Journal of Computer Science

Original Research Paper

Route Optimization using Hybrid GRU Learning Model for SDN and Edge-Based VANET Topology

[1]Savitha Kumar and [2]Chandrasekar Chinnasamy

[1]*Department of Computer Science, Government Arts College, Udumalpet, India*
[2]*Department of Computer Science, Government Arts and Science College for Women, Coimbatore, India*

Article history
Received: 03-05-2022
Revised: 20-06-2022
Accepted: 19-07-2022

Corresponding Author:
Savitha Kumar
Department of Computer Science, Government Arts College, Udumalpet, India
Email: ymsv85savi@gmail.com

Abstract: Intelligent Transportation System (ITS) offers outstanding features, including security applications and emergency alerts. Unfortunately, ITS limits traffic control services, adaptability, and adjustability due to the traffic volume. Hence, expanding the standard Vehicular Ad hoc Networks (VANET) framework is a requirement. As a result, in the latest days, the concept of Software-Defined Networking based Vehicular Ad hoc Networks (SDN-VANETs) has drawn considerable attention, creating VANETs smarter. The SDN-VANETs design is capable of addressing the aforementioned VANET issues. The integrated (analytically) SDN architecture is customizable and it also contains domain knowledge about the VANET architecture. Packet forwarding is a fundamental challenge in VANET wherein a router, as in the form of an RSU, determines the next hop of every signal in the pipeline to provide it to its recipient as fast as possible. Reinforcement Learning (RL) is used to develop autonomous routing protocol rules; however, the limitation of RL's depth prevents it from representing more comprehensively dynamic network conditions, restricting its true value. In this research, we present a VANET infrastructure based on SDN + EDGE with a new Deep Reinforcement Learning (DRL) route optimization framework, "Gated Recurrent Reinforcement Learning (GRRL)" neural network, whereby each router has its hybrid GRU + Feed Forward Neural Network (NN) for learning and making decisions in an entirely distributed environment. The GRRL collects routing characteristics from valuable information about huge backlog packets and previous operations, substantially approximating the weighting scheme of Q-learning. We also enable every route to connect with its immediate neighbors regularly such that a more comprehensive view of network topology may be integrated. Trial findings demonstrated that the multi-agent GRRL strategy could achieve a delicate balance between congestion awareness and the fastest routes, considerably reducing packet transmission time in general network topologies compared to its competitors.

Keywords: SDN, VANET, ML, Reinforcement Learning, GRU, Edge Computing

Introduction

The VANET is a methodology that provides moving vehicles as nodes to create a mobile network. Every vehicle within the transmission radius of 100 to 300 m is converted into a wireless router or node. So, vehicles in this range connect to establish an extensive network (Bitam *et al*., 2013). The two main elements of VANET are vehicles and roadside infrastructure. Vehicle-to-Vehicle (V2V) states communication between vehicles, while Vehicle-to-Infrastructure (V2I) states communication between Vehicles and roadside Infrastructure (V2I). Data transmission in cities is challenging due to bulk data production and big data (Siddiqa *et al*., 2016). The flexibility of Inter-Vehicle Communications (IVC) and the vehicle traffic in smart cities for increased data transmission capacity led to effective communication. At the same time, there should be prevention measures against the overuse of internet bandwidth in networks (Cheng *et al*., 2018).

Moreover, there is an expectation of a vast range of VANET-enabled applications along with a point-to-point data transfer that permits the best use of network resources and infrastructure (Zhu *et al*., 2015). Short delivery delay times or high network throughput are the requirements of these applications. VANET is, however, rarely connected to the mobile network. There is a frequent alteration of network topology and network partition because of the high mobility of vehicles. The routing decisions are made easier through SDN by placing an interface route table to identify the available routes in the network. Here, the subnet mask will show the open network path in multi-hop and then transfer the data by specifying a traffic-free path to reach the destination by a gateway. It is quite challenging that routing messages possess a short delivery delay time and low routing overhead (Daraghmi *et al*., 2013). Generally, VANET contains typical routing protocols based on topology or geography. Packet forwarding uses routing protocols based on topology (for example, Ad hoc On-Demand Distance Vector Routing (AODV) and Open Link State Routing (OLSR)) that use link information present in the network. In VANET, there can be quite frequent breakdowns that hardly discover a multi-hop route to the destination. Additionally, huge routing overhead occurs while maintaining network topology because of frequent changes (Perkins and Royer, 1999).

A clear solution to managing ITS is to conceptualize Software-Defined Networks (SDN), a comparatively new technology (Muhizi *et al*., 2017). Modern networks have already been extensively managed using SDNs. An SDN controller obtained from any manufacturer can be installed over the server equipment. Most software solutions that ease the study of SDNs and their communication with third-party networks are open-source products. The SDN-VANET has effectively exercised its dominance over others. The name justifies that it is the amalgamation of SDN and VANET. The quick mobility of vehicles for safe driving is ensured by Edge Computing (EC) and the smart transportation system to manage the intricacy of diverse and adequate vehicles, massive data flow, and recurrent topological modification. VANET produces tremendous research interest in worldwide network technologies with unexpected increases in smart vehicles in smart cities. This has advanced in safer road transportation with 5G wireless communication and well-organized administration over public roads. It greatly facilitates node mobility and drives researchers toward highly efficient data distribution over the network. In the process of optimizing Internet of Things (IoT) networks, the integration of SDN with Edge Computing (EC) has countless merits. Firstly, the global view of IoT aids network resources and infrastructures are managed by SDN. Secondly, for heterogeneous fog devices, SDN offers easy management (Vladyko *et al*., 2016). Thirdly, comprehensive knowledge about the resources and tasks is possessed by SDN.

Furthermore, SDN in vehicular networks is responsible for enhancing the routing process and optimizing network resources (Qaisar and Riaz, 2016). The completion of the function within the stipulated time is facilitated by moving task requests from the IoV sensors to the neighboring Edge device. Additionally, more complications are linked to in-vehicle mobility while sending responses to IoV sensors. With a minimum response time, it requires a unique routing strategy for sending these responses. The future SDN-based VANETs show how to build advantageous confinement towards traditional VANETs. This diminishes the burden of the present wireless system and provides various benefits to solve primary challenges in the network that are required in terms of services and applications (Truong *et al*., 2015).

To obtain knowledge, a bio-inspired Machine Learning (ML) method called RL is used to discover interactions with the environmental conditions caused by external supervision. Hence, the routing challenge in distributed networks is suitable to address in which the per-hop delivery delay is measured by each node (compatible with a router) as the reward of its activities and thus, it studies the best activities. To investigate and take direct decisions from high-dimensional raw data that is more compatible with control and decision-making tasks, DL is integrated with Deep Reinforcement Learning (DRL) enabled RL. The extraction of input environmental state features and finding their suitability is performed by DL. According to DNN’s output and surveys, RL is obliged to complete decisions in understanding the state-to-action mapping (Ku *et al*., 2014). Man-Machine games, machine vision, and other areas are testimonies of DRL's significant accomplishments in recent times. The primary motive of this research is to design SDN-VANET for customizing the network based on the present network-defined route for transmitting data over the network. This will act as a customizable network for packet forwarding through multi-hop route selection over the dynamic network. For this reason, reinforcement learning is utilized for autonomous route identification in the active route optimization framework.

Literature Review

The authors (He *et al*., 2016a) present an SDN-VANET architecture that utilizes Fog Computing (FC) services to fulfill the needs of future VANET environments like automatic overtaking, ITS, and independent driving, as these applications are sensitive to delays and aware of their location. Taking one step forward, the incorporation enabled multiple access technologies are provided by an innovative Mobile-EC-based SDN-VANET architecture proposed by the authors (He *et al*., 2016b). Various methodologies have been developed to identify wired and wireless network routes by distributing true table data. This accurate route node identification is provided by a secured Global Position System (GPS) based signal for alternate arrangement in route node selection in the network. Providing communication with low latency and more trustworthiness in the network is the primary goal of this method. The proposed architecture related to reliable data communication is executed through a case study of city traffic management. The needs specific to the application, like dormancy, trustworthiness, and data rate, are very explicit in the simulation results.

Moreover, an SDN-VANET architecture with EC services is proposed (Vaquero and Rodero-Merino, 2014) for flexible distribution of the vehicles with software updates. Perceiving global topology at SDN controllers that enables organized network management is created by the architecture using V2V beaconing information. A multi-hop graph-based method for energy-efficient routing protocol for Wireless Sensor Networks (WSN) for distributing energy utilization among clusters at a flat rate was suggested. So the network's lifespan will increase depending on the multi-hop route selection by the formation of clusters in a distributed manner. Besides, the controller runs a method based on mathematical optimization models for allocating heterogeneous functional frequencies to each vehicle to encounter the challenges of interference and hidden nodes.

The network developers and service providers propose the association between VANETs and Fifth Generation (5G) technology with growing progress in advanced technologies like 5G and automotive technologies. SDN is being used as a crucial enabler to carry out this efficient integration. The authors (Liu *et al*., 2017), for example, envision an incorporated architecture of SDN, VANETs, and 5G for offering a security-by-design approach in VANETs and also between the corresponding networking services, mobility of vehicles, and the performance of network and privacy factors. The envisioned architecture is assessed despite a series of privacy issues like depletion of resources, Denial of Service (DoS), and link-layer discovery attacks leading to either the SDN controllers or the vehicles. Moreover, discussions about various feasible techniques for identifying the source of attacks are conducted for the vindication of the authors. They were incorporating VANET and 5G network architectures based on SDN that use innovative real-time multimedia streaming and application-oriented buffer-aware streaming approaches. The network performance improvement would be measured through various metrics in SDN-VANET to increase scalability, flexibility, and flat connectivity in data communication. The ITS provides different advancements for VANETs (Arif *et al*., 2018) by identifying pipelines for a route through the routing protocol.

VANET provides high mobility nodes for data transmission with efficient distribution over the wireless network like 5G, Artificial Intelligence (AI), IoT, EC, FC, etc. Where reinforcement learning plays a challenging task in choosing a signal (data transmission path) based on true value by overwhelming available issues in the network. At the same time, by connecting handovers between successive evolved NodeBs (eNBs), the proposal targets maintaining the least communication latency and guaranteeing good Quality of Service (QoS). For SDNs information gathering is related to user mobility and player buffer status to accomplish sufficient QoS and for multimedia streaming, the network signal's intensity is used to present an effective transmission strategy.

The routing of packets in a general network topology occurs using the first multi-agent Q-Learning approach proposed by the authors (Azizian *et al*., 2017). Compared to the shortest-path method benchmark, this direct routing policy attains a significantly lower mean delivery delay. The dual RL-based Q-routing model developed by Shirmarz and Ghaffari (2020) enhances routing convergence; in wireless networks, the joint Q-routing and power control policy implemented by Arif *et al*. (2020) findings in delay-sensitive applications. Several other RL-based routing techniques are evaluated. The present routing algorithms based on RL cannot outperform the history of network traffic dynamics. They discover adequate paths before the determination of packet forwarding due to their familiarity with the "curse of dimensionality" and the small space of the state-action space of RL (Hussein *et al*., 2017). The real challenge of training RL with an ample state-action space helps prevent RL-based data transmission from being applied. The new opportunity for networking applications based on RL was previously confused due to the prohibitive training burden provided by the innovation of DRL. In two features, the network designer can impact its benefits with the integration of a Deep-Neural Network (D-NN) as a reliable Q-table estimator: (1) For framing optimum policy and the NN receives more information input by expanding the space of state-action; (2) to extract obscure features from the high-dimensional input data of the NN automatically, hence, accomplishing all the time-consuming decision making by still fine-tuning the tailored selection technique. The allocation of cloud resources, modifiable video streaming, and cellular scheduling are some of the successful applications in recent times (Hussein *et al*., 2017). As opposed to the unpredictable dynamic traffic pattern, DRL is also used to create a routing policy. However, (Lai *et al*., 2017) researchers consider a unified routing policy that performs at the flow level and requires the global topology and matrix of vehicular traffic. We have set a goal to make a quick effort to enhance fully distributed packet routing policies with the help of multi-agent DRL because of the inspiration of RL and considering the restrictions of Q-Routing.

Background

Software-Defined Network (SDN)

The simplification of network management and enabling innovation *via* network programmability are accomplished using the architecture of SDN-VANET, smart technology in recent years. So, academia and industry are focusing their attention on it. The conventional network architecture is separated into two planes, i.e., data and control. Simple devices in the data plane forward only the data packets, whereas the brain and crucial part of this architecture is denoted by the control plane's controller (s). The control plane (Arif *et al*., 2020) can be unified or disseminated. The control and data planes' decoupling in SDN-VANET is enabled by the SDN technology that offers:

a) A generalization to the primary networking infrastructure of VANET applications
b) A sensible unification of networking intelligence and situation

The merging of SDN and VANET to address many existing tasks of VANET is viewed as a significant direction. The VANETs' increasing demands for features like exalted mobility, low communication latency, excessive throughput, scalability, heterogeneity, etc., can be fulfilled because of the SDN attributes. The services provided by data centers, cloud computing, and access networks have improved mainly by the SDN paradigm. Because of the high mobility in transport systems and the floating nature of network traffic, connection and network disconnection of links occur between nodes with a high frequency, leading to an ever-changing topology with network partitioning. This results in avoiding traffic frequently over alternates and heavy traffic, respectively, leading to intermittent connectivity and network congestion.

Edge Computing

To reduce latency and bandwidth, EC brings computing to the nearest possible data source. Simply stated, EC is the method of changing reduced processes in the cloud and moving them to designated points such as the user's device, an IoT device, or an edge server. The amount of long-distance interaction, which is supposed to occur between a client and a server, is reduced by carrying computation to the network's edge.

Personal computing has been the leading computing model for some time. On a user's device, applications were made to run and data was stored locally or sometimes inside an on-premise data center. In locally based on-premise computing, cloud computing provides numerous merits. Any IoT device can access cloud services integrated with a vendor-managed "cloud" or accumulation of data centers. But because of the distance between users and the data centers that host cloud services. EC reduces data travel distance by bringing computing closer to end users while maintaining the centralized nature of cloud computing.

Edge in VANET

The evolving MEC without-mode vehicular networks are comprised of VANET-Edge Computing (VEC). VEC's primary aim is to enhance communication, computing, and catch resources in the neighborhoods of vehicular users. VEC possesses the potential to play a critical role in addressing the increasing requirements of edge devices for low latency and high bandwidth. The notable traits of VEC, such as fast mobility of vehicles, differ from conventional MEC resulting in recurrent and more active topology changes and complex communication traits because of the quickly changing channel environment over time. Vehicles possess VEC communication, computation, and storage resources; edge servers frequently acted upon by RSUs are fixed in nearby vehicles to collect, process, and conveniently store data.

Because of the limited capacity, the computation-intensive and latency-sensitive tasks to the edge servers of the vehicles can be offloaded by vehicles to minimize the response time and brilliantly lessen the enormous load on backhaul networks. Without accessing the core network, the required content can be directly acquired from catching nodes in a VEC background, minimizing total latency and optimizing the usage efficiency of network bandwidth. The high dimensionality VANET provides multi-hop network communication through a truth table that has been produced by a routing protocol. This pointed to high data transformation through the best packet forwarder tested in the simulation testbed and performance was also measured. Each caching node cannot cache all the contents as the storage space is limited. Therefore, significant problems like deciding what to cache, where to cache, and how to decide cache policy arise. Integrating new developing technologies like SDN, Artificial Intelligence, and blockchain is predicted to enable VEC services efficiently and effectively.

Proposed EDGE + SDN + VANET Architecture

This section describes the proposed architecture, responsibility, and functional mode. As shown in Fig. 1, VEC architecture's foundation is laid on three layers: Edge cloud, cloud, and smart vehicular layer.

A list of vehicles, wireless switches, Base Station (BS), SDN controllers, and Road-Side Unit SDN Controllers (RSUCs) are contained in the network. With the integration of a centralized SDN controller at its peak level, the network control plane owns a hierarchical structure. The RSUs, RSUCs, and BSs at the lower level can provide fog services. The network infrastructure layer includes vehicles and wireless routers. The network topology is an electronic map-based graph at the infrastructure layer, with junctions and roads serving as the graph's vertexes and edges. Each connection with a wireless router takes responsibility for sending data packets between vehicles in the preset trajectory. The planned switches are part of the Smart City infrastructure and offer various services to organizations and end-users, thus reducing the cost of implementation for ITS. GPS determines the location of vehicles if they are equipped with GPS technology. The data is regularly sent to the fog nodes by vehicles like position, velocity, source and destination address, and direction of movement. The proposed routing method's network model, apart from setting up the components and their relationships.

The following sections describe the function of each network component model in the proposed routing method.

SDN Controller

The controller performs tasks like routing, data traffic management, and intellectual network. It receives the routing information from disseminated network fog nodes. The SDN controller is in the process of collecting data and establishing a general overview of the network's identification. SDN and MANETs provide a good solution for wired and wireless networks to identify a route through a routing protocol in a dynamic environment. The information regarding data transmission in the route node will be carried out from source to destination through SDN. The network performance in the VANET framework can be managed by the controller accordingly. It also satisfies the task of arranging fog nodes. According to the instructions and guidelines framed by the controller, the plan is executed. The controller frames the policies like those associated with computing resources, virtual memory, and security codes in coincidence with fog nodes.

Road-Side Unit SDN Controller (RSU-SDNC)

They are nothing but the incorporation of SDN controller capabilities into fog nodes controlled by the SDN central controller. Depending on the information gotten from the fog nodes, RSUCs are responsible for accomplishing routing within the coverage area. RSUCs also arrange fog nodes within their coverage. Through wireless communication and the OpenFlow process, the RSUCs interact with switches positioned at junctions.

Edge Nodes (RSU/BS)

The RSUs and BSs are featured with abilities like data collection and local computing. Their's behavior is as fog nodes controlled by the roadside unit SDN controller. Correspondingly, fog nodes transmit traffic information for general computing and long-term storage.

Network Switches

The transmission of data packets between vehicular communications within the predetermined path of the controller is the responsibility of wireless switches installed at junctions. RSUCs configure the wireless switches of the network through wireless communication and open flow.

Vehicles

To transmit data packets in Inter-Vehicle Communication (IVC), vehicles should be equipped with an Onboard Unit (OBU). The data plane components include vehicles and wireless switches.

Edge Cloud Layer

This layer ensures that the smart vehicular and the cloud layer communicate. They were achieved using the communication protocols like 802.11 p, 3rd Generation Partnership Project (3GPP), Third Generation (3G), Fourth Generation (4G), Long-Term Evolution (LTE), and 5G by the vehicles equipped with devices.

Routing in VANET

Data forwarding from source to destination through multi-hop steps is known as routing. Significantly to determine the ways packet transmission to its destination, acclimatize with the path at times of failure, record the connectivity data, and much more. The features of a suitable routing protocol are that it delivers a packet within a short time and uses minimal bandwidth. The routing protocols deployed in VANETs are different from Mobile Ad Hoc Networks (MANETs) and the VANET environment must handle the below challenges.

Extremely Dynamic Topology

The formation and sustainability of VANETs are unplanned, such that, at any time, the vehicles may join and leave the network and it goes to the extent of sustaining in just for a few seconds.

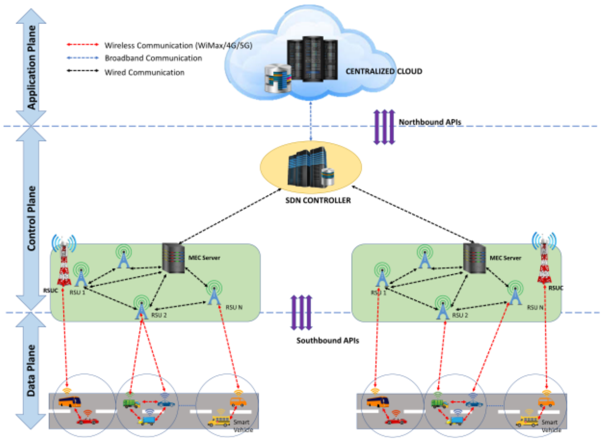

Fig. 1: Proposed SDN+EDGE VEC architecture

Network Divisions

Due to the sparse traffic in rural areas, the networks break up, forming partitions.

Time Sensitive Transmissions

The transmission of safety cautions must be rapid because users must be prioritized over usual data.

The routing protocols of traditional MANET don't consider the features of VANET, so applying these protocols straight away in the VANET environment is ineffective. VANETs are genuinely needed for the cloud servers with high computational and storage capacities for ITS because major challenges are available in FC. The primary challenges of over-optimizing network nodes are solved through fog nodes in urban regions by generating routes. The cloud servers store the network's performance and storage abilities of the present ITS system through the internet. Moreover, the cloud doesn't provide a fully satisfying network for extended latency accessibility of centralized networks. For this reason, the SDN-VANET is considered more effective with fewer constraints towards route node selection over the network**.** So, to make the routing protocols in VANET very efficient, it is better to alter or create new MANET routing protocols, particularly for VANET.

Materials and Methods

As mentioned previously, Deep Learning (DL) systems perform periodic processes such as data aggregation, comparative analysis, classification, responses, and self-tuning to validate data. So, enhancing the DL system's output accuracy requires widespread training data and adequate training time. Despite the application's dynamic forecasting skills, this is the reason for the current restrictions on the applications of DL systems. Our proposed paper embeds a dedicated DL system in RSUs to make their functionality very developed and efficient. The scenario briefed in this study can quickly meet the needs of the DL system.

System Model

As shown in Fig. 2, an SDN + EDGE that contains three layers is considered. The data layer, also known as the infrastructure layer, is installed with hardware-like sensors. The data layer's key focus is to perform data transmission between network cluster centers. A logically centralized controller provides connectivity between an application and the control layer's data layer. It also has functionalities like dynamically updating routing protocols and software development system resources. As such South-Bound Interfaces (SBIs) communicate between the control and data layers, whereas North-Bound Interfaces (NBIs) communicate between the control and application layers. SDN's upper layer is the application layer. Servers, data centers, storage, and applications are all key components. This layer is responsible for developing the conceptual model of software service providers. Network architecture information is collected through the data layer and used by several network applications such as network monitoring, data protection, and routing.

The routing application determines the routes for source-destination pair requests. Depending on the existing state of the network, the path is discovered for each source and destination pair request because the routing application collects network statistics regularly. Within the routing application, the proposed Deep Q-Routing (DQR) method is executed. The network's existing state is input into the routing application and gives a path as the output.

A network is considered to connect routers and links between them. Packet forwarding sends data packets from a source node to a destination node by discovering a route. The numerical analysis for the packet routing problem is addressed.

Network

The network modeling as a directed graph is $g = (\mathcal{N}, \mathcal{E})$ where the finite sets of nodes and transmission links between them are $\mathcal{N}$ and $\mathcal{E}$, respectively. The origination of each packet is s *src* node and *src*, $dest \in \mathcal{N}$ and $src \neq d$ are the destinations for nodes with arbitrarily generated intervals.

Routing

Packet routing helps transport each packet to its destination using a network of routers. The routers' queue adopts the First-In-First-Out (FIFO) criterion. The HOL is delivered continuously by each router to its neighbors' nodes until the packet moves into its termination.

Target

The packet routing problem uses a few routing standards to determine the best path between a source and a destination node. It is described in our test as the average time required for packets to be delivered. For each packet *ht*: $pht \in pset$, *'pset* is properly denoted as the packet set and ET_p as the entire transmission time. The average delivery time $AT = \sum_{p \in p} ET_p / \mathbb{P}$ is aimed to be reduced, where, $\mathbb{P}$ denotes the number of packets in *the pset*.

Reinforcement Learning Formulation

The link selection model is modeled as a model-free RL problem in this subsection. Each RSU is created by a DRL agent that measures metadata as well as input from source nodes. Each node acts as an autonomous agent, monitoring the local network state and communicating with other nodes to verify its routing policy. The state-space (S), the action space (A), and our method of designing composite benefits are mentioned.

State Space

The state of an agent at time *t*, $\mathbb{S}_{at} \in \mathbb{S}$, is given by $\mathbb{S}_{at}$: $\{dest_p, \chi_a, M_a\}$, where the existing packet's destination is $dest_p$, the additional information received for agent a is χ_a and M_a is the information shared from neighboring nodes of a. Moreover, the possible links between each node in the form of cluster vectors $\left(C_{v_1}\left[limit_m...L\right]C_{v_2}\left[limit_m...L\right]\right)$ are also included in the state that denotes the network topology, the parameter $limit_m \geq 1$ can be selected depending on the memory restriction of the routing nodes. The state includes the assessment, queuing delay, and transmission delay of the set of possible links $\varepsilon_{C_{v_1}[L],C_{v_2}[L]}$. This way $\mathbb{S}_{a_t} = \left(C_{v_1}\left[limi_m....L\right], C_{v_2}\left[limi_m...L\right], Q_a(t), \{V_l(t), D_l(t)\} l \in \varepsilon_{C_{v_1}[L]C_{v_2}[L]}\right)$ gives the state of the agent at a given time *'t'*.

Action Space

$\mathbb{A}_{a_t} : \mathcal{F}_a, \mathbb{A}_{a_t} \in \mathcal{A}$ is the representation of the action space of agent a. Therefore, the action space's size equals the number of adjacent nodes for each agent. Moreover, using $l \in \varepsilon_{C_{v_1}[L],C_{v_2}[L],}$ the agent can choose a link for routing requests in the form of link representation.

Reward

A reward $\mathbb{R}_{a_t} \in \mathcal{R}$ is received by the agent at the state $\mathbb{S}_t$ that takes an action $\mathbb{A}_a \in \mathcal{A}$. Both the global and regional aspects of the path calculation problem are obtained by the composite reward as follows:

Global Reward

Solely based on the available regional information, the group leader takes action, not knowing its consequences on the End-to-End QoS-Type restrictions. We design a global reward control signal distributed to all cluster heads involved in routing a single homogenous flow to address this issue. Especially a control signal $\vartheta_{src,\ dest}$ is sent to the other RSU agents by a source node *src* that distributes a route request to a destination node *dest*. The routing between 0 and 1, denoting the source satisfaction with the chosen path, is performed by the RSU agents. Since the source does not constantly send the global reward signal, it is noted that the signal may not be instantaneous. After selecting the path, the source transmits it from time to time and the $\mathcal{H}$ hops are supposed to be received by the agent. It means that the link $l \in \varepsilon_{C_{v_1}[L],C_{v_2}[L],}$ selected by the group leader's global reward in the state $\mathbb{S}_{a_t}$ is expressed as following Eq. (1):

$$\mathbb{R}^g_{t+\mathcal{H}} = \Upsilon_1 \vartheta_{src,dest} \tag{1}$$

where the weight of the global reward is $\Upsilon_1 \geq 0$.

Local Reward

Based on the independent contribution of each agent, the local reward is allocated separately. Mainly the queuing and transmission delay of the selected link's packet loss and the possibility of balancing the load by the agent over possible connections. The following Eq. (2) is the description of the local reward:

$$\begin{aligned}\mathbb{R}^\ell_{t+\mathcal{H}} &= \Upsilon_2 1\left(V_l(t) \leq V_{th}\right) - \Upsilon_3\left(Q_a(t) + \mathcal{D}_l(t) - \Upsilon_4 P_l(t)\right) \\ &+ \Upsilon_5 ex\left(-\sum\nolimits_{l \in \varepsilon_{C_{v_1}[L],C_{v_2}[L]}} V_l(t) - \bar{V}(t)\right)\end{aligned} \tag{2}$$

where the network operator chooses the weights Υ_2, Υ_3, Υ_4, Υ_5, ≥ 0, utilization threshold is denoted as V_{th} that is unwanted to go beyond, and the links' average utilization in $\varepsilon_{C_{v_1}[L],C_{v_2}[L],}$ .T is denoted as $\bar{V}(t)$. Hence, the combination of global and local rewards that gives the composite reward is expressed as Eq. (3):

$$\mathbb{R}_t = \mathbb{R}^g_t + \mathbb{R}^\ell_t \tag{3}$$

The accumulation of the maximum positive reward in each episode is the objective of the agent. The reward function's design supports this. If an agent selects an invalid action, it penalizes the agent most strongly since it results in divergence. Because the agent can choose behavior from a large time domain, where only a few activities are valid in each frame buffer, the agent must first select good behavior. Good behavior is chosen and network loops are prevented with this experience and understanding due to the high risks associated with initially set behavior. The agent enhances selected behavior by minimizing the negative reward, i.e., determining a path that significantly improves the QoS metrics:

(i) The reward function is designed when the QoS parameters are optimized to encourage the agent to reach the destination node
(ii) The reward function was developed to reward the agent for quickly reaching the destination node while optimizing the QoS metrics

Gated Recurrent Reinforcement Learning (GRRL) Model

In this part, the design of the Gated Recurrent Unit NN framework for our algorithm is introduced. Each node in the distributed manner of the RL methodology is an autonomous agent with its own hybrid Gated Recurrent Unit +Feed Forward Neural Network (GRU + FFNN) for decision-making. Thus, the following is the framework for a NN optimized for a single agent:

GRU + FFNN Model

At least two layers of Artificial Neural networks comprise the GRU + Fast Forward Neural Network Model (FFNN). The hidden layer depth in GRU + FFNN increases when the same data is appropriated, which means that the capability of data fitting is improved when there is a node reduction in each layer. The merits of the GRU + FFNN are: Accomplishing correlation or feature combination finding that have never appeared, combining hidden attributes; minimizing the difficulty of feature engineering; and enhancing the model's standardization capability.

The GRU + FFNN comprises an input and multiple layers with GRU, hidden and output layers, as illustrated in Fig. 3.

The nodes are entirely linked. According to the data scale, the number of layers can be adjusted. The selection of corresponding hidden node and output layer activation functions are flexible. The fusion of algorithms is the essence of the model. The GRU + FFNN's mathematical structure is as follows:

(i) The number of input layer nodes and features of the input data is equal. More features are required to minimize the impact of underfitting or overfitting if there are more hidden layers
(ii) Neurons enable the composition of the hidden layer node. Both Tanh activation at the GRU layer and Swish activation at the Hidden layer is comprised of the neurons and the Swish activation function is the activation function in the default neuron. Both Swish and Rectified Linear Unit (ReLU) are computationally efficient and the former performs better than ReLU on deeper models. The range of swish values starts from negative infinity to infinity. The definition of function is as follows: Eq. (4) and (5):

$$f(x) = x * sigmoid(x) \tag{4}$$

$$f(x) = \frac{x}{(1-e^{-x})} \tag{5}$$

There is a single node in the output layer. The resultant product of the hidden layer and the weight to a bias on the output node is done to get the prediction value. The following function encapsulates the process, where *'i'* signifies the number of nodes in the preceding layer and *'c'* indicates the bias, Eq. (6):

$$f(x;W,c) = \sum_i \left(W_i^T X_i + c\right) \tag{6}$$

The GFFNNM's full function expression is a multi-level nested form. This means that the previous layer's output is the next layer's input, the input feature in the function is *'x'*; the weight of the layer is *'w'* and the node biases are *'c'* and *'b'*, EQU (7):

$$f(x;W,c,w,b) = W^T \sum_i \left(W_i^T X_i + c\right) + b \tag{7}$$

The Adam optimizer was the common optimization function. One of the most selected optimizers is the Adam optimizer. The Adam optimizer principle aggregates the flow theory from "SGD with velocity" and the adaptive learning ratio from "Ada Delta." The implementation is direct and it is computationally efficient. Sparse memory is required and is ideal for issues with noisy or sparse gradients. Adam uses exponentially moving averages computed on the gradient assessed on a current mini-batch. The aim is to store and use the gradient's estimated first and second moments to update the parameters.

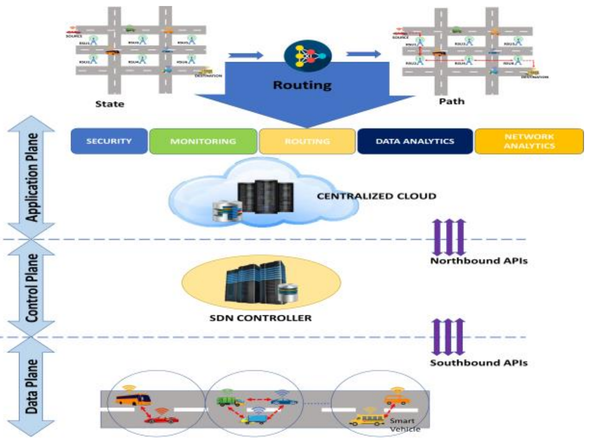

Fig. 2: Proposed SDN and EDGE-based routing model

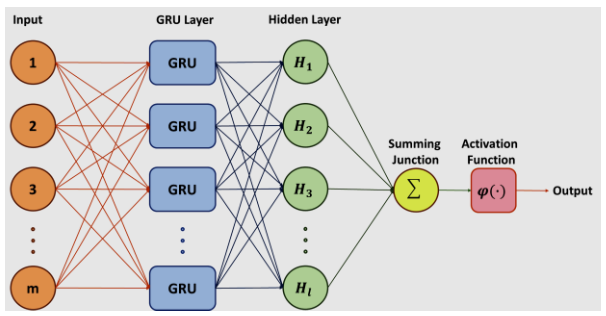

Fig. 3: GRU+FFNN mathematical structure

In the making of the state space, '*$dest_{pkt}$*', the existing node corresponds, χ_a, the action history and the future destination correspond, 'M_a', the max-queue node corresponds. All the above data will be processed with one-hot encoding before being fed into the neural network. The advantage is that the result of One-Hot-Encoding is binary instead of ordinal and everything sits in an orthogonal vector space. The demerit is that the feature space can blow up fast and face the consequence of dimensionality. Followed by Principal Component Analysis (PCA), one-hot-encoding is deployed in these cases to reduce dimensionality. Other encoding schemes rarely knock out recently discovered judicious combination of one-hot plus PCA. Thus, PCA will naturally intend to group similar features into the same feature because PCA discovers a linear overlap.

The three hidden layers denote a hidden neuron chain of four subsets. 36 neurons are possessed by each subset and are linked with the next part of the input neurons individually. Another hidden layer with 36 neurons is added following the first GRU layer. Every agent can only receive observational data in a partially visible environment connected to the complete environment. To maintain an inner state and accumulate observations over time, a GRU layer is added. A target network with weights θ_t is used for the action evaluation, an imitation of neural network weights updated every Γ step. The agent's veiled state 'v' '$\mathbb{v}'$' will be added to represent the Q-value defined as Q ($\mathbb{O}$, $\mathbb{v}$, a), despite the partial observation $\mathbb{O}$. At each step 't', an experience replay memory M of size M is deployed to store a group leader's experience. Based on the priority of these experiences, they are later replayed at a rate. The priority depends on how surprising that experience was. Moreover, the output layer's size and the agents' action spaces $\left|\mathbb{A}_{a_t}\right|$ are identical and the estimated Q-value of the corresponding action is the value of each output neuron. The parameters of NNs are updated with this change to represent the Q-value rather than the value of the Q-table.

GQ-R using GRRL with Hybrid Neural Network

The Gated Q-Routing (GQ-R) algorithm is proposed here with multi-agent DRL that uses a fusion of GRU + FFNN where both training and execution processes are performed. Algorithm 1 shows the ML algorithm's pseudo-code, in which, for each node, the initialization and the training process are the same. An individual agent is represented by each node a and Q_a is its neural network with a specific parameter θ_a for estimating $Q_a(\mathbb{O}, \mathbb{v}, \mathbb{a}; \theta_a)$ $Q_a(\mathbb{O}, \mathbb{v}, \mathbb{a}; \theta_a)$, a state-action value function represents the packet's anticipated delivery time to reach the destination while executing action $\mathbb{a}$n in-state $\mathbb{O}$ and hidden state $\mathbb{v}$. The environment transitions are retrieved for each agent by replaying the memory $\mathbb{M}_a$ with a capacity of 100 prepared individually and the network parameters are updated by sampling a random mini-batch with a size of 16 from it. When a packet *pkt* comes to the head of a line of some node a, local information l_{pkt} and O_a will be observed by an agent a and gather shared information M_a by communicating with neighbor nodes. Agent a will execute an action based on decaying ϵ-greedy policy by assimilating existing state $\mathbb{S}_t\left\{l_{pkt}, O_a, \mathcal{M}a\right\}$ and hidden state $\mathbb{v}_t$. This means that a random action will be chosen by a node a from its action space $\mathbb{A}_a$ with probability ϵ or select the action with the highest Q-value with probability 1- ϵ:

Algorithm of ML Pseudo-Code

```
Step 1. Initialize $n$, $\mathbb{N}$;
Step 2. Set n $\leftarrow$ 1 $\mathbb{N} \leftarrow$ no. of nodes; //initialize node
        variable
Step 3. Repeat
Set $\mathbb{M}_n \leftarrow \varnothing$; //initializing replay memory.
Set $Q_n \leftarrow Rand(\theta_n)$; //initializing Q networks with random weights.
Step 4. Until$n < \mathbb{N}$,
Step 5. Repeat Call Allocate($Cur_a$, pkt); // call function
        to assign the current agent with the packet to be
        transmitted.
Call Get ($l_{pkt}$, $O_a$); //observe local information $d_p$ and $E_n$.
Call Obtain ($M_a$); //Collect the shared information.
Call Join ($\mathbb{S}_t$, $\mathbb{v}_t$); //incept function to integrate the current
state with the hidden state.
$act_t \leftarrow argmax_a\ Q_a(\theta_a)$ //execute an action with
probability 1-$\in$; or a random action with probability $\in$.
Call Allocate ($nxt_t$, $pkt$); //allocate the packet to the
following agent
Compute $R \leftarrow \mathbb{r}_t$; //collect the reward for the current
action
Call Get ($\mathbb{S}_{t+1}$, $v_{t+1}$); //observe next state and hidden state
If ($nxt_a$ = = $l_{pkt}$) Then
Set $\mathbb{f}_t \leftarrow 1$;
Else
Set $\mathbb{f}_t \leftarrow 0$;
Store: $\mathbb{M}_a \leftarrow \{\mathbb{S}_t, \mathbb{r}_t, \mathbb{v}_t, nxt_t, \mathbb{S}_{t+1}, \mathbb{v}_{t+1}, \mathbb{f}_t\}$
Rand ($D_n$); //obtain random batch $\{\mathbb{S}_j, \mathbb{r}_j, \mathbb{v}_j, nxt_j, \mathbb{S}_{j+1}, \mathbb{v}_{j+1},$
$\mathbb{f}_j\}$ form $\mathbb{M}_a$
Set
$yj \leftarrow \mathbb{r}_j + max_{act'} Q_{nxtj}(\mathbb{S}_{j+1}, \mathbb{v}_{j+1}, act'; \theta_{nxtj})\ (1-\mathbb{f}_j)$; $\theta n \leftarrow$
gradientsescent $((y_j-Q_n)(\mathbb{S}_j, v_j, act_j; \theta_n))^2)$;
Until $t \leftarrow true$;
Until episode < $M$;
```

The assignment of it is given in Eq. (8):

$$a_t = \begin{cases} a\,random\,action\,with\,probability \in \\ argmax_a\, Q_a\left(\mathbb{S}_t, act, \mathbb{V}_t; \theta_a\right) with\, probability 1 - \in \end{cases} \tag{8}$$

Next, to the corresponding neighbor node '*nxt*$_a$', the existing packet '*pkt*' is forwarded, and reward '$\mathbb{r}_t$' is calculated and sent back to agent a. The transition of $\mathbb{S}_{t+1}$ and $\mathbb{v}_{t+1}$ to the existing state and hidden state is done, respectively. Additionally, if the next '*nxt*$_t$' matches the packet's destination '*dest*$_{pkt}$' or is set to '0', the transmission flag '$\mathbb{f}_t$' will be set to '1'. The following Eq. (9) is the assignment of '$\mathbb{f}_t$':

$$\mathbb{F}_t = \begin{cases} 1 nxt_t = dest_{pkt} \\ 0\ otherwise \end{cases} \tag{9}$$

The transition ($\mathbb{S}_t$, $\mathbb{r}_t$, $\mathbb{v}_t$, nxt_t, $\mathbb{S}_{t+1}$, $\mathbb{v}_{t+1}$, $\mathbb{f}_t$) will be recorded by an agent into its replay memory. '$\mathbb{M}_a$' after receiving this information. As opposed to the sequential update process of the Deep Recurrent Q Network (DRQN), the violation of the DQN random sampling policy is avoided by randomly sampling a training batch {$\mathbb{S}_j$, $\mathbb{r}_j$, $\mathbb{v}_j$, nxt_j, $\mathbb{S}_{j+1}$, $\mathbb{v}_{j+1}$, $\mathbb{f}_j$} from ' $\mathbb{M}_a$'. The remaining delivery time 't' is expected to be spent by packet '*pkt'* from nxt_t to l_{pkt} 'by recalculating before the training process. 'τ' is given by Eq. (10):

$$\tau = \max_{act'} Q_{nxtj}\left(\mathbb{S}_{j+1}, \mathbb{V}_{j+1}, act'; \theta_{nxtj}\right) \tag{10}$$

At the end of the decision time step 't', the gradient descent technique determines the neural network $Q_n(n)$. The sum of the immediate reward '$\mathbb{r}$' and the remaining delivery time is the target value 'y_j', Eq. (11):

$$y_j = r_j + \tau\left(1 - f_j\right) \tag{11}$$

By reducing the loss function '*Lt*', the parameter of $Q_n(\theta_n)$ can be trained, Eq. (12):

$$h_t = \left(y_j - Q_a\left(\mathbb{S}_j, \mathbb{V}_j, act_j; \theta_a\right)\right)^2 \tag{12}$$

'θ_a' can be updated as the following Eq. (13) after distinguishing the loss function h_t concerning the weights 'θ_a':

$$\theta_a \leftarrow \theta_a + \alpha \nabla_{\theta_a}\left(y_j - Q_a\left(\left(\mathbb{S}_j, \mathbb{V}_j, act_j; \theta_a\right)\right)^2 \tag{13}$$

where the learning rate is θ. In this same way, until convergence, each agent's network parameters are updated with episodic training.

Results and Discussion

Simulation Setup

Our simulation setup, configurations, and results are described in this section. Our work is simulated in TensorFlow *v.1.6.0* and Optimized Network Engineering Tools (OPNET) *v.14.5*. TensorFlow is an open-source machine learning library. Additionally, our experiment has two distinct types of nodes: (a) Normal nodes that reliably forward data packets, and (b) malicious nodes that randomly drop received information packets. In our experiment, the number of malicious nodes is lower than that of normal nodes. The training efficiency of path learning is addressed using normal NN with multiple hidden layers and the proposed GRRL as a Deep Q-Network. Additionally, our proposed GRRL model is tested using simulation models for PDR and average network throughput.

The Intel (R) Core (TM) i7-6600 CPU with 16 GB memory processor has been implemented with the simulation. TensorFlow 1.6.0 was associated with Python 3.6 and OPNET 14.5 on Windows 10 64-bit. With the two existing schemes, we have compared the proposed scheme in our simulation. Within the sensing range, randomly configured SDN-VANET, one RSU, and one centralized SDN controller are feasible. The source's transition probability and the destination vehicle's to 1 are set in the meantime. The evaluation network's architecture in our simulation is similar to the target Q-network. Only the analysis network can be trained using the gradient descent method and the destination Q-network is replaced with the Q-value every five steps (after 200 steps). The fundamental parameters' value is summarized in Table 1.

Our proposed hierarchical system's efficiency is proved by comparing it with.

Existing Method # 1

Presents an SDN-VANET architecture that utilizes FC services to fulfill the needs of future VANET environments like automatic overtake, ITS, and independent driving as these applications are sensitive to delay and aware of the location.

Existing Method # 2

The incorporation of multiple access technologies is provided by an innovative Mobile EC-based SDN-VANET architecture proposed by the authors. Providing communication with low latency and more trustworthiness in the network is the primary goal of this method. The proposed architecture related to reliable data communication is executed through a case study of city traffic management.

But neither of the models incorporates ML techniques for effectiveness. In this study, we compare our proposed model against the existing models for Packet Delivery Ratio (PDR), End-to-End Delay (EED), and Packet Loss Ratio (PLR) under two major categories: Speed of Internet of Vehicles (SoV) and Number of Tasks (NoT).

The following are Simulation Scenarios

Scenario 1: Speed of Internet of Vehicles (S-IoV)

The proposed system's execution is studied in this scenario. In this scenario, the number of S-IoV is 650, which runs at different speeds from 5 to 25 ms and 600 is the NoT, with 550 Mb as its fixed size.

Figure 4 illustrates the percentage of PDR of responses within the specified time of our proposed SDN-VANET model; the percentage decreases as the S-IoV vehicles increase. The increasing speed is S-IoV's migration from one domain and getting into another before the responses are delivered and this also increases the broken cases of links. The delay and number of responses that cross the time limit are increased due to these problems. However, when compared to other systems, our proposed method is superior because it measures routes with less data transmission and can compute paths regionally using DL-based SDN Controllers (DL-SDN-C) that significantly reduce delays and perform time-sensitive tasks.

The average End-to-End Delay (EED) with various S-IoV is illustrated in Fig. 5. Due to the broken links that require maintenance, there is a delay in the high S-IoV vehicles. However, since the proposed system considers the transmission time while selecting the optimal routes, it yields good results. The paths can be computed and many broken links are maintained regionally by the GRRL + SDN in the proposed system. So, the time for correcting the broken links is reduced, resulting in a minimization of total EED. Nonetheless, the SDN controller that always contains holistic information related to the whole network is dependent on the proposed system. Thus, the optimal routes are computed by it rapidly.

The number of hops and bottlenecks on the network determines the EED. Since the controller calculates and controls the paths for each segment to each RSU on the fly, S-IoV works better, which helps get data packets to where they need to go quickly.

The PLR with numerous S-IoV is shown in Fig. 6. High vehicle speed is defined as the frequent switching of Internet of Vehicle (IoV) between a few FC device domains and the increase in the number of packets that cannot reach their destinations accurately due to many failed links. The GRRL + SDN is used by our proposed system to manage many of the migrations of IoV quickly and fail connections. Thus, the number of PLR is minimized, resulting in better performance than other systems.

Scenario 2: Tasks Count

The performance of the proposed system is studied in this scenario with an increased number of activities. 900 IoV moves at a constant speed of 12 ms. 250, 500, 750, 1000, 1500, and 2000 are the NoT. Fixed-size of 500 Mb is possessed for all of these tasks.

The percentage of responses that come from not crossing our proposed system's time limit with multiple activities is shown in Fig. 7. Having excessive processing operations and overhead that escalates the delay means increasing the NoT. However, since it is a distributed system, this performance metric's result is better in the proposed system than in other systems. As a result, the processing operations and overhead are shared among the SDN controllers, significantly lowering EED and increasing the response frequency that achieves their delay.

The average EED is illustrated in Fig. 8 with adequate activities and the delay will be more for the same reasons explained above. Similarly, the predominant routes with shorter transmission time and densely packed tasks for the original response increase the NoT. So, the roads will be chosen with a high broadcasting time for the subsequent responses. Nonetheless, as the proposed system computes the routes with less delay and considers the available bandwidth, its delay is lower than other systems.

The proposed hierarchical system is shown in Fig. 9 with the existing models and the proposed SDN-VANET system is compared in the aspects of packet loss with increased activity. Our proposed system computes the routes with less delay and applies the GRRL + SDN in a shared manner capable of building new routes and rebuilding the broken links perfectly, it is the optimum one.

Scenario 3: Routing Overhead (RO)

For all protocols, RO is determined as in Fig. 10. The total RO is said to have increased with the average SoV. It shows that more traffic will be generated by redundancy in congested road segments. When compared with the more RSUs, routing overhead for DL-based SDN VANET is less. EC devices maintain the road segment level routing table and only when it demands the routes be found outside the road segments and the SDN controller purely determined this, so the RREQ is not required to be sent to all road segments.

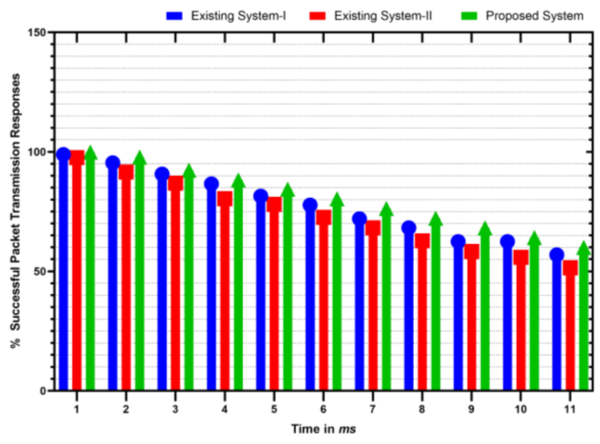

Fig. 4: % of Successfully PDR on time

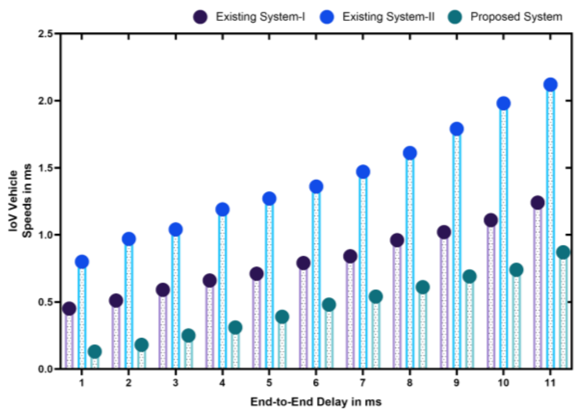

Fig. 5: Average EED vs S-IoV

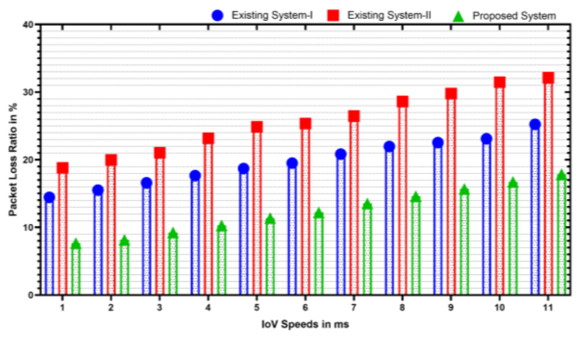

Fig. 6: Packet Loss Ratio (PLR) vs S-IoV

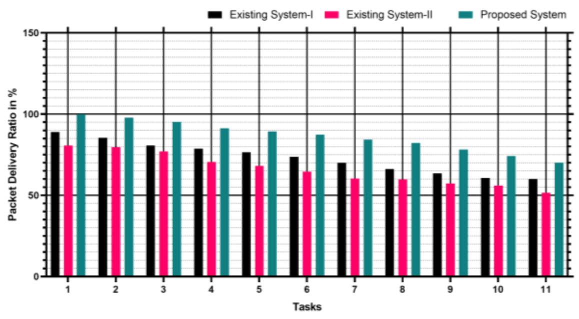

Fig. 7: PDR vs NoT

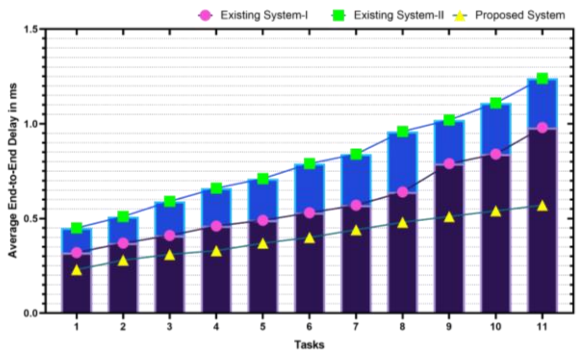

Fig. 8: Average EED vs NoT

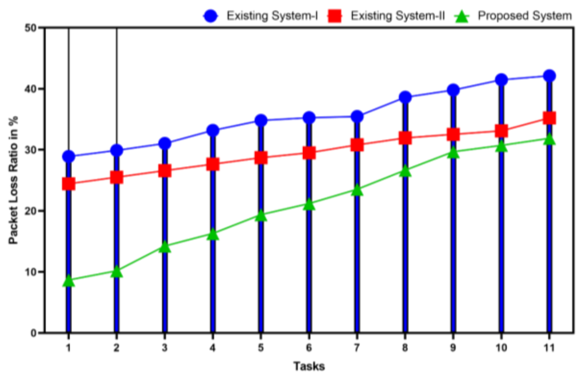

Fig. 9: PLR vs NoT

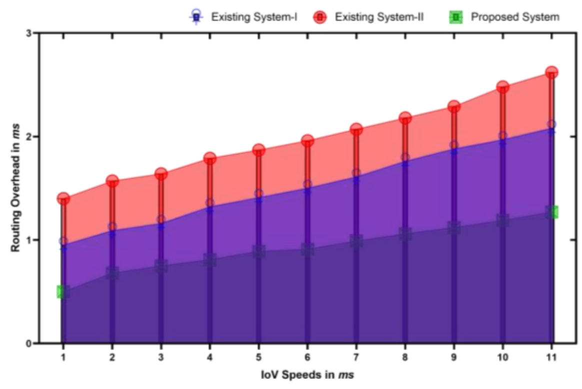

Fig. 10: RO vs S-IoV

Table 1: Metrics of simulation

Parameter for simulation	Threshold Value
Topology	Random and grid
Size of packets	1024 bytes
IEEE 802.11 MAC	802.11a
Covered area by topology	5 × 5 km^2
Data transfer rates	2 Mbps
Mobility	Static (none)
No. of EC clusters	3
No. of RSUs provided within each fog cluster	2
Each IoV has a specified No. of sensors	5
The speed limit for vehicles	25 ms
Network	Open flow 5.0
Simulation time	15 m

Conclusion

The proposed model attempts to offer cars in the SDN-VANET infrastructure congestion-free routes with an average minimal latency. Congestion causes longer journey durations, lowers vehicle speed, and prolongs latency in VANET. In a VANET, the SDN controller gathers data about the automobiles on every road to determine the congestion on that route. The data is subsequently analyzed using our suggested learning method. GRRL, a wholly distributed packet routing method based on multi-agent DRL, was introduced. To enhance the route learning more accurately, we created a GRU + FFNN framework with appropriate activation functions. Through acknowledgment of the information exchange mechanism, each agent in GRRL may learn an adaptive routing strategy through contact with the environment. The findings demonstrate that the suggested model has a higher PDR and a lower EED than the existing models.

Acknowledgment

We appreciate the experts' pertinent and helpful comments to enhance this research work. We also thank Bharathiyar University, Coimbatore, 641046, Tamil Nadu, India, for supporting our works.

Author’s Contributions

Savitha Kumar: Writing - original drafted, validation, conceptualization, investigation, and formal analysis.

Chandrasekar Chinnasamy: Project administration, visualization, writing-review, and edited, supervision.

Ethics

This manuscript substance is the author’s own original work and has not been previously published somewhere else. The authors have already read and approved the manuscript, and no potential ethical issues are immersed.

References

Arif, M., Wang, G., Balas, V. E., Geman, O., Castiglione, A., & Chen, J. (2020). SDN-based communications privacy-preserving architecture for VANETs using fog computing. *Vehicular Communications*, *26*, 100265. DOI.org/10.1016/j.vehcom.2020.100265

Arif, M., Wang, G., Wang, T., & Peng, T. (2018, December). SDN-based secure VANETs communication with fog computing. In *International Conference on Security, Privacy and Anonymity in Computation, Communication, and Storage* (pp. 46-59). Springer, Cham. DOI.org/10.1007/978-3-030-05345-1_4

Azizian, M., Cherkaoui, S., & Hafid, A. S. (2017). Vehicle software updates distribution with SDN and cloud computing. *IEEE Communications Magazine*, *55*(8), 74-79. DOI.org/10.1109/MCOM.2017.1601161

Bitam, S., Mellouk, A., & Zeadally, S. (2013). HyBR: A hybrid bio-inspired bee swarm routing protocol for safety applications in vehicular ad hoc networks (VANETs). *Journal of Systems Architecture*, *59*(10), 953-967. DOI.org/10.1016/j.sysarc.2013.04.004.

Cheng, N., Lyu, F., Chen, J., Xu, W., Zhou, H., Zhang, S., & Shen, X. (2018). Big data-driven vehicular networks. *IEEE Network*, *32*(6), 160-167. DOI.org/10.1109/MNET.2018.1700460

Daraghmi, Y. A., Yi, C. W., & Stojmenovic, I. (2013). Forwarding methods in data dissemination and routing protocols for vehicular ad hoc networks. *IEEE Network*, *27*(6), 74-79. DOI.org/10.1109/MNET.2013.6678930

He, Z., Cao, J., & Liu, X. (2016a). SDVN: Enabling rapid network innovation for heterogeneous vehicular communication. *IEEE Network*, *30*(4), 10-15. DOI.org/10.1109/MNET.2016.7513858

He, Z., Zhang, D., & Liang, J. (2016b). Cost-efficient sensory data transmission in heterogeneous software-defined vehicular networks. *IEEE Sensors Journal*, *16*(20), 7342-7354. DOI.org/10.1109/JSEN.2016.2562699

Hussein, A., Elhajj, I. H., Chehab, A., & Kayssi, A. (2017, May). SDN VANETs in 5G: An architecture for resilient security services. In *2017 Fourth International Conference on Software Defined Systems (SDS)* (pp. 67-74). IEEE. DOI.org/10.1109/SDS.2017.7939143

Ku, I., Lu, Y., Gerla, M., Gomes, R. L., Ongaro, F., & Cerqueira, E. (2014, June). Towards software-defined VANET: Architecture and services. In *2014 13th annual Mediterranean ad hoc networking workshop (MED-HOC-NET)* (pp. 103-110). IEEE. DOI.org/10.1109/MedHocNet.2014.6849111

Lai, C., Zhou, H., Cheng, N., & Shen, X. S. (2017). Secure group communications in vehicular networks: A software-defined network-enabled architecture and solution. *IEEE Vehicular Technology Magazine*, *12*(4), 40-49. DOI.org/10.1109/MVT.2017.2752760

Liu, J., Wan, J., Zeng, B., Wang, Q., Song, H., & Qiu, M. (2017). A scalable and quick-response software-defined vehicular network assisted by mobile edge computing. *IEEE Communications Magazine*, *55*(7), 94-100. DOI.org/10.1109/MCOM.2017.1601150

Muhizi, S., Shamshin, G., Muthanna, A., Kirichek, R., Vladyko, A., & Koucheryavy, A. (2017, June). Analysis and performance evaluation of SDN queue model. In *International Conference on Wired/Wireless Internet Communication* (pp. 26-37). Springer, Cham. DOI.org/10.1007/978-3-319-61382-6_3

Perkins, C. E., & Royer, E. M. (1999). Ad-Hoc on-Demand Distance Vector Routing. In *Proceedings WMCSA’99. Second IEEE Workshop on Mobile Computing Systems and Applications*, 90–100. IEEE. DOI.org/10.1109/MCSA.1999.749281

Qaisar, S., & Riaz, N. (2016, July). Fog networking: An enabler for next-generation internet of things. In *International Conference on Computational Science and its Applications* (pp. 353-365). Springer, Cham. DOI.org/10.1007/978-3-319-42108-7_27

Shirmarz, A., & Ghaffari, A. (2020). Performance issues and solutions in SDN-based data center: A survey. *The Journal of Supercomputing*, *76*(10), 7545-7593. DOI.org/10.1007/s11227-020-03180-7

Siddiqa, A., Hashem, I. A. T., Yaqoob, I., Marjani, M., Shamshirband, S., Gani, A., & Nasaruddin, F. (2016). A survey of big data management: Taxonomy and state-of-the-art. *Journal of Network and Computer Applications*, *71*, 151-166. DOI.org/10.1016/j.jnca.2016.04.008

Truong, N. B., Lee, G. M., & Ghamri-Doudane, Y. (2015, May). Software-defined networking-based vehicular Adhoc network with fog computing. In the *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)* (pp. 1202-1207). IEEE. DOI.org/10.1109/INM.2015.7140467

Vaquero, L. M., & Rodero-Merino, L. (2014). Finding your way in the fog: Towards a comprehensive definition of fog computing. *ACM SIGCOMM computer communication Review*, *44*(5), 27-32. DOI.org/10.1145/2677046.2677052

Vladyko, A., Muthanna, A., & Kirichek, R. (2016). Comprehensive SDN testing based on model network. In *the Internet of Things, Smart Spaces and Next Generation Networks and Systems* (pp. 539-549). Springer, Cham. DOI.org/10.1007/978-3-319-46301-8_45

Zhu, M., Cao, J., Pang, D., He, Z., & Xu, M. (2015, August). SDN-based routing for efficient message propagation in VANET. In *International conference on wireless algorithms, systems, and applications* (pp. 788-797). Springer, Cham. DOI.org/10.1007/978-3-319-21837-3_77