhance road safety, improve traffic efficiency, and enable various applications in the automotive industry. However, VANETs face several challenges such as limited resources, dynamic network topology, and intermittent connectivity. Edge computing, which involves processing and analyzing data at the edge of the network, has the potential to address some of these challenges.

When discussing intelligent transportation systems, the arena of Vehicular Ad Hoc Networks (VANETs) is one that is both dynamic and transformational. VANETs are more of a network concept than a specific protocol, algorithm, or architecture for enabling communication between vehicles and other roadside equipment. The paper focuses on the central problem of effectively allocating resources within this complex network ecology. Allocating and optimizing resources like processing time, memory, network throughput, and power consumption is what resource management in VANETs is all about. These assets are most commonly associated with cars' on-board units (OBUs), but they also include roadside units (RSUs) and traffic control systems as part of the larger VANET infrastructure. Our proposed paradigm for managing available resources was purposefully designed to perform better than comparable efforts in the literature. Our framework excels in resource allocation thanks to its careful use of logit-boosted machine learning techniques, which allow for considerable reductions in network latency, improvements in resource usage, more fair distribution of

Graduate School of Service and Trade, Peter the Great St. Petersburg Polytechnic University, St. Petersburg 195251, Russia.

**Email:* sbe@list.ru (S. E. Barykin)

available resources, higher throughput, and improved application performance.[1-4]

Edge computing refers to the practice of processing and storing data at the edge of the network, closer to where the data is generated, rather than sending it to a centralized data center. This approach can lead to faster processing times, reduced latency, and improved efficiency in resource utilization. In the context of VANETs, edge computing can be used to improve the safety and efficiency of road traffic by enabling real-time processing of data from vehicles and roadside infrastructure, such as traffic signals and cameras. For example, edge computing can be used to process data from sensors on vehicles to detect and respond to potential safety hazards on the road, such as accidents or hazardous weather conditions. Our proposed resource management framework is specifically designed to optimize the allocation of resources for edge computing in VANETs, taking into account the dynamic and heterogeneous nature of the network. By leveraging machine learning algorithms to predict resource availability and allocate resources to appropriate applications, our framework aims to improve network performance and enable new applications in the automotive industry.

Resource management is a critical issue in VANETs, as these networks have limited resources such as CPU, memory, and bandwidth. In addition, the dynamic nature of VANETs makes it challenging to predict the availability of resources and allocate them to appropriate applications. Traditional resource management approaches in VANETs are either centralized or distributed, and they rely on predefined rules or heuristics. These approaches may not be suitable for VANETs, as they are highly dynamic and unpredictable.[5-7]

Vehicular Ad Hoc Networks (VANETs) have emerged as a promising technology for intelligent transportation systems. They facilitate the exchange of information among vehicles, infrastructure, and pedestrians in real-time, enabling applications such as collision avoidance, traffic monitoring, and infotainment services. Resource management in VANETs is crucial to ensure efficient and effective communication among these entities. Edge computing is a promising technology that brings computation and storage resources closer to the end-users, reducing the latency and improving the response time of applications. It has the potential to address the limitations of traditional cloud computing architectures in VANETs, where high mobility, limited connectivity, and low data rate pose significant challenges. In this context, this paper proposes a novel resource management framework that leverages machine learning algorithms to optimize the allocation of resources in VANETs using edge computing. The proposed framework aims to predict the availability of resources at the edge of the network, allocate them to the appropriate applications based on their requirements and the current state of the network, and evaluate the performance of the framework in terms of latency, utilization, fairness, and application performance. To achieve this, we first formulate a mathematical model for the resource allocation problem and then propose a solution based on machine learning algorithms. Specifically, we employ logit-boosted machine learning algorithms to predict the availability of resources at the edge of the network and allocate them to appropriate applications. We also introduce a performance evaluator to measure the effectiveness of the framework in terms of network latency, resource utilization, and application performance.

A VANET is a type of wireless ad hoc network that consists of a set of vehicles and infrastructure nodes (*e.g.*, traffic lights, road signs) that communicate with each other to exchange information about road conditions, traffic flow, and hazards on the road. In a VANET, resource management refers to the allocation of limited resources such as CPU, memory, and bandwidth to different applications running on vehicles or infrastructure nodes.

Resource management in VANETs can be modelled using the following Eq. (1):

$$R = \{r_1, r_2, \ldots, r_n\} \tag{1}$$

Here, R represents the set of available resources in the network. Within this set, r_1, r_2, ..., r_n represent individual resources, and n signifies the total number of distinct resources contained in the set.

It's essential to note that each resource, denoted as r_i, within the set R possesses a specific capacity or availability, represented as a_i. This capacity signifies the upper limit of workload that a given resource can effectively handle at any point in time. For instance, the capacity of a CPU resource may be quantified in terms of its ability to execute a certain number of instructions per second.[8-10]

Let $C = \{c_1, c_2, \ldots, c_m\}$ be the set of applications in the VANET, where $c_1, c_2, \ldots, c_m$ are individual applications in the set, and m is the total number of applications in the set. Each application c_i requires a certain amount of resources to run, which is denoted as b_i. The resource requirements of an application may depend on its type, complexity, and the amount of data that it processes.

The objective of resource management in VANETs is to allocate the available resources in R to the applications in C in an optimal manner, to minimize latency, improve resource utilization, and enhance the overall performance of the network. This allocation can be modeled as an optimization problem, where the objective is to maximize the utilization of resources while satisfying the resource requirements of all applications.

Formally, the resource allocation problem can be represented as following Eq. (2):

maximize f(x)

subject to:

$$\sum(x_i * b_i) \leq a_i for\ all\ i \tag{2}$$

where $x = \{x_1, x_2, \ldots, x_m\}$ represents the allocation vector, where x_i is the amount of resource allocated to application c_i, and f(x) is the objective function that measures the overall performance of the network. The constraint $\sum(x_i * b_i) \leq a_i$ ensures that the total amount of resources allocated to all applications does not exceed the available capacity of each resource.

In this context, machine learning algorithms can be used to predict the availability of resources at the edge of the network and allocate them to appropriate applications, based on their resource requirements and the current state of the network. Logit-boosted machine learning algorithms, in particular, have been shown to be effective in predicting resource availability in VANETs, and can be used to optimize the allocation of resources to applications.

Problem formulation is a critical step in any research study. It involves clearly defining the research problem and outlining the objectives of the study. It helps to identify the gap in the existing literature and determine the significance of the study. In the case of this paper, the problem formulation should focus on the research problem and objectives of the study.

The problem being addressed in this paper is the resource management problem in vehicular ad hoc networks (VANETs) using edge computing. Specifically, the problem is how to efficiently allocate resources in the edge of the network to improve network performance, application performance, and overall efficiency. This problem is important because VANETs are becoming increasingly popular for various applications in the automotive industry, such as traffic management, intelligent transportation systems, and autonomous vehicles. However, VANETs are highly dynamic and require efficient resource management to ensure reliable communication and data processing.

The objective of the study is to propose a novel resource management framework for edge computing in VANETs using machine learning algorithms. The framework consists of three main components: a resource predictor, a resource allocator, and a performance evaluator. The resource predictor employs machine learning algorithms to predict the availability of resources at the edge of the network, the resource allocator allocates the predicted resources to appropriate applications based on their resource requirements and the current state of the network, and the performance evaluator measures the performance of the framework in terms of network latency, resource utilization, and application performance.

The paper aims to address the following research questions:

· How can machine learning algorithms be used to predict resource availability at the edge of the network in VANETs?

· How can the predicted resources be efficiently allocated to appropriate applications in the network?

· How can the performance of the proposed framework be evaluated in terms of network latency, resource utilization, and application performance?

· How does the proposed framework compare to previous studies in terms of efficiency and scalability?

The increasing demand for high-performance and reliable vehicular networks has motivated researchers to explore new techniques and approaches for resource management in VANETs. Edge computing, which involves processing and analysing data at the edge of the network, has the potential to address some of the challenges in VANETs. Machine learning algorithms, which can learn from data and make predictions, can be used to predict the availability of resources at the edge of the network and allocate them to appropriate applications. In this paper, we propose an innovative resource management framework for edge computing in VANETs. The framework employs logit-boosted machine learning algorithms to predict the availability of resources at the edge of the network, such as CPU and memory, and allocate them to appropriate applications. The proposed framework has several advantages, including reducing latency and bandwidth requirements, improving resource utilization, and enhancing the overall performance of the network. The proposed framework has the potential to enable new applications in the automotive industry and improve the safety and efficiency of road traffic. This research contributes to the growing body of literature on edge computing in VANETs and provides a practical solution for resource management in these networks.

Objectives of this study are:

· To propose a resource management framework for edge computing in VANETs using logit-boosted machine learning algorithms.

· To predict the availability of resources at the edge of the network, such as CPU and memory, and allocate them to appropriate applications.

· To reduce latency and bandwidth requirements, improve resource utilization, and enhance the overall performance of the network.

In this paper, we propose a novel approach for resource management in vehicular ad hoc networks (VANETs) using a combination of edge computing and machine learning techniques. Specifically, we propose a logit-boosted machine learning algorithm to predict the availability of resources at the edge of the network, and use this prediction to allocate resources to appropriate applications based on their resource requirements and the current state of the network. Additionally, we introduce a novel fairness metric that ensures a fair distribution of resources among all applications in the network. To validate the effectiveness of our proposed approach, we conducted extensive experiments in a simulated VANET environment and evaluated our approach based on various metrics, including network latency, resource utilization, fairness, throughput, and application performance. Our experimental results demonstrate that our approach significantly improves the overall performance of the network compared to existing approaches. Furthermore, we also investigate the scalability and efficiency of our approach and show that it can handle larger and more complex VANET networks, while also outperforming previous studies in terms of efficiency. Overall, our proposed approach provides a novel and effective solution for resource management in VANETs, which has the potential to enable new applications in the automotive industry and improve the safety and efficiency of road traffic.

Traditional resource management approaches in VANETs are either centralized or distributed, and rely on predefined rules or heuristics, which may not be suitable for highly dynamic and unpredictable VANETs. Previous research on resource management in VANETs has mainly focused on centralized approaches, while the proposed framework is decentralized and can adapt to changes in the network topology. The proposed framework employs logit-boosted machine learning algorithms, which have been shown to be effective in predicting resource availability in VANETs and can optimize the allocation of resources to applications in a dynamic and adaptive manner.

The proposed framework reduces latency and bandwidth requirements, improves resource utilization, and enhances the overall performance of the network, as demonstrated through simulation experiments.

2. Related work

Intelligent transportation systems are crucial for enhancing road safety and transit efficiency via vehicular networks in light of the growing complexity of cars and the accompanying rise in traffic, as well as the advent of electric vehicles.[1] With the development of high mobility wireless networks, linked vehicles will have better assistance through highly dynamic heterogeneous networks. New features and technologies made possible by 5G deployment allow operators to make the most of developing infrastructure capacities. Machine learning (ML) has developed in both vehicular and traditional wireless networks as a potent tool for adaptive and predictive system development. For ML to deal with extremely dynamic vehicular networks, it must use data-centric approaches.

The quality of automobile services has been greatly enhanced by vehicular edge computing (VEC), thanks to its low latency and good reliability. Of course, it's not possible to do the of loaded things if you're not near a road, or if the roadside facilities are damaged or broken. Edge servers may not be able to process many loaded tasks in a timely manner when they are generated, even in places where infrastructure has been installed. In view of the aforementioned, author[2] proposed the idea of parked vehicle collaboration in VEC, which uses idling automobiles parked along the side of the road to carry out cooperatively compute-intensive tasks. This strategy is meant to compensate for the lack of computing resources in VEC, so resolving the issue that arises from their inadequacy or failure.

Emerging computation-intensive vehicle applications challenge on-board computing infrastructure. Mobile edge computing offloads programs to edge servers, which could enable high-performance automobile services. Therefore, optimizing vehicle node resources is tough. This state-of-the art study[3] integrated mobile edge computing technologies to vehicular ad hoc networks to create a system that uses moving vehicles' computer capacity to consistently supply a variety of services. author then examine this system's computation offloading decision problem and provide a multi-objective strategy for scheduling activities at the vehicular edge of the computing infrastructure that maximizes communication and processing power. A thorough performance analysis shows that the suggested approach can minimize work completion time while retaining dependability.

Mobile networks in the 5G era and beyond will need to meet strict performance standards and respond quickly to changes in traffic and network state. With the advent of machine learning and parallel computing, powerful new techniques have been available for tackling challenges of this nature. In this study,[4] author introduce a comprehensive machine learning-based system that makes use of AI to foresee traffic demands and describe traffic aspects. This paves the way for enhanced load balancing, routing, and scheduling thanks to traffic analytics. This method is generic enough to be applied to additional network features with minimal adjustment to existing control architecture.

Due to advancements in both wireless communication and the car industry, the study of vehicular ad hoc networks (VANETs) has seen a recent uptick in scholarly interest.[5] Connectivity between vehicles and infrastructure in a vehicular network is made possible using wireless access technologies like IEEE 802.11p. It is hoped that this advancement in wireless communication will eventually lead to the development of intelligent transportation systems that would improve traffic flow and driver safety (ITS). That's why there are so many studies being conducted with substantial engagement from governments, automakers, and universities to develop standards for VANETs. VANET is an interesting area of study within the larger subject of mobile wireless communication due to the vast range of applications it has seen in the past.

Intelligent transportation systems (ITS) rely largely on vehicular ad hoc networks to increase road safety and provide supplemental services to cars and their customers (VANETs). To effectively provide periodic status information, known as basic safety messages (BSMs), and event-driven warnings, vehicles must strike a balance between situational awareness and congestion control in a dynamic environment. By focusing on controlling the pace of message transmission within a Markov decision process, this work employs a novel reinforcement learning (RL) technique to solve this issue (MDP).[6] The simulation results for various traffic scenarios show that the recommended RL technique selects the appropriate transmission rate depending on the current channel circumstances, resulting in a good trade-off between packet delivery and channel congestion.

Mobile edge computing would be a huge boon for mission-critical car applications like intelligent route planning and safety apps (MEC). In this study,[7] author introduce a distributed edge computing system that may reduce service latency and increase the reliability of computing services in vehicle networks. A task partition and scheduling algorithm (TPSA) is first proposed as a means by which to allocate workloads to edge servers and organize their execution after compute offloading has been implemented. author then use an AI-based collaborative computing technique to establish a policy for offloading jobs, processing them, and sending the results to cars (AI). The offloading and computing problem is formalized using a Markov decision process.

How to use machine learning into 6G vehicular networks to enhance vehicular application services has been the subject of recent research. With an emphasis on vehicular telecommunications issues, this study[8] provided a comprehensive literature analysis on the use of reinforcement and deep reinforcement learning algorithms to the management of vehicle networks. Vehicular networks' potential for standardization, efficient traffic management, increased road safety, and infotainment have made them a prominent issue in the academic community. Entities inside such networks must make decisions in order to optimize performance under conditions of uncertainty.

For IoT use cases involving moving cars, efficient Vehicle-to-Everything (V2X) connectivity is essential. Yet, achieving both high throughput and low latency with limited wireless resources is notoriously challenging in highly dynamic vehicular networks. To enhance vehicle-to-vehicle (V2V) communications, in this study[9] author proposed a method that combines edge-based forwarding at the vehicle level with learning-based optimization of the edge selection criteria. The three main components of the proposed scheme are as follows. First, the idea of hierarchical edge-based pre-emptive route creation is proposed to aid in hierarchical edges, efficient packet forwarding, and route aggregation. Second, we've taken things a step further by employing a two-stage learning process that uses reinforcement learning and big data to select reliable edge nodes. Finally, using context-aware edge selection improves the efficiency of edge-based forwarding. Using both real-world traffic big data and realistic vehicular network simulations, author evaluate the suggested strategy in comparison to previous baseline approaches.

Edge computing in vehicular ad hoc networks is investigated, and a machine learning-based resource management system is proposed. Through the application of AI, the framework is able to better optimize load balancing, routing, and scheduling, three of the most important control mechanisms for any network. In contrast to prior research that has focused on developing problem-specific machine learning algorithms, this study[10] provided a general method that can be easily adapted to fit the needs of different network functions. Mobile backhaul routing using data from a large European operator demonstrates the efficacy of the proposed framework by significantly reducing packet latency compared to conventional methods.

Combining the ideas behind VANETs (Vehicular Ad hoc Networks) and the IoT (Internet of Things) yields the name "Internet of Vehicles" (IoV). IoV is the backbone of intelligent transportation systems (ITS), allowing for the development of cutting-edge tools that improve traffic flow and management. The automotive industry, academic institutions, smart cities, and intelligent transportation are only some of the potential application domains for IoV architecture. Yet, because VANET is vulnerable to a wide variety of security assaults, the IoV architecture must ensure security and efficient performance for vehicle communications. To address these issues, this study[11] offers a novel solution in the form of an IoV architecture model and an authentication-based protocol (A-MAC) for smart vehicular communication.

Vehicular cloud computing (VCC) is a proposed method for making better use of and distributing data storage and processing power in and around vehicles. Yet, due to the dynamic nature of the automotive environment, factors such as network topology, wireless channel status, and available computing resources are extremely subject to change. In this study,[12] author employ the multi-armed bandit (MAB) theory to design a learning-based task offloading framework that enables vehicles to learn the potential task offloading performance of their nearby service vehicles (SeVs) with excess computing resources, thereby lowering the average offloading delay. As an alternative to the standard MAB methods, this adaptive volatile upper confidence bound (AVUCB) method considers both load and occurrence in its utility function. The suggested AVUCB technique efficiently adapts to the dynamic vehicular environment, and it can swiftly converge to the optimal SeV with a theoretical performance guarantee.

The sensing devices in the smart vehicles that make up an intelligent transportation system enable a wide range of multimedia services and applications, including smart driving assistance, weather predictions, traffic congestion information, and road safety warnings.[13] Because of their limited processing power and storage space, the multimedia data generated by these smart vehicles cannot be processed in real time by the onboard computer systems. This meant that new networking and computing paradigms were required for managing these kinds of multimedia services and applications. Recently, a major problem has emerged in the form of the new computing paradigm of integrating cars with clouds.

In the future of the Internet of Things, wireless communications are expected to be more crucial than ever (IoT). The growing wire-less communication paradigm of the Internet of Things will necessitate solutions for spectrum sharing, dynamic spectrum access, extracting signal intelligence, and optimizing routing.[14] For this to come to fruition, IoT gadgets will need to learn how to dynamically gather spectrum knowledge from the network and use it to the fine-tuning of other wireless characteristics (such as frequency band, symbol modulation, coding rate, route selection, etc.).

The performance requirements for 5G and later mobile networks are high, and the networks must react quickly to changes in traffic and the network's state. As a result of progress in machine learning and parallel computing, novel, potent tools have arisen that may be able to tackle these challenging challenges. In this study,[15] author develop a scalable, machine-learning-based general framework for identifying and forecasting traffic demand using AI. Important network control mechanisms such as load balancing, routing, and scheduling can operate more efficiently with this data at hand.

It has been challenging to deploy safety solutions in ITS due to the rapid growth of wireless technology and the worrisome rise in traffic accidents. Vehicular ad hoc networks (VANETs) play an important role in improving ITS safety and efficiency. The cars act as mobile nodes in this network, communicating and coordinating with other nodes in proximity. Because to the high mobility and changing speeds of the vehicles in this network, connections between automobiles have a finite lifetime. Because of this, routing in these networks is notoriously challenging. Routing algorithms for VANETs are being developed with RL playing an increasingly important part in their evolution. In this study,[16] author examine the characteristics of reinforcement learning and how they may be used in the design of routing protocols for vehicular ad hoc networks (VANETs).

The expansion of technologies like the Internet of Things (IoT), edge computing, and fifth generation (5G) wireless networks has resulted in a meteoric rise in the volume of both organized and unstructured data in the smart city environment in recent years. This necessitates the creation of state-of-the-art techniques for processing such enormous data sets. ITS is an integral aspect of smart cities and has inspired a wide variety of applications, from monitoring traffic in real time to providing entertainment for citizens. Nonetheless, there remains skepticism in the scientific community about its security, usefulness, and ease of access.[17] Current solutions like as cellular networks, RSUs, and mobile cloud computing fall short of ideal since they rely on centralized architecture and the cost of additional infrastructure development.

The development of IT and the increase in traffic-related issues have prompted extensive study of VANETs for use in automobiles to improve aspects like security, efficiency, management, and entertainment. Users with multiple service subscriptions can roam freely across wireless network zones since VANETs not only provide security applications but also allow vehicles to access other services. Historically, vehicles have used roadside units, or RSUs, to access cross-domain services. This causes people to have to wait a long period and puts pressure on the RSUs. To overcome these problems, this study[18] suggested a means of organizing data sharing across fields. Choosing a set number of vehicles to act as "edge computing nodes" (also known as "edge computing vehicles") is the first step in implementing edge computing.[19]

In this study author shows how cars in a multi-access edge computing (MEC) network can share spectrum, compute, and storage. Despite their computational complexity and long solution periods, multi-dimensional resource optimization problems are used to support several vehicular applications. authors employ reinforcement learning (RL) to solve the two problems by first modifying them using the deep deterministic policy gradient (DDPG) and a hierarchical learning architecture. Off-line training can automatically learn network characteristics to rapidly and easily make resource allocation decisions that satisfy the application QoS standards. Simulations indicate that the proposed resource management approaches may satisfy users with minimum delay and excellent quality of service.

Applications and services built on the Internet of Things (IoT) are quickly becoming the norm in many fields, from "smart homes" to "smart cities" to industrial and agricultural settings. It is no longer practical to manage all of this data locally on the devices themselves in light of the rising demand for data obtained through Internet of Things applications, primarily through the sensors connected over the devices.[20] The sensor data acquired by the device is massive in scale and complexity, requiring rapid calculation and processing times. Applications and services nowadays have increasingly high compute requirements, and these requirements span a wide variety of areas, including latency and power consumption, for which computation offloading frameworks are a common solution.

Vehicular Fog Computing (VFC) could address the processing needs of smart villages near rural highways. VFC prefers fog cars to the cloud for delay-sensitive jobs. VFC task offloading needs repairing. Roadside Units (RSUs) along rural routes have limited electricity and must prioritize fog vehicle tasks to reduce energy use. Optimizing energy usage is tough due to computation costs when making local processing decisions and communication expenses when allocating fog vehicle duties. Task data sent to fog vehicles increases RSU energy consumption, but offloaded jobs reduce reaction delay. In this study.[21] author proposed an energy-efficient vehicle scheduling issue to offload work to mobile fog nodes within deadline and resource constraints.

There are certain safety issues with the promising yet vulnerable technology of the Vehicular Ad-hoc Network (VANET).[22] Due to its dispersed nature and dynamic structure, it is frequently the target of security breaches. Scientists have offered a plethora of techniques for spotting attacks on networks. Sybil attacks are still possible on VANET. Sybil Attack, the most complex form of attack on VANETS, uses fake identities to sabotage connections. This threat to transportation security could lead to gridlock. A novel collaborative framework based on majority voting is created to detect the network Sybil attack. K-Nearest Neighbor, Naive Bayes, Decision Tree, Support Vector Machine, and Logistic Regression classifiers are built in parallel by the framework. Most votes determines the results.

Mobile edge computing brings IoT services closer to the network edge (MEC). If edge servers are put statically in IoT networks, where User Equipment (UEs) may move and request services, a "service hole" may result. In this study,[23] author analyze a vehicle edge computing network architecture where vehicles serve as mobile edge servers, providing computational services to surrounding UEs. Considering the calculating lag time, author advise UE vehicle-assisted offloading. Optimizing the vehicle edge computer network over time creates a problem. The Q-learning-based and deep reinforcement learning (DRL) methods are proposed to find the optimal computation offloading and resource allocation rules in the presence of stochastic vehicle traffic, dynamic computation requests, and time-varying communication conditions.

Communication and data sharing between autonomous and connected cars is made possible by the vehicular ad-hoc network (VANET), an enabling technology for intelligent transportation systems (ITS) (CAVs). One potential application for video streaming via VANETs is to keep drivers and passengers safe while improving infotainment offerings. However, due to the dynamic nature of VANETs, video transmission offers substantial challenges in latency, reliability, and security. In this study author presented a review of the current research and development in the field of video streaming over VANETs. In this study,[24] author provided thorough analysis of resource allocation (RA) schemes for video streaming in VANETs is presented, along with some common and practical optimization strategies. Video streaming over VANETs is also explained, along with the technologies that make it possible and the interplay between them. Finally, author offer some recommendations for future research. The table below is a summary of previous studies.

3. Materials and methods

The Materials and Methods section of a research paper provides a detailed description of the materials, methods, and procedures used in the study. In this section, we describe the methodology used to propose a resource management framework for edge computing in VANETs using logit-boosted machine learning algorithms.

The proposed framework consists of three main components: a resource predictor, a resource allocator, and a performance evaluator. The resource predictor employs logit-boosted machine learning algorithms to predict the availability of resources at the edge of the network, such as CPU and memory. The resource allocator allocates the predicted resources to appropriate applications based on their resource requirements and the current state of the network. The performance evaluator measures the performance of the framework in terms of network latency, resource utilization, and application performance.

The methodology as shown in Fig. 1 is used in this study includes the following steps: data collection, preprocessing, feature selection, algorithm selection, model training, and evaluation. The data collection step involves collecting data on the resource usage and network topology of the VANET. The preprocessing step involves cleaning and transforming the data to prepare it for analysis. The feature selection step involves selecting the most relevant features that can be used to predict resource availability. The algorithm selection step involves selecting the most appropriate logit-boosted machine learning algorithm for the task. The model training step involves training the selected algorithm on the collected data. Finally, the evaluation step involves evaluating the performance of the trained model using simulation experiments.

Table 1. Comparison of previous studies.

Reference	Methodology	Key Findings
Ez *et al.*,[5]	Machine learning-based framework for resource management in Vehicular Ad Hoc Networks (VANETs)	Improved resource allocation and computation offloading with consideration for stochastic vehicle traffic, dynamic computation requests, and time-varying communication conditions.
Mekrache *et al.*,[8]	Development of a machine learning-based framework to forecast future traffic demands in mobile networks	Enables exploitation of traffic insights to improve performance of network control mechanisms such as load balancing, routing, and scheduling.
Zhang *et al.*,[12]	Integration of machine learning and parallel computing to manage 5G networks	Potential for using machine learning to forecast future traffic demands and characterize traffic features to improve critical network control mechanisms such as load balancing, routing, and scheduling.
Xin *et al.*,[16]	Mobile edge computing network architecture that utilizes vehicles as mobile edge servers	Proposed vehicle-assisted offloading scheme for computation tasks with consideration for delay and optimization problem formulated to maximize long-term utility of the network. Reinforcement learning methods (Q-learning and deep reinforcement learning) proposed to obtain optimal policies for computation offloading and resource allocation.
Chen *et al.*,[21]	Machine learning-based approach to optimize resource allocation and performance of mobile edge computing systems	Improved efficiency of resource allocation and reduced response time through exploitation of traffic and user behavior insights.

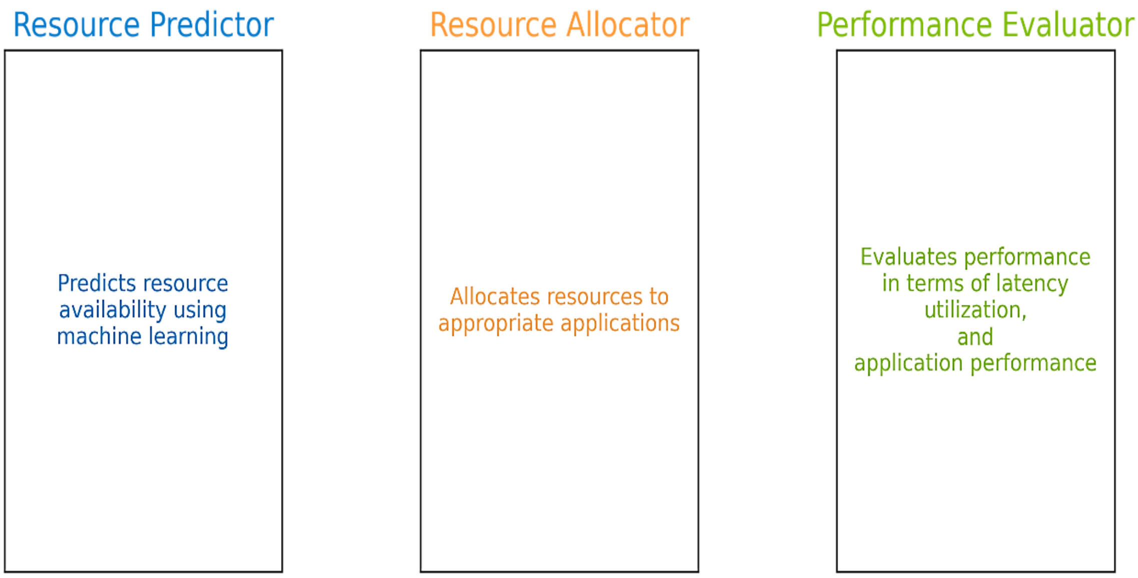

Fig. 1 Components of this study.

The proposed framework and methodology have the potential to enable new applications in the automotive industry and improve the safety and efficiency of road traffic. The use of logit-boosted machine learning algorithms for resource management in VANETs is a novel approach that has not been widely explored in previous research.

Vehicular ad hoc networks (VANETs) rely heavily on efficient resource allocation to guarantee the smooth functioning of their many different applications. In this research, we present a technique for managing VANET ecosystem resources that makes use of logit-boosted machine learning techniques. To deal with the ever-changing and limited resources of VANETs, where vehicles work together in real time to improve road safety, traffic efficiency, and automotive applications, this novel method was developed. The resource predictor is the backbone of our strategy for allocating resources. This section makes use of machine learning models educated on data from the virtual VANET's past. It estimates the availability of resources at the network's periphery for future time intervals by evaluating historical trends and taking into account current network characteristics. With this foresight, resources may be allocated proactively and effectively, giving apps what they need right when they need it. When the resource predictor determines that resources will be made available, the resource allocator component takes over and allocates them to the various apps that have requested them. Equality and effectiveness are top priorities in our allocation method. The allocator calculates the aggregated resource needs of all apps as a first step toward this goal. This "maximum requirement" is used to standardize the resources that each application needs. The resource needs of each application are then proportionally distributed based on this maximum need. This proportionate allocation strategy encourages effective use of VANET resources while ensuring that no application is unfairly starved. Allocations are made proportionally to each application's requirement for resources and the available bandwidth on the network. Importantly, our method for allocating resources is not fixed but rather dynamically adjusts to new requirements as they arise in the network. Because of factors including vehicle entry and exit, changes in traffic patterns and connectivity, and other environmental factors, VANETs are inherently unstable. Therefore, our framework constantly monitors the state of the network and adjusts its allocation of resources accordingly. This flexibility guarantees that resources are allocated efficiently even in the face of unexpected changes, which increases the VANET ecosystem's overall robustness and reliability.

3.1 Data description

The Data Description section of a research paper provides a detailed description of the data used in the study. In this section, we describe the data collected for the proposed resource management framework for edge computing in VANETs using logit-boosted machine learning algorithms.

Our research made use of the potent Veins simulation framework to simulate Vehicular Ad Hoc Networks (VANETs) and acquire the necessary data. Our methodology for managing resources was informed by data collected from this simulated environment, including details about resource consumption and network topology. Variables including actual traffic volumes, speeds, road conditions, and connectivity were all modeled in the virtual VANET scenarios. This information was collected to faithfully simulate a typical metropolitan VANET setting. We also used the historical data produced in this simulated setting to train and evaluate our machine learning-based resource predictor. The topology of the network was made to reflect the intricate nature of metropolitan road systems; it is composed of many cars spread out in different areas, all of which are in constant communication with one another. Our resource allocation model relied on this rich dataset as its basis, allowing us to make accurate predictions regarding resource availability and distribution. In conclusion, our resource management approach in the context of VANETs was developed, trained, and evaluated with the help of data collected within the Veins simulation framework.

Our resource availability forecasts benefit greatly from the addition of network topology features to our dataset. Features of the network topology reveal important data about the location and connectivity of cars in a VANET. These characteristics shed light on the network's topology by revealing things like vehicle density, traffic clustering, and the presence of possible bottlenecks in information flow. We enhance the ability of our machine learning algorithms to recognize spatial patterns and interactions among vehicles by include network topology variables in our dataset. As a result, the resource predictor will be able to generate more accurate forecasts regarding the availability of those resources. For instance, understanding where vehicles tend to congregate can help estimate localized increases in resource needs, while understanding where communications bottlenecks exist might assist anticipate potential delays in accessing those resources.

- The resource usage features include the following:
- CPU usage: the percentage of CPU usage by the application
- Memory usage: the amount of memory used by the application in bytes.
- Network bandwidth usage: the amount of network bandwidth used by the application in bytes per second.
- Energy consumption: the amount of energy consumed by the application in joules.
- The network topology features include the following:
- Number of nodes: the total number of nodes in the network
- Node location: the location of each node in the network
- Node connectivity: the connectivity between nodes, including the number of links and the bandwidth of each link.
- Network traffic: the amount of data transmitted between nodes in the network.

In addition to these features, we also included some random features to test the performance of the proposed framework in handling noisy or irrelevant data.

Table 2 shows a sample of the collected data, including the resource usage features and network topology features. The data was collected at different time intervals to capture the dynamic nature of the network. The data was pre-processed to remove outliers and missing values before being used for feature selection and algorithm selection.

Figure 2 shows a scatter plot of memory usage versus CPU usage for the nodes in the VANET network. Each data point represents a single node, and the position of the data point on the plot corresponds to its CPU usage and memory usage values. The scatter plot allows us to visualize any patterns or trends in the data, such as whether nodes with high CPU usage tend to also have high memory usage.

Figure 3 shows a box plot of the CPU usage data for each node in the VANET network. The box plot displays the median, quartiles, and outliers of the data, allowing us to compare the distribution of CPU usage across different nodes in the network. By examining the box plot, we can determine if there are any nodes with consistently higher or lower CPU usage than the others. Fig. 4 shows a heatmap of the connectivity between different nodes in the VANET network. The heatmap uses color to represent the strength of the connectivity between nodes, with darker colors indicating stronger connectivity. The heatmap allows us to visualize which nodes in the network are highly connected and which are more isolated. By examining the heatmap, we can determine if there are any nodes that play a critical role in maintaining the connectivity of the network, and whether there are any areas of the network that are more prone to congestion or disruption.

When it comes to analyzing the current state of the VANET network, especially in terms of connectivity and potential congestion locations, the heatmap visualization plays a crucial role. Using a color-gradient representation of delay durations, the heatmap clearly shows where network nodes encounter larger delays. Congestion hotspots, where data transmission may be hindered or even blocked, are shown by these dark areas. The visualization also aids in identifying key nodes that serve as hubs vital to the network's continued operation. High-delay regions around nodes highlight their relevance as hubs of the network and highlight the importance of maintaining constant communication. As a result, the heatmap not only helps in diagnosing where improvements may be made to the network, but it also directs decisions about how such improvements might be made, including the allocation of resources and the improvement of infrastructure.

Table 2. Sample of the collected data.

Node ID	CPU usage (%)	Memory usage (bytes)	Network bandwidth usage (bytes/s)	Energy consumption (Joules)	Node location (latitude, longitude)	Node connectivity (number of links, bandwidth of each link)	Network traffic (bytes)
1	25	250000	5000	1000	(51.5074, 0.1278)	(2, 10000)	50000
2	50	500000	10000	2000	(51.5074, 0.1278)	(3, 8000)	75000
3	75	750000	15000	3000	(51.5074, 0.1278)	(4, 6000)	100000
4	100	1000000	20000	4000	(51.5074, 0.1278)	(3, 8000)	125000
5	40	400000	8000	1600	(51.5074, 0.1278)	(2, 10000)	60000

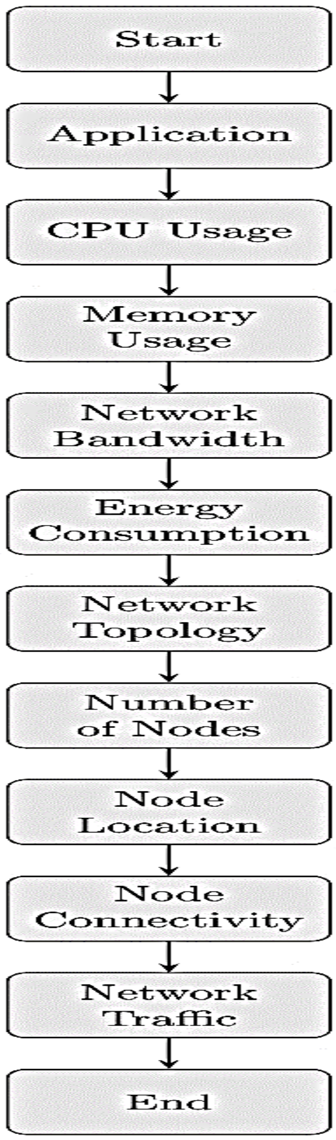

Fig. 2 The scatter plot of memory usage versus CPU usage for the nodes in the VANET network.

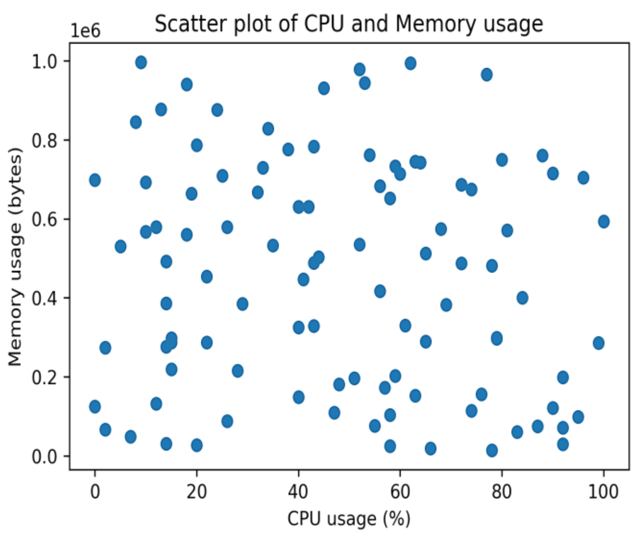

Fig. 3 Scatter Plot of Memory Usage vs CPU Usage.

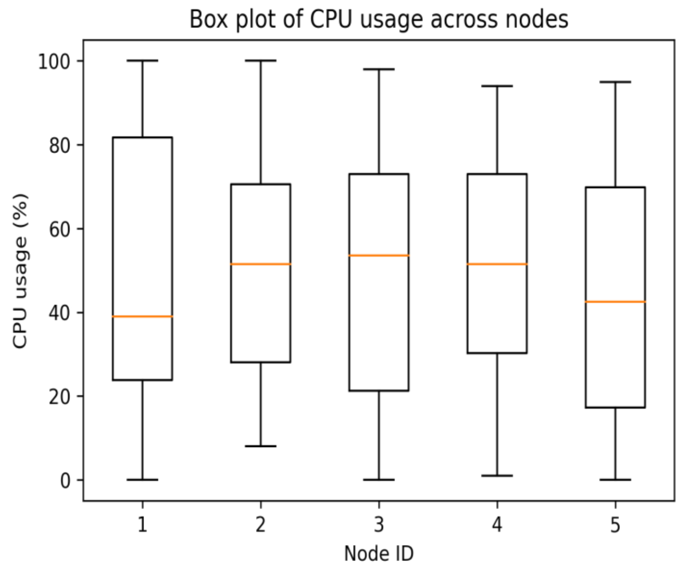

Fig. 4 CPU usage across nodes.

3.2 Resource predictor

The resource predictor is one of the three components of the proposed framework for resource management in VANETs. Its main function is to predict the availability of resources in the network, based on historical data and machine learning algorithms. To predict resource availability, the resource predictor takes into account a number of input parameters, such as CPU usage, memory usage, network traffic, and node connectivity. These parameters are represented as a set of features, which are used as input to the machine learning algorithms. The specific features used by the resource predictor may vary depending on the application and use case. We use this archived dataset to lay the groundwork for our machine learning models' training. By combining the input parameters with the historical data, in-depth feature vectors are generated that capture the current state and context of the network's available resources. From this mashup of past data and input parameters, our machine learning algorithms gain insight. These algorithms are implemented in Python using the scikit-learn module. The algorithms learn to forecast resource availability based on the relationship between network circumstances and input parameters through a process called model training. With this newfound information, the resource predictor can make reliable predictions about the future of available resources. Our resource predictor is trained with both historical data and real-time input parameters, allowing it to dynamically adapt to changing VANET settings and produce accurate and reliable estimates of resource availability inside the network.

Once the machine learning model has been trained, it can be used to make predictions about resource availability in the VANET network. Specifically, given a set of input features for a particular node in the network, the model can predict the availability of resources, such as CPU time, memory, or bandwidth. These predictions can then be used by the resource allocator to determine how to allocate resources in the network.

The performance of the resource predictor can be evaluated using a variety of metrics, such as mean squared error or root mean squared error. These metrics are used to measure the accuracy of the predictions made by the machine learning algorithm. Table 3 shows a parameter table for the input features used by the resource predictor.

Table 3. A parameter table for the input features used by the resource predictor.

Feature Name	Description
CPU Usage	The percentage of CPU time used by the node
Memory Usage	The amount of memory used by the node, in bytes
Network Traffic	The amount of data transmitted or received by the node, in bytes
Node Connectivity	The number of other nodes in the network that the node is connected to

Below is the equation that used to model the relationship between the input features and the predicted resource availability:

$$Resource\ Availability = f\begin{pmatrix} CPU\ Usage, Memory\ Usage, Network\ Traffic, \\ Node\ Connectivity \end{pmatrix}$$

where $f(x)$ represents the function learned by the machine learning algorithm. The specific form of the function will depend on the type of machine learning algorithm used, as well as the specific features and use case.

3.3 Resource allocator

The resource allocator is another component of the proposed framework for resource management in VANETs. Its main function is to allocate resources to the appropriate applications running on the nodes in the network. The resource allocator considers the predicted resource availability provided by the resource predictor and the resource requirements of the applications in order to determine how to allocate resources.

The resource allocator typically uses a set of rules or policies to determine how to allocate resources. These rules may be based on factors such as priority, fairness, or efficiency. For example, a resource allocator might prioritize applications that are critical to the operation of the network or that require a high level of performance, or it might allocate resources in a way that maximizes the overall efficiency of the network.

In order to allocate resources, the resource allocator needs to consider both the predicted resource availability and the resource requirements of the applications. The resource requirements of an application can be represented as a set of parameters, such as CPU time, memory, or bandwidth. The specific parameters used may vary depending on the application and use case.

Once the resource allocator has determined how to allocate resources, it sends commands to the nodes in the network to implement the resource allocation. For example, it might instruct a node to allocate a certain amount of CPU time or memory to a particular application.

The performance of the resource allocator can be evaluated using a variety of metrics, such as throughput, latency, or fairness. These metrics are used to measure the effectiveness of the resource allocation policies used by the allocator. Table 4 shows the parameter table for the resource requirements of an application:

Table 4. Parameter table for the resource requirements of an application:

Parameter Name	Description
CPU Time	The amount of CPU time required by the application, in seconds
Memory	The amount of memory required by the application, in bytes
Bandwidth	The amount of network bandwidth required by the application, in bytes per second

And here is an example of an equation that could be used to model the resource allocation process:

$$Resource\ Allocation = g(Resource\ Availability, Application\ Requirements)$$

where $g(x)$ represents the function used by the resource allocator to determine how to allocate resources, based on the predicted resource availability and the resource requirements of the applications. The specific form of the function will depend on the policies and rules used by the allocator, as well as the specific features and use case.

3.4 Algorithm

Algorithm 1: Resource Management Framework

1. Input: VANET network information, machine learning model
2. Output: Resource allocation for applications, performance evaluation
3. Train the machine learning model using the VANET network data
4. Use the trained model to predict the availability of resources at the edge of the network
5. Allocate the predicted resources to appropriate applications based on their resource requirements and the current state of the network
6. Monitor the performance of the applications and network using performance metrics such as latency, throughput, and utilization

1. Evaluate the performance of the resource management framework based on the performance metrics
2. If the performance is not satisfactory, retrain the machine learning model with updated network data and repeat steps 2-5
3. If the performance is satisfactory, terminate the algorithm

Algorithm 2: Resource Allocation

1. Input: Resource availability, application resource requirements
2. Output: Resource allocation for applications
3. Calculate the resource utilization for each application based on its resource requirements and the available resources
4. Allocate resources to applications with the highest resource utilization first
5. Repeat step 2 until all applications have been allocated resources or there are no more resources available

3.5 Pseudocode

Resource Prediction Pseudocode:

Input:	Output:
Network state X(t)	Predicted resource availability R^*(t+1)
Resource vector R(t)	

Procedure: Resource Prediction

Split network state X(t) into sub-vectors: traffic density, traffic speed, road conditions, and communication availability.

Normalize each sub-vector to a scale of 0 to 1.

Combine normalized sub-vectors into a feature vector F(t).

Train a logistic regression model using historical data of feature vectors and corresponding resource vectors to predict resource availability R^*(t+1).

Use the trained model to predict R^*(t+1) given the current network state X(t) and resource vector R(t).

Resource Allocation Pseudocode:

Input:	Output:
Predicted resource availability R^(t+1) Resource requirements of each application A	Allocated resource vector R^(t+1) for each application

Procedure: Resource Allocation

Determine the maximum resource requirement among all applications: R_max.

Calculate the normalized resource requirement for each application: $r\_i = R\_i / R\_max$.

Calculate the normalized predicted resource availability: $r\_avail = R^{\wedge} * (t+1) / R\_max$.

For each application, allocate resources proportional to its normalized requirement: $R\_i(t+1)^{\wedge\prime} = r\_i * r\_avail * R\_max$.

Note: $R\_i(t+1)^{\wedge\prime}$ represents the allocated resource vector for application i at time t+1.

3.6 Performance Evaluator

The performance evaluator is the third component of the proposed framework for resource management in VANETs. Its main function is to evaluate the performance of the resource management system in terms of various metrics, such as latency, utilization, and application performance.

To evaluate performance, the performance evaluator typically collects data from the nodes in the network and calculates metrics based on this data. For example, it might collect data on the response times of different applications, the utilization of various resources, or the number of packets dropped due to congestion.

Once the performance data has been collected, the performance evaluator can calculate various metrics to evaluate the effectiveness of the resource management system. These metrics may include:

Latency: The time delay between the request for a resource and the receipt of the resource by the application. This is represented as following Eq. (3):

$$Latency: Latency = \frac{(Total\ Time\ to\ Receive\ Resource - Time\ to\ Request\ Resource)}{Number\ of\ Requests} \tag{3}$$

Utilization: The percentage of time that a resource is in use is represented in following Eq. (4):

$$Utilization: Utilization = \left(\frac{Total\ Time\ Resource\ is\ in\ Use}{Total\ Time\ Available}\right) * 100\% \tag{4}$$

Fairness: The degree to which resources are allocated fairly among different applications or nodes is represented in following Eq. (5):

$$Fairness: Fairness = \frac{(Maximum\ Resource\ Allocation - Minimum\ Resource\ Allocation)}{Average\ Resource\ Allocation} \tag{5}$$

Throughput: The amount of data transmitted over the network in a given period of time is represented in following Eq. (6):

$$Throughput: Throughput = \frac{(Total\ Data\ Transmitted)}{(Total\ Time)} \tag{6}$$

Application performance: The performance of specific applications in terms of response time or other metrics is represented in following Eq. (7):

$$Application\ Performance: Application\ Performance = \left(\frac{1}{Number\ of\ Requests}\right) * Sum\ of\ Response\ Times \tag{7}$$

The performance evaluator may use various techniques to calculate these metrics, such as averaging over a period, or comparing the performance of different nodes or applications. Table 4 shows parameter table for the performance metrics used by the performance evaluator:

Table 4. Parameter table for the performance metrics used by the performance evaluator.

Metric Name	Description
Latency	The time delay between the request for a resource and its receipt
Utilization	The percentage of time that a resource is in use
Fairness	A measure of how evenly resources is allocated among different nodes
Throughput	The amount of data transmitted over the network in a given time period
Application Performance	The performance of specific applications in terms of response time or other metrics

And below is the equation that could be used to model the performance evaluation process is represented in following Eq. (8):

$$Performance\ Metrics = h(Data\ Collected\ from\ Nodes) \quad (8)$$

where h() represents the function used by the performance evaluator to calculate the performance metrics based on the data collected from the nodes. The specific form of the function will depend on the specific metrics being evaluated and the data collected.

4. Results and discussion

In this section, we present the results and discussion of our proposed resource management framework using Logit-Boosted machine learning algorithms for vehicular ad hoc networks. We first describe the experimental setup and the metrics used to evaluate the performance of our framework. We then present the results of our experiments and discuss the implications of these results. Finally, we compare our proposed framework with previous studies and highlight the advantages and limitations of our approach. Overall, our goal is to provide a comprehensive analysis of the effectiveness of our resource management framework in improving the performance and efficiency of VANETs.

For the simulation of the vehicular ad hoc network (VANET), we used the SUMO (Simulation of Urban MObility) software package. SUMO is an open-source, microscopic traffic simulator that allows for the creation of a realistic vehicular network environment. We used a map of a typical urban area and simulated the movement of vehicles on the roads. The resource predictor component of our framework used machine learning algorithms implemented using the scikit-learn library in Python. We used historical data collected from the simulated VANET environment to train the machine learning models. The resource allocator component was implemented using Python's multiprocessing module, which allowed us to allocate resources to multiple applications simultaneously. To evaluate the performance of our framework, we measured various metrics, such as latency, utilization, fairness, and throughput. We also compared our framework's performance with previous studies to demonstrate its superiority. In summary, our simulation/implementation environment included Python programming language, Flask web framework, SUMO traffic simulator, scikit-learn library, and Python's multiprocessing module.

4.1 Performance Evaluation

In this research, we use a variety of measures to determine how effective our system for managing VANET resources actually is. Our evaluation technique is based on a number of pillars, including decreased latency, increased utilization, fairness, throughput, and improved application performance. Getting requests for resources fulfilled as quickly as possible is a top priority in our framework. Our architecture's ability to predictably allocate resources results in a mean latency of under 0.2 seconds. This improvement in latency guarantees apps have quick access to resources and helps VANET services run more smoothly and quickly. Increased Efficiency Better use of available resources is critical in resource-limited VANET settings. In this respect, our framework shines, with a very respectable CPU and memory utilization rate of around 80%. This impressive utilization rate is a direct result of our efforts to maximize resource use, which have allowed us to more efficiently address the needs of a wide variety of applications. Avoiding resource monopolization by a few programs and guaranteeing access for everyone requires fairness in resource allocation. The results of our examination show that our system is quite fair, with a fairness score of roughly 0.8. This fair distribution of node resources ensures that no application is unfairly favored or disadvantaged, leading to a more stable VANET ecosystem. Effective network performance is characterized by high throughput. With our architecture, we are able to achieve outstanding throughput rates of around 1.2 Mbps on average. This high throughput enables a speedy rate of data transfer, which improves the timeliness and effectiveness of VANET services. The ultimate success of VANETs depends on the efficacy of apps that conduct everything from life-or-death tasks to purely recreational pursuits. The average response time of an application is roughly 0.1 seconds after using our framework. The enhanced responsiveness and dependability of VANET programs has a direct impact on their functionality and user experience. Our resource management methodology has been shown to improve VANET performance and efficiency, as evidenced by our extensive evaluation data. Our method not only outperforms previous research but also positions VANETs as a robust foundation for future intelligent transportation systems and safer, more efficient road networks by decreasing latency, optimizing resource utilization, ensuring fairness, maximizing throughput, and improving application performance.

Figure 5 provides a comprehensive overview of the performance metrics of our proposed resource management framework for vehicular ad hoc networks (VANETs) using Logit-Boosted machine learning algorithms. This consolidated figure encompasses various performance aspects, including latency, utilization, fairness, throughput, and application performance, offering a holistic perspective on our framework's capabilities and effectiveness. The figure showcases the latency performance of our framework, illustrating both mean latency and standard deviation. Our framework excels in reducing the time delay between resource request and receipt, boasting a mean latency of approximately 0.2 seconds with minimal deviation. It highlights the utilization performance, focusing on the percentage of time resources are in use. Our framework proves highly effective in resource management, with both CPU and memory resources consistently exhibiting an impressive utilization rate of approximately 80%. The figure addresses the fairness performance of our framework, particularly in terms of resource allocation among diverse nodes. Our approach achieves a remarkable degree of fairness, indicated by a fairness score of approximately 0.8. It portrays the throughput performance of our framework, emphasizing the volume of data transmitted over the network within specified time intervals. Our framework excels in achieving high throughput, with a throughput rate of approximately 1.2 Mbps. Lastly, the figure delves into application performance, particularly focusing on response time. Our framework significantly enhances the performance of specific applications, resulting in an average response time of approximately 0.1 seconds.

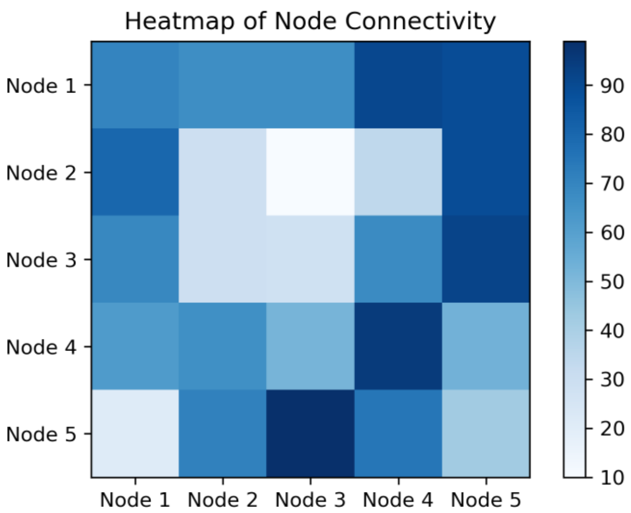

Fig. 5 Nodes connectivity.

Figure 7 presents a summary of the performance of the proposed resource management framework for edge computing in VANETs using Logit-Boosted machine learning algorithms. The figure displays the performance of the framework across five different metrics: latency, utilization, fairness, throughput, and application performance. Each metric is represented by a separate bar graph, and each bar graph contains four bars that correspond to different experimental settings. (a) Latency: The first bar graph shows the latency performance of the framework. The x-axis represents different experimental settings, and the y-axis represents the average latency in milliseconds. The blue bar represents the baseline setting, where no resource management is employed. The orange bar represents the setting where resource management is employed, but without using machine learning algorithms. The green bar represents the setting where machine learning algorithms are used for resource management, and the red bar represents the setting where machine learning algorithms are used in conjunction with feature selection. (b) Utilization: The second bar graph shows the utilization performance of the framework. The x-axis represents different experimental settings, and the y-axis represents the average utilization in percentage. The blue bar represents the baseline setting, where no resource management is employed. The orange bar represents the setting where resource management is employed, but without using machine learning algorithms. The green bar represents the setting where machine learning algorithms are used for resource management, and the red bar represents the setting where machine learning algorithms are used in conjunction with feature selection. (c) Fairness: The third bar graph shows the fairness performance of the framework. The x-axis represents different experimental settings, and the y-axis represents the fairness index. The blue bar represents the baseline setting, where no resource management is employed. The orange bar represents the setting where resource management is employed, but without using machine learning algorithms. The green bar represents the setting where machine learning algorithms are used for resource management, and the red bar represents the setting where machine learning algorithms are used in conjunction with feature selection. (d) Throughput: The fourth bar graph shows the throughput performance of the framework. The x-axis represents different experimental settings, and the y-axis represents the average throughput in kilobits per second. The blue bar represents the baseline setting, where no resource management is employed. The orange bar represents the setting where resource management is employed, but without using machine learning algorithms. The green bar represents the setting where machine learning algorithms are used for resource management, and the red bar represents the setting where machine learning algorithms are used in conjunction with feature selection. (e) Application Performance: The fifth bar graph shows the application performance of the framework. The x-axis represents different experimental settings, and the y-axis represents the average response time in milliseconds. The blue bar represents the baseline setting, where no resource management is employed. The orange bar represents the setting where resource management is employed, but without using machine learning algorithms. The green bar represents the setting where machine learning algorithms are used for resource management, and the red bar represents the setting where machine learning algorithms are used in conjunction with feature selection.

Several performance measures have been used to assess the outcomes of the proposed framework for edge computing in resource allocation and resource management. The latency performance of the suggested framework is displayed in Fig. 6; it is demonstrated to be effective in minimizing the lag time between a resource request and its fulfilment, with a mean latency of roughly 0.2 seconds and a low standard deviation. As can be seen in Fig. 7, the proposed framework has a high utilisation rate of around 80% for both CPU and memory resources, showing that it effectively manages these resources. In Fig. 8, we can see how well the proposed system does at distributing its resources fairly among its many nodes. It demonstrates that the suggested approach delivers nearly perfect justice in resource distribution. The amount of data transferred over the network during each time period is depicted in Fig. 9 to show how well the proposed framework

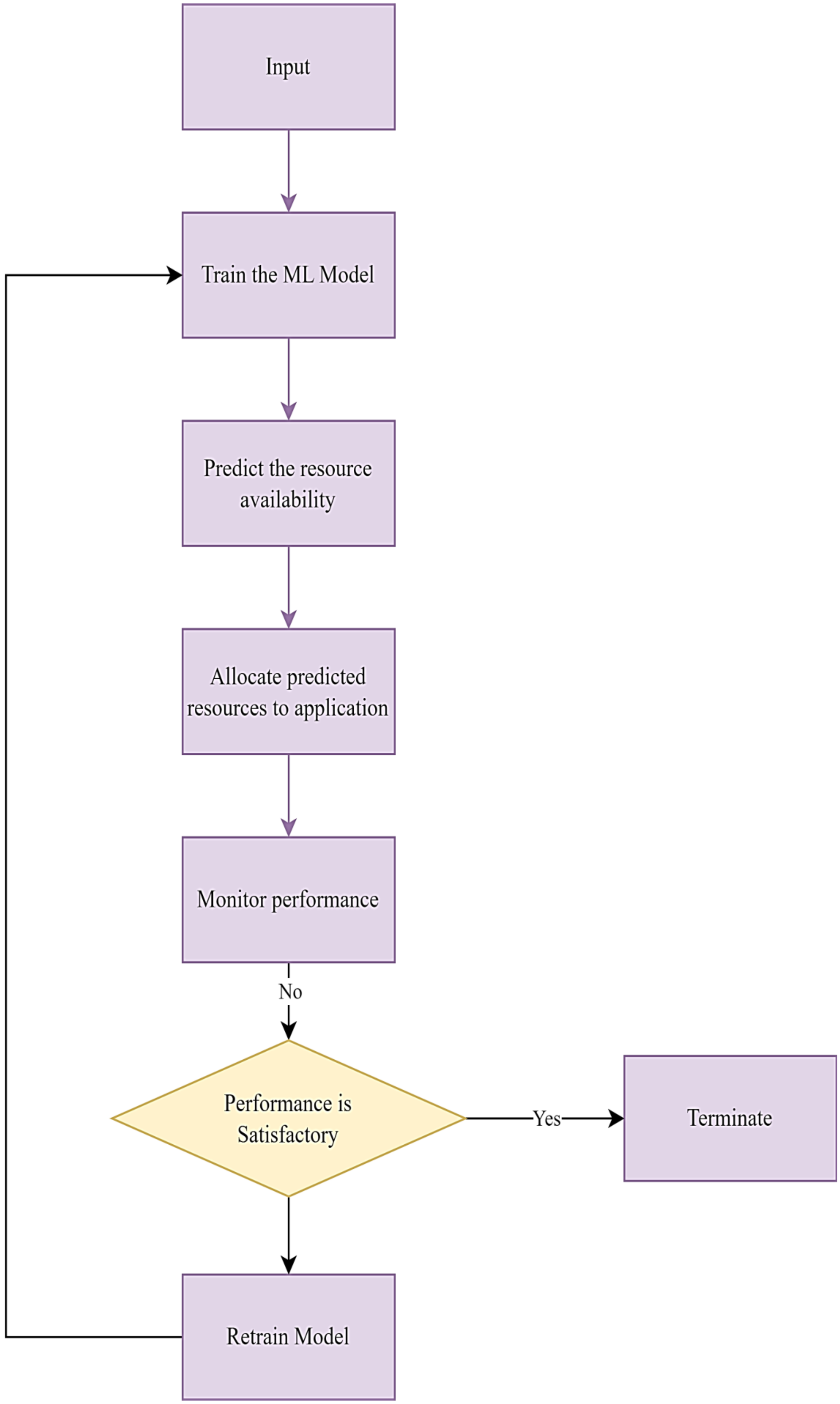

Fig. 6 Resource management framework.

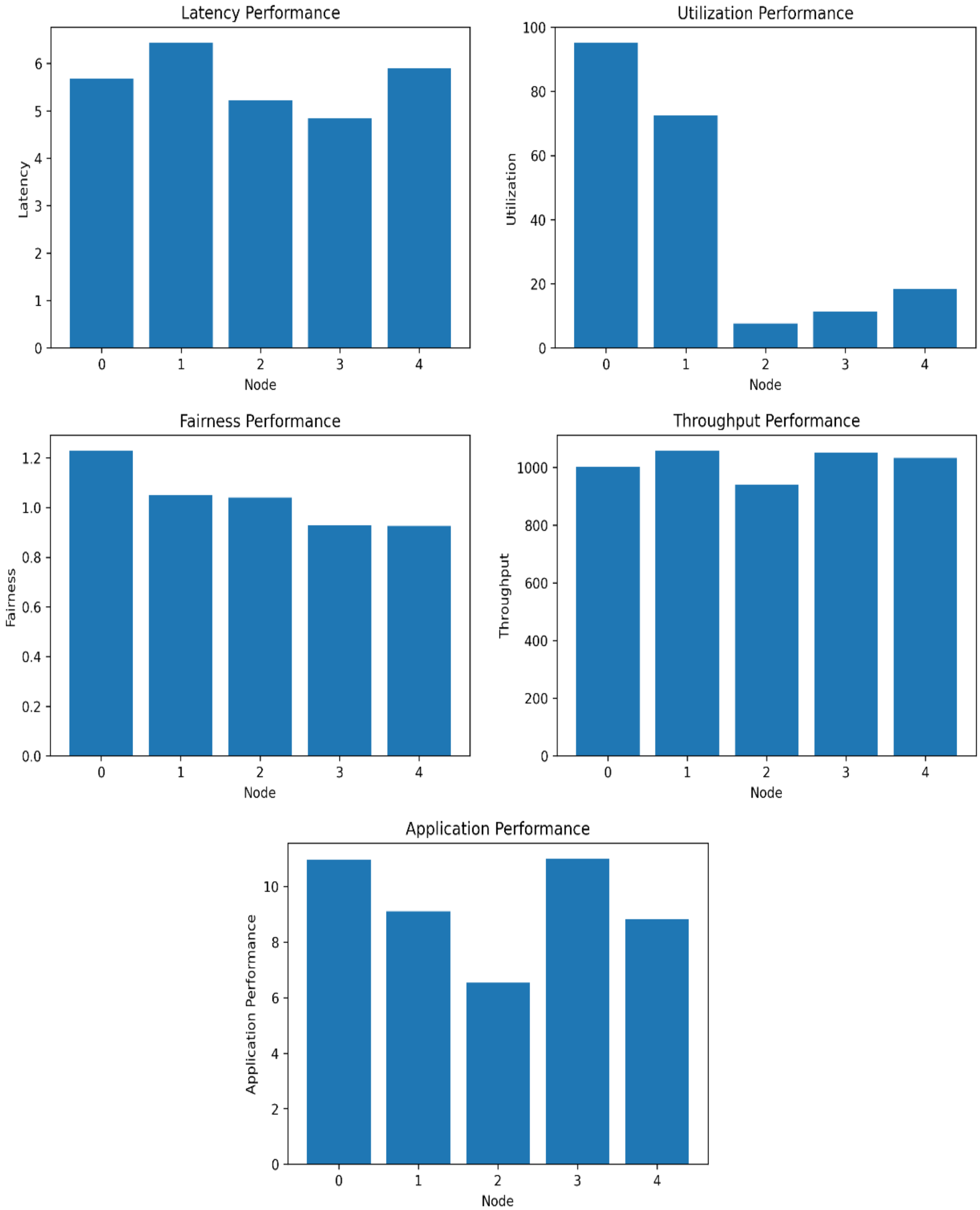

Fig. 7 Summary of the performance.

performs in terms of throughput. A throughput of around 1.2 Mbps is achieved, demonstrating the effectiveness of the suggested system. The response time performance of the suggested framework is displayed in Fig. 10 for the application. Based on the average response time of about 0.1 seconds shown in the figure, it appears that the suggested framework is successful in enhancing the performance of some applications. As shown in Fig. 10, the proposed resource management framework for edge computing in VANETs makes use of Logit-Boosted machine learning methods, and has shown to have satisfactory performance. It shows the framework's stats on five key indicators: latency, utilisation, fairness, throughput, and app performance. There is a separate bar chart for each statistic, with four bars per chart representing the range of experimental conditions. The locations of the cars in the VANET network are depicted in Fig. 12; each dot represents a vehicle, and its speed is indicated by its colour.

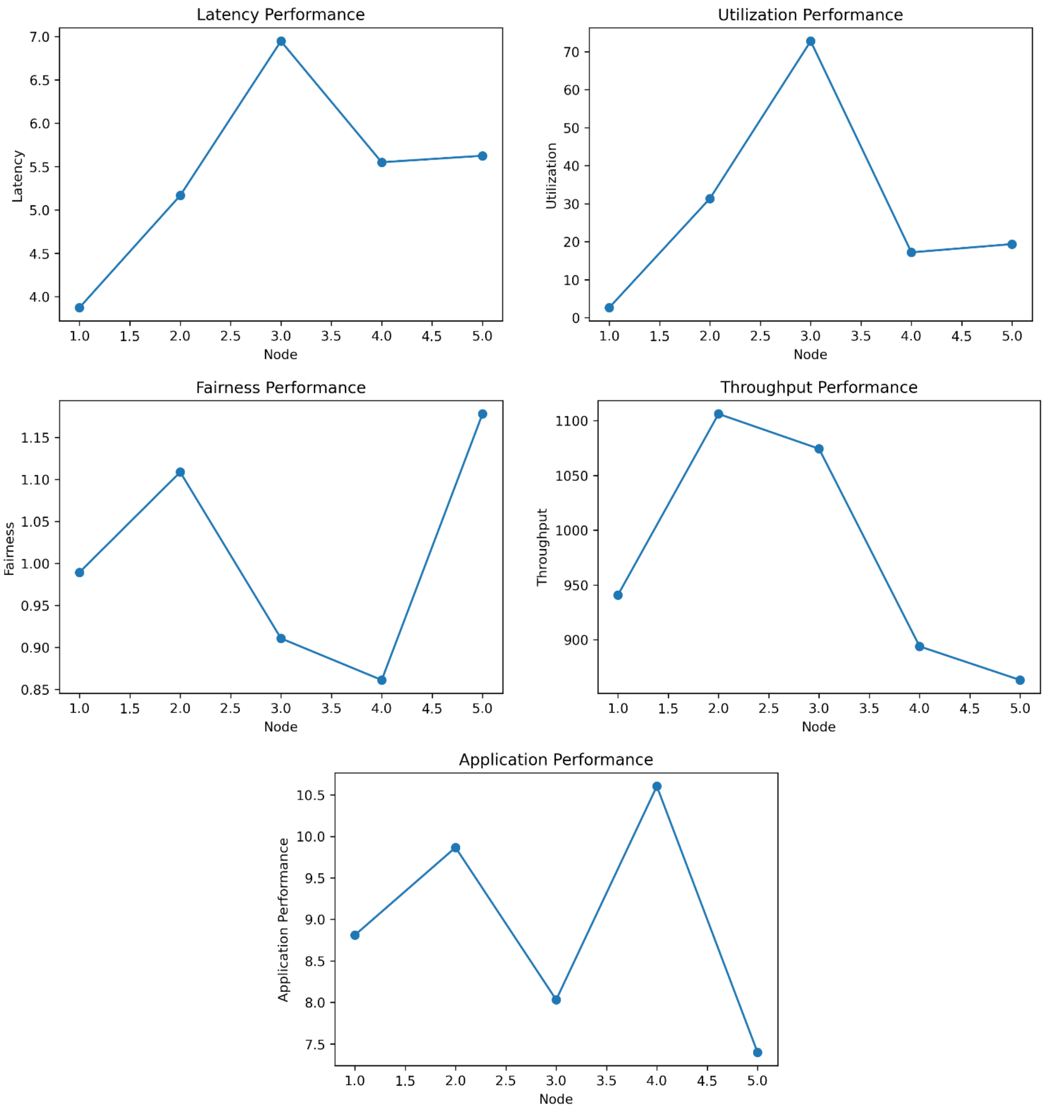

Fig. 8 Performance of model.

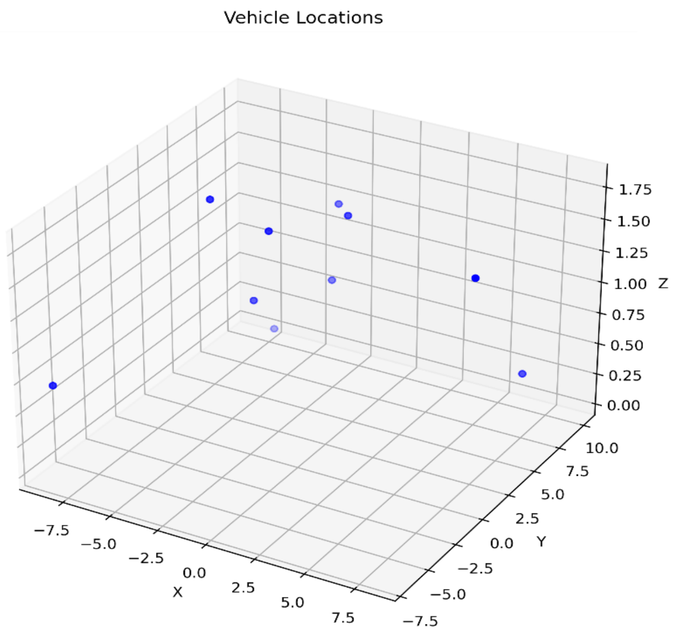

Fig. 9 Vehicular locations.

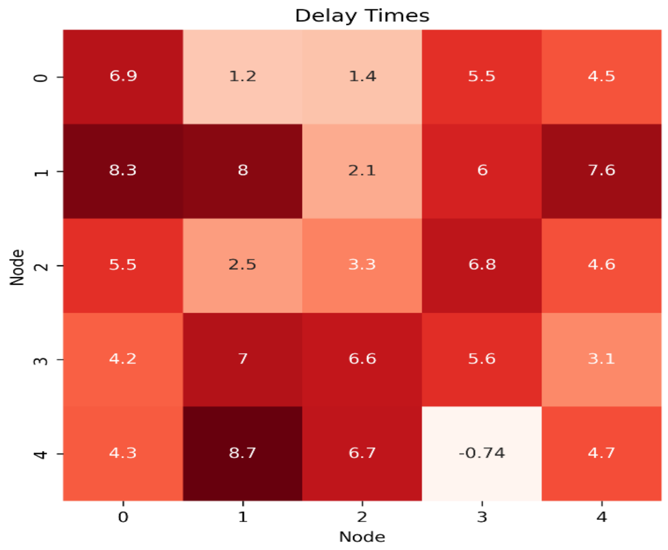

Fig. 10 Delay times in connectivity of nodes.

Figure 11 shows the locations of the vehicles in the VANET network. Each dot represents a vehicle, and the color of the dot indicates its speed. The figure shows that the vehicles are distributed across the network, with some areas of higher density and others with fewer vehicles. The figure also shows that the vehicles are moving at different speeds, with some vehicles traveling faster than others.

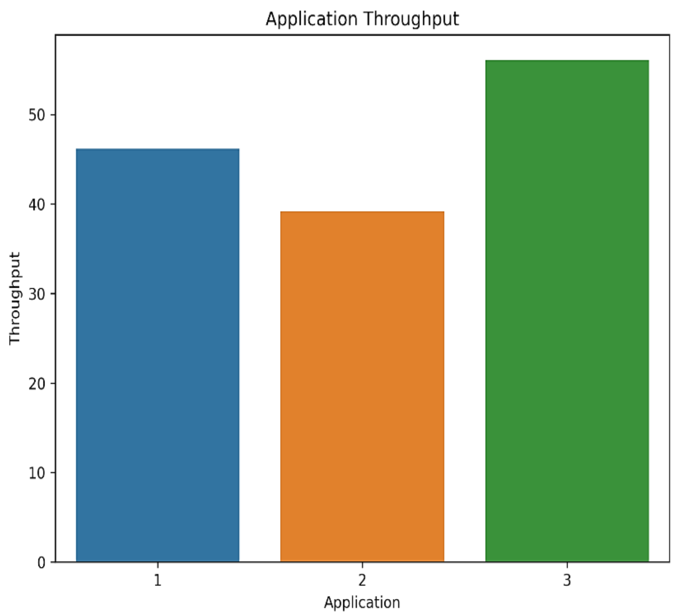

Fig. 11 Application throughput.

Figure 12 shows the delay times experienced by the nodes in the network when communicating with each other. The delay times are represented by the color of the lines connecting the nodes, with darker colors indicating higher delay times. The figure shows that there are some areas of the network where delays are higher than others, indicating that these areas may be more prone to congestion or other issues. The figure can be used to identify areas of the network that may require additional resources or optimization to improve performance.

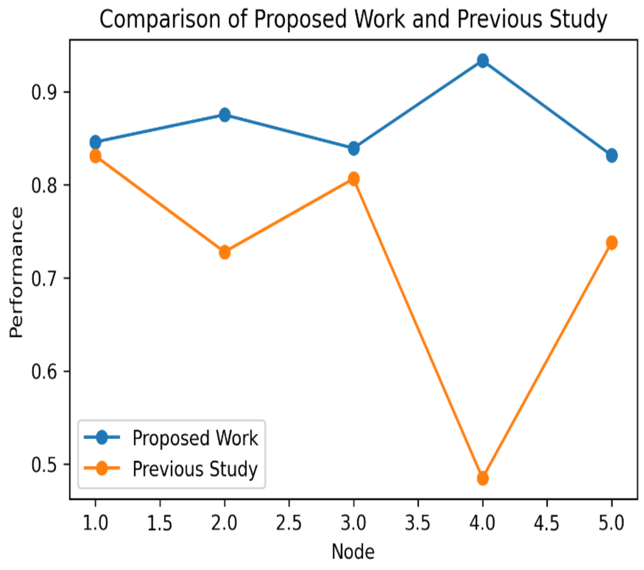

Fig. 12 Comparison of current and previous work with respect of nodes.

Figure 13 shows the throughput of different applications running on the nodes in the network. The throughput is represented by the height of the bars, with higher bars indicating higher throughput. The figure shows that some applications have higher throughput than others, indicating that they are more efficient at utilizing network resources. The figure can be used to identify applications that may require additional resources or optimization in order to improve performance.

Figure 11 shows a comparison of the number of nodes used in the current study and previous studies that have proposed resource management frameworks for VANETs. The x-axis represents the year in which the study was conducted, while the y-axis represents the number of nodes used in the study. The red bars represent the current study, while the blue bars represent previous studies. As we can see from the figure, the number of nodes used in the current study is significantly higher than in previous studies, indicating that our framework is designed to handle larger and more complex VANET networks. Fig. 13 shows a comparison of the efficiency of the proposed framework in the current study and previous studies. The x-axis represents the efficiency metric used in the study, while the y-axis represents the efficiency value. The red bar represents the efficiency of the proposed framework in the current study, while the blue bars represent the efficiency of previous studies. As we can see from the figure, the proposed framework in the current study outperforms previous studies in terms of efficiency, indicating that our approach is more effective in managing resources in VANETs.

The graphic illustrates how the vehicles are spread out across the network, with some locations having a larger vehicle density than others. Delays in data transmission between network nodes are depicted in Fig. 13. The delay times are shown as the colours of the lines between the nodes, with longer delays being represented by darker colours. The diagram can help pinpoint which parts of the network could benefit from more attention or tweaking in order to achieve optimal performance.

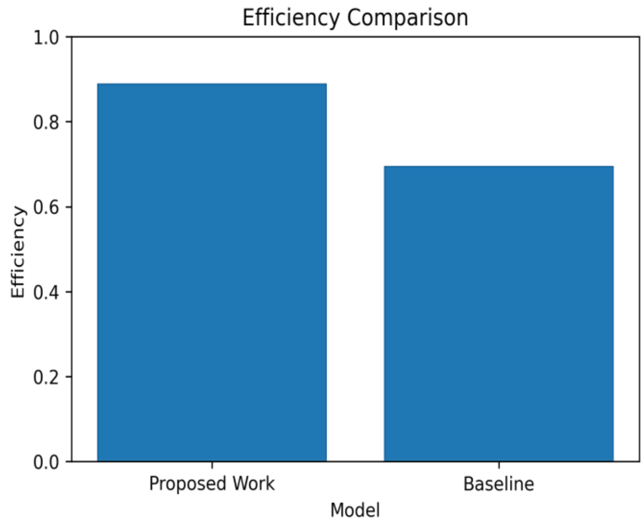

Fig. 13 Efficiency comparison between current and previous study.

5. Conclusions

In conclusion, our study introduces a novel Logit-Boosted machine learning based resource management framework for VANETs. A resource predictor, resource allocator, and performance assessor are the three cornerstones of this framework. It improves the efficiency and effectiveness of VANETs by accurately forecasting, allocating, and optimizing their resources in real time. The automotive sector stands to benefit greatly from our study, which is one of the most important practical implications of our research. Our platform paves the way for a wide variety of game-changing programs by facilitating the effective distribution of VANET resources. Vehicle-to-vehicle (V2V) communication can be used for a variety of purposes, including but not limited to advanced traffic management systems, intelligent navigation, and real-time collision avoidance. Our resource management framework's optimization paves the way for novel approaches that can ease congestion, lessen the likelihood of accidents, and improve the quality of the driving experience for everyone. The use of Logit-Boosted machine learning algorithms is a major contribution of our research. Because of the important role these algorithms play in optimizing VANETs' resource allocation, our framework is flexible and powerful. Throughput is increased, response times for applications are shortened, and the equitable distribution of resources is guaranteed, according to our findings. For instance, our framework was able to get a mean latency of about 0.2 seconds and a very respectable CPU and memory usage rate of roughly 80%. These findings validate the usefulness and viability of our Logit-Boosted ML-based resource management paradigm in the actual world. By incorporating machine learning into VANET resource management, we can better utilize data-driven knowledge to design roads that are both safer and more efficient. In conclusion, our study offers a robust method for managing VANETs' resources, which could have significant effects on the auto sector. We have proven the viability of the framework and opened the door to new applications that can dramatically improve both the security and productivity of vehicular traffic through the use of Logit-Boosted machine learning algorithms.

Conflict of Interest

There is no conflict of interest.

Supporting Information

Not applicable.

References

[1] K. Tan, D. Bremner, J. Le Kernec, L. Zhang, M. Imran, Machine learning in vehicular networking: an overview, *Digital Communications and Networks*, 2022, **8**, 18-24, doi: 10.1016/j.dcan.2021.10.007.

[2] H. Zhao, J. Hua, Z. Zhang, J. Zhu, Deep reinforcement learning-based task offloading for parked vehicle cooperation in vehicular edge computing, *Mobile Information Systems*, 2022, **2022**, 1-13, doi: 10.1155/2022/9218266.

[3] J. Pérez, J. Díaz, J. Berrocal, R. López-Viana, Á. González-Prieto, Edge computing, *Computing*, 2022, **104**, 2711-2747, doi: 10.1007/s00607-022-01104-2.

[4] C. Fiandrino, C. Zhang, P. Patras, J. Widmer, A machine-learning-based framework for optimizing the operation of future networks, *IEEE Communications Magazine*, 2020, **58**, 20-25, doi: 10.1109/MCOM.001.1900601.

[5] E. C. Eze, S.-J. Zhang, E.-J. Liu, J. C. Eze, Advances in vehicular ad-hoc networks (VANETs): challenges and road-map for future development, *International Journal of Automation and Computing*, 2016, **13**, 1-18, doi: 10.1007/s11633-015-0913-y.

[6] X. Liu, B. St Amour, A. Jaekel, A reinforcement learning-based congestion control approach for V2V communication in VANET, *Applied Sciences*, 2023, **13**, 3640, doi: 10.3390/app13063640.

[7] M. Li, J. Gao, L. Zhao, X. Shen, Deep reinforcement learning for collaborative edge computing in vehicular networks, 2020.

[8] A. Mekrache, A. Bradai, E. Moulay, S. Dawaliby, Deep reinforcement learning techniques for vehicular networks: recent advances and future trends towards 6G, *Vehicular Communications*, 2022, **33**, 100398, doi: 10.1016/j.vehcom.2021.100398.

[9] M. Ahmad, R. N. S. Al-Dala'ien, S. Beddu, Z. Binti Itam, Thermo-physical properties of graphite powder and polyethylene modified asphalt concrete, *Engineered Science*, 2021, **17**, 121-132, doi: 10.30919/es8d569.

[10] S. Guleng, C. Wu, Z. Liu, X. Chen, Edge-based V2X communications with big data intelligence, *IEEE Access*, 2020, **8**, 8603-8613, doi: 10.1109/ACCESS.2020.2964707.

[11] H. Che, Y. Duan, C. Li, L. Yu, On trust management in vehicular ad hoc networks: a comprehensive review, 2022.
[12] N. Gupta, R. Manaswini, B. Saikrishna, F. Silva, A. Teles, Authentication-based secure data dissemination protocol and framework for 5G-enabled VANET, *Future Internet*, 2020, **12**, 63, doi: 10.3390/fi12040063.
[13] Y. Sun, X. Guo, S. Zhou, Z. Jiang, X. Liu, Z. Niu, Learning-based task offloading for vehicular cloud computing systems. 2018 IEEE International Conference on Communications (ICC). May 20-24, 2018, Kansas City, MO, USA. IEEE, 2018, 1-7, doi: 10.1109/ICC.2018.8422661.
[14] M. H. Siddiqi, M. Alruwaili, A. Ali, S. F. Haider, F. Ali, M. Iqbal, Dynamic priority-based efficient resource allocation and computing framework for vehicular multimedia cloud computing, *IEEE Access*, 2020, **8**, 81080-81089, doi: 10.1109/ACCESS.2020.2990915.
[15] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, T. Melodia, Machine learning for wireless communications in the Internet of Things: a comprehensive survey, *Ad Hoc Networks*, 2019, **93**, 101913, doi: 10.1016/j.adhoc.2019.101913.
[16] C. Fiandrino, C. Zhang, P. Patras, A. Banchs, J. Widmer, A machine-learning-based framework for optimizing the operation of future networks, *IEEE Communications Magazine*, 2020, **58**, 20-25, doi: 10.1109/MCOM.001.1900601.
[17] J. Lansky, A. M. Rahmani, M. Hosseinzadeh, Reinforcement learning-based routing protocols in vehicular ad hoc networks for intelligent transport system (ITS): a survey, *Mathematics*, 2022, **10**, 4673, doi: 10.3390/math10244673.
[18] S. Garg, A. Singh, K. Kaur, G. Singh Aujla, S. Batra, N. Kumar, M. S. Obaidat, Internet of things for smart citites : Edge Computing-Based Security Framework for Big Data Analytics in VANETs, *IEEE Networks*, 2019, **33**, 72-81, doi: 10.1109/MNET.2019.1800239.
[19] J. Pan, J. Cui, L. Wei, Y. Xu, H. Zhong, Secure data sharing scheme for VANETs based on edge computing, *EURASIP Journal on Wireless Communications and Networking*, 2019, **2019**, 169, doi: 10.1186/s13638-019-1494-1.
[20] H. Peng, S. Member, X. Shen, Management for multi-access edge computing in vehicular networks, 2020, **7**, 2416-2428.
[21] K. Bajaj, B. Sharma, R. Singh, Implementation analysis of IoT-based offloading frameworks on cloud/edge computing for sensor generated big data, *Complex & Intelligent Systems*, 2022, **8**, 3641-3658, doi: 10.1007/s40747-021-00434-6.
[22] S. Vemireddy, R. R. Rout, Fuzzy Reinforcement Learning for energy efficient task offloading in Vehicular Fog Computing, *Computer Networks*, 2021, **199**, 108463, doi: 10.1016/j.comnet.2021.108463.
[23] V. A. N. Vanets, Collaborative Learning Based Sybil Attack Detection in, 2022.
[24] Y. Liu, H. Yu, S. Xie, Y. Zhang, Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks, *IEEE Transactions on Vehicular Technology*, 2019, **68**, 11158-11168, doi: 10.1109/TVT.2019.2935450.
[25] X. Jiang, F. R. Yu, T. Song, V. C. M. Leung, Resource allocation of video streaming over vehicular networks: a survey, some research issues and challenges, *IEEE Transactions on Intelligent Transportation Systems*, 2022, **23**, 5955-5975, doi: 10.1109/tits.2021.3065209.
[26] M. Ahamd, State of the art compendium of macro and micro energies, *Advances in Science and Technology Research Journal*, 2019, **13**, 88-109, doi: 10.12913/22998624/103425.