A Reliable Zero-Trust Network for Task Offloading in Vehicular Systems Using an Asynchronous Federated Learning Approach in 6G

Prakhar Consul*
Neeraj Joshi*
E21SOEP004@bennett.edu.in
E22SOEP004@bennett.edu.in
School of Computer Science, Engineering and Technology, Bennett University
Greater Noida, Uttar Pradesh, India

Ishan Budhiraja
School of Computer Science, Engineering and Technology, Bennett University
Greater Noida, Uttar Pradesh, India
ishan.budhiraja@bennett.edu.in

Sujit Biswas
Cybersecurity and FinTech, Computer Science Department, City, University of London, and Research Associate (Honorary) at CBT, University College London (UCL)
United Kingdom
sujit,biswas@city.ac.uk

Neeraj Kumar
Department of CSE, Thapar Institute of Engineering and Technology
Patiala, Punjab, India
neeraj.kumar@thapar.edu

Sachin Sharma
Chief Manager (Systems), State Bank of India
Panchkula, Chandigarh, Punjab, India
sachin.sharma@sbi.co.in

Ajith Abraham
School of Artificial Intelligence, Bennett University
Greater Noida, Uttar Pradesh, India
Ajith.Abraham@bennett.edu.in

ABSTRACT

In the emerging 6G era, vehicles are extensively connected to wireless networks through edge-accessible roadside units (RSUs). The increasing number of connected vehicles and vehicle services introduces a significant security challenge known as the "zero-trust network (ZTN)." This necessitates a shift from traditional methods of resource slicing and scheduling. This study focuses on ensuring reliable 6G vehicular services, particularly addressing the scenario of task offloading between vehicles, which involves managing communication resources. We propose a method that uses a logical model to assign an edge node score (ENS) to evaluate the security of edge nodes, thereby protecting vehicles from potential threats posed by untrusted edge access points. Vehicles select edge nodes with high ENS scores for task offloading. Also, we used a federated asynchronous reinforcement learning approach to enhance the management of offloaded tasks. Simulation results show that the proposed approach effectively organizes the resources and ensures the security of vehicle data.

CCS CONCEPTS

• **Zero Trust Architecture**; • **Next Generation Communications**; • **Asynchronous Federated Learning**; • **Task Offloading**;

KEYWORDS

6G, Edge Vehicular Network, Edge Node Score, Asynchronous Federated learning, Resource slicing, Zero-trust Network

ACM Reference Format:
Prakhar Consul, Neeraj Joshi, Ishan Budhiraja, Sujit Biswas, Neeraj Kumar, Sachin Sharma, and Ajith Abraham. 2024. A Reliable Zero-Trust Network for Task Offloading in Vehicular Systems Using an Asynchronous Federated Learning Approach in 6G. In *Proceedings of SIGCOMM 2024 Workshop on Zero Trust Architecture for Next Generation Communications ((ZTA-NextGen'24))*. ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Federated learning (FL) and reinforcement learning (RL) are two effective learning technologies that have been widely used in recent investigations to address wireless network problems with optimization. FL is a machine learning framework that may efficiently allow numerous users to use data while adhering to legal constraints, data security standards, and privacy concerns for users [1, 5]. RL is used in conjunction with the Markov process, a continuous mathematical simulation technique, to solve non-convex problems [12]. In order to minimize the energy consumption, the authors in [10, 11] integrated FL with deep Q-network (DQN) in RL and optimization will be done with the task offloading method based on the environmental state seen by the base station. Nevertheless, this effort fails to integrate cloud servers to deliver car services; rather, it solely takes into account scenarios that fall inside a base station's coverage area. The issue of resource allocation of the edge storage of smart devices was studied by the author in [9] utilizing FL and double-DQN-based RL. According to reports, this method's simulation outcome is superior to the traditional method that was compared. However, the DDQN algorithm's complexity increases when addressing the continuous action state, and its study goal does not contain moving vehicles. The three main edge computing problems—resource allocation, computation offloading, and service caching placement—were simultaneously optimized by the authors

*Both authors contributed equally to this research.

(ZTA-NextGen'24), , Sydney, Australia

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

of [8, 14]. The author used FL in conjunction with a two-layer DQN method to optimize these problems; however, FL is only utilized for improving the edge storage; the trustworthiness of the edge nodes is not taken into account in the overall plan.

In general, an effective resource allocation strategy must safeguard client privacy in addition to optimizing resource deployment and enhancing service dependability [7]. Choosing reliable edge nodes and minimizing the amount of original data transmitted are two crucial factors to take into account. The authors of [4, 13] developed an edge node selection technique based on reputation values. Reputation value was computed by analyzing the historical service records of edge nodes. Motivated by this effort, we could urge the cars to assign jobs to the edge services based on the RSU's reputation value in order to guard against hostile vehicle attacks on the RSU. Once the right edge nodes have been chosen, vehicular jobs must be reliably fulfilled. The author of [6] proposed a dependable estimation approach for the UAV's condition using coding technology; the dependability is measured by the communication link's error rate. The author of [3] presented a testing strategy for IoTD that primarily identified access device security from a software and hardware standpoint. The deployment of vehicular fog node slices was optimized in [2] by the author using the simplest RL technique, Q-learning, and measuring reliability using the probability of vehicle communication interruption.

1.1 Contribution

- In this study, we primarily focus on edge, convergence, and cloud servers in a typical vehicular edge computing situation. The edge server can be installed on the RSU to service vehicles, but the convergence server has a higher processing capacity. To finish offloading a portion of the vehicular tasks and develop the DL models to guard against the convergence servers' lack of processing power, the cloud server will interact with several convergence servers.
- We set up an identity assessment technique based on a personal logic model to continuously watch the reputation rating and assess the security degree of the network connection points in order to address the possible security concerns from the trustless system. Using the access point ratings and the needs of their tasks, vehicles may locate reliable network access nodes. In order to satisfy the availability needs of on-board tasks, the federated asynchronous RL algorithm is utilized to improve slice resource allocation.

1.2 Organisation

The article is structured as follows: The system model and the formulation of the problem are described in depth in Sections II and III. In Section IV, the optimized problem's solution is demonstrated. In Section V, the suggested scheme's effectiveness is assessed. Section VI contains the final presentation of the conclusion.

2 SYSTEM MODEL

2.1 Vehicular Network

Consider a three-layered infrastructure made up of edge servers, convergence servers, and cloud servers for a hierarchical vehicular network. RSUs are positioned close to highways and equipped with edge servers, as illustrated in Fig. 1. Wired backhauls connect the convergence servers and edge servers. In terms of computing capacity, a convergence server is significantly more potent than an edge server. The convergence servers' service zones encompass the linked edge servers in the interim. The cloud server is connected to convergence servers, offering redundant computational resources. The convergence and cloud server, an edge server, may manage the vehicle activities in order to reduce service latency.

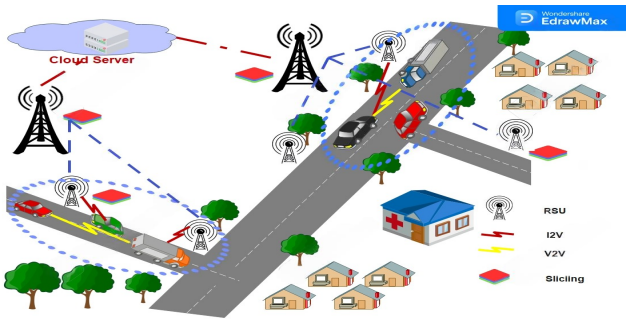

Figure 1: System Model: VN Architecture

2.2 Communication and Computation Model

The communication between the vehicle and RSU is referred to as V2I. V2V stands for vehicle-to-vehicle communication. The vehicular task could be offloaded via V2V to the future service area of a new convergence server when a vehicle is ready to leave the convergence server's service area, cutting down on the handover delay. Let us consider the following: $k \in \mathcal{K} = \{0, 1, 2, \ldots, K\}$, $j \in \mathcal{J} = \{0, 1, 2, \ldots, J\}$. A vehicle k uses V2I communication, and a vehicle j uses V2V communication. For effective spectrum use, the orthogonally allotted uplink spectrum is intended to be shared by the V2V and V2I. We now derive the transmission rate for V2I and V2V.

The transmission rate of any resource block (RB) x that vehicle k occupies in V2I communication is expressed as

$$q_k^x = l \cdot \log(1 + \gamma_k) - \sqrt{\mu^{-1}\left[1 - (\gamma_k + 1)^{-2}\right]}\frac{V^{-1}(\varepsilon)}{\ln 2}, \quad (1)$$

where l is the bandwidth and $x \in \mathcal{X} = (1, 2, \ldots, X)$. In this case, μ represents the URLLC packet size. The inverse Q-function is $V^{-1}(\varepsilon)$, and the URLLC reliability threshold is ε. Next, γ_k can be acquired by

$$\gamma_k = \frac{PW_k g_k}{\sigma^2 + \sum_{j \in \mathcal{J}} \rho_{j,k} PW_k \tilde{l}_j} \quad (2)$$

where g_k represents the channel gain of V2I and PW_k, PW_j stands for the transmission power of the vehicle k, j. In this case, $\tilde{l}_j$ represents the vehicle j's interference power gain, and $\rho \in \{0, 1\}$, $\rho_{j,k} = 1$ indicates that during V2V communication, the j^{th} vehicle uses the spectrum resource of the k^{th} vehicle, which results in the channel interference. Consequently, for vehicle k, the data transmission rate

is provided by

$$Q_k = \sum_{x=1}^{X} q_k^x \tag{3}$$

The V2V transmission rate of vehicle j-occupied resource block (RB) x is obtained using

$$q_j^x = l \cdot \log\left(1+\gamma_j\right) - \sqrt{\mu^{-1}\left[1-\left(\gamma_j+1\right)^{-2}\right]}\frac{V^{-1}(\varepsilon)}{\ln 2}, \tag{4}$$

where γ_j represents

$$\gamma_j = \frac{PW_j l_j}{\sigma^2 + Z_{\mathrm{q21}} + Z_{\mathrm{Q2\ Q}}} \tag{5}$$

Here, l_j denotes the channel gain of Q2 Q, and Z_{Q21} denotes the interference when vehicle k shares RB with j, which is given by

$$Z_{\mathrm{Q2X}} = \sum_{k\in K} \rho_{k,j} PW_k \tilde{g}_k, \tag{6}$$

where $\rho \in \{0,1\}, \rho_{k,j} = 1$ and $\tilde{g}_k$ is the interference power gain of vehicle k indicates that, during V2I communication, the k^{th} vehicle utilizes the j^{th} vehicle's spectrum resource.

The interference that occurs when vehicle j shares RB with vehicle j' is denoted by $Z_{\mathrm{Q2\ Q}}$. Next, we have

$$Z_{\mathrm{Q2\ Q}} = \sum_{j'\in J, j'\neq j} \rho_{j',j} PW_{j'} \tilde{l}_{j'} \tag{7}$$

$$Q_j = \sum_{x=1}^{X} q_j^x. \tag{8}$$

Task u's wireless access network transmission time takes the following form:

$$D_u^{\mathrm{wireless}} = \phi\frac{c_u}{Q_j^u} + \frac{c_u}{Q_k^u}(j\in\mathcal{J}, k\in\mathcal{K}), \tag{9}$$

The likelihood of RB transmission failure is given by taking into account the situation of RB transmission failure as well.

$$PW_u^{\mathrm{error}} = 1-(1-\eta)^{X_u}, \tag{10}$$

where X_u is the number of RBs occupied by task u, and η is the likelihood of an RB transmission failure.

Depending on the offloading scenarios, task u may need to be routed over wired backhauls to the cloud server or convergence servers after reaching the RSU via the 6H wireless network. $Q_{\mathrm{wired}} = \left\{Q_{\mathrm{wired}}^{\mathrm{con}}, Q_{\mathrm{wired}}^{\mathrm{cld}}\right\}$ is the transmission rate of wired backhauls. The transmission rates between the RSU the cloud server and the convergence server are indicated by the variables $Q_{\mathrm{wired}}^{\mathrm{con}}$ and $Q_{\mathrm{wired}}^{\mathrm{cld}}$, respectively. Task u's time delay when transmitted over wired backhauls is seen as

$$D_u^{\mathrm{wired}} = \frac{c_u}{Q_{\mathrm{wired}}}, \tag{11}$$

and the total transmission delay of task u is expressed as

$$D_u^{\mathrm{tran}} = D_u^{\mathrm{wireless}} + D_u^{\mathrm{wired}}. \tag{12}$$

Following transmission of the vehicular task u to the target offloading server, the server will furnish computational resources. The delay in task execution is expressed as

$$D_u^{\mathrm{comp}} = \frac{c_u b_u}{i_m^u e_m} \tag{13}$$

where I_m^u represents the percentage of server m's computing resources that are occupied by task u, and e_m indicates the total number of computation resource slices of server m. As a result, the whole time taken to do the vehicular task u takes the form:

$$D_u = D_u^{\mathrm{tran}} + D_u^{\mathrm{comp}} \tag{14}$$

In addition to the time delay, the energy usage for the vehicular task u has to be taken into account. It takes the shape of

$$F_u = PW_j\phi\frac{c_u}{Q_j^u} + PW_k\frac{c_u}{Q_k^u} + PW_{\mathrm{wired}} D_u^{\mathrm{wired}} + PW_{\mathrm{comp}}^m I_m^u, \tag{15}$$

where the computing power of server m is represented by PW_{comp}^m, and the transmission power over the wired backhauls is denoted by PW_{wired}.

This study calculates the total cost by weighting and adding the time delay and energy consumption of job u at time d.

$$H_u(d) = \theta D_u(d) + (1-\theta)F_u(d), \tag{16}$$

$\theta \in [0,1]$ in this case. The percentage of energy consumption and time delay in the cost function is adjusted using the weight θ.
Requirement:

- $\mathbb{S}$ state space and $\mathbb{A}$ action space
- ϕ discount factor and g^t penalty parameter

[1] Initialization: Process step counter $t \leftarrow 1$ Initialization: ω, ω_v, t_c and $T = 0$ Initialization: ω', ω_v' **Repeat** $d\omega \leftarrow 0$ and $d\omega_v \leftarrow 0$ Process Synchronization $\omega' = \omega$ and $\omega' = \omega t_{\mathrm{start}} = t$ Find out the state s_t **Repeat** Perform k_d as per policy $\pi\left(k_d \mid n_d\omega'\right)$ and find reward u_d and new state k_{n+1} $d \leftarrow d+1$ $U \leftarrow U+1$ **Until** terminal u_d or $d - d_{\mathrm{start}} == d_{\max}$ $x \in \{d-1, \ldots, d_{\mathrm{start}}\}$; $Z \leftarrow z_x + \xi Z$ **if** $t\%t_c = 0$ **then** $\omega = \frac{1}{\mathrm{con}}\sum_{i=1}^{i=con}\varphi_i \cdot \omega_i$ Perform asynchronous update of ω using $d\omega$ and of ω_I using $x\omega_u$ **until** $D > D_{\max}$

2.3 Zero trust architecture

The age of 6G is one of pervasive intelligence. In contrast to 5G, 6G will include a sizable number of devices linked to the network, and it is more possible that some of those devices may be malicious and could cause network damage. Information security will provide major issues in the 6G era. This study suggests a zero trust architecture for 6G vehicle networks in the standard scenario to withstand the security risks of 6G. A significant number of autos will interact with RSU and assign it responsibilities. The proposed zero trust architecture incorporates an RSU selection method based on the subjective logic model, taking into account the possibility of hostile manipulation of the vehicle to attack the RSU. The security issues that the network faced in the context of zero trust may be summed up as follows:

- The reputation of a network cannot be determined solely by its location.
- Every piece of hardware, every user, and every network communication need to be verified and approved.
- Numerous sources of data must be used to calculate the security procedure, which must be dynamic.

Similar dangers will also be encountered by the 6G vehicular network about security issues in zero-trust networks. The use of C-V2X (Cellular-Vehicle to Everything) will result in an increasing number of RSUs being positioned alongside the road. Traffic accidents are most likely to occur once an RSU is attacked and malfunctions. By

extending the use case to a 6G vehicular network and considering the job offloading scenario with V2I communication, all participating objects in vehicular networks should collaborate to establish an end-to-end trust relationship. These participating objects include vehicles, RSUs, and even the cloud centre.

3 PROBLEM FORMULATION

In order to solve the issue of vehicular task offloading, vehicles will first locate a reliable network access point and then set up slice resources according to the task's bandwidth and latency requirements. In addition to ensuring extremely high task reliability, the allotted slice resources should minimize energy consumption. The following is how the problem is stated:

$$
\begin{aligned}
P.F. \quad &: \min_{\lambda_m^u, \phi_u, I_u^U, X_u} \lim_{d \to \infty} \sum_{u}^{U} \sum_{m=1}^{m=3} \lambda_m^u(d) \cdot H_u(d) \qquad (17)\\
s.t.\ \mathbb{C}_1 \quad &: \sum_{m=1}^{m=3} \lambda_m^u(d) = 1, \forall u \in \mathcal{U},\\
\mathbb{C}_2 \quad &: 0 \leq \sum_{u=1}^{U} \lambda_m^u I_m^u \leq 1, \forall m \in \mathcal{M},\\
\mathbb{C}_3 \quad &: D_u \leq \tau_u, \forall u \in \mathcal{U},\\
\mathbb{C}_4 \quad &: \sum_{u=1}^{U} X_u \leq X,
\end{aligned}
$$

where the major variables are X_u, I_m^u, ϕ_u, and λ_m^u. The problem has the following properties: $C1$ indicates that any task u can only be offloaded to one of the three servers; $C2$ indicates that server m cannot provide task u with computing resources greater than its computation capacity; C3 indicates that no vehicular task's completion time can be longer than the required delay; and C4 indicates that no task's slice of communication resources can be larger than the total number of slices. Problem (17) contains discrete or continuous variables. It is not a convex problem, and calculating the objective function directly will not yield the optimal solution. The challenge is solved in this work using a federated asynchronous reinforcement learning technique.

4 PROPOSED SOLUTION

First and foremost, the problem formulated is transformed into a multiagent model-free decision-making problem using the Markov decision process (MDP) model. By utilizing the MDP, an RL multiagent environment is generated, enabling an intelligent agent to be trained in making optimal sequential decisions through trial-and-error interactions with the environment.

4.1 MDP

The MDP model is denoted by the set $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{V})$, where $\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}$, and $\mathcal{V}$ correspond to the state space, action space, state transition probability, immediate reward function, and state value function, respectively. A comprehensive explanation of the model is provided below:

4.1.1 State Space. For each agent n, the state that has been observed at the t^{th} time slot is presented in the following manner:

$$
s_n^t = \left\{ h_{n,1}^t, h_{n,2}^t, I_{m,n}^t, I_{n',n}^t, Q_n^t, E_n^t \right\} \qquad (18)
$$

where $h_{n,1}^t \& h_{n,2}^t$ are the channel gain from the DDT to 1^{st} DDR link and 2^{nd} DDR link, respectively. $I_{m,n}^t$ and $I_{n',n}^t$ shows the interference link from the CUE-to-DDR link and interference links from the DDT-to-DDR link at time slot (t), respectively. Q_n^t is the data queue link for the n^{th} DGU. E_n^t represents the energy queue link of the n^{th} DGU.

4.1.2 Action Space. This state determines which time-period to use and how much power should be transmitted during that time slot. As a result, the action of each DGU during the t^{th} time slot is specified as

$$
a_n^t = \{\tau_0, \alpha_n\} \qquad (19)
$$

where $\tau_0 \in \{0, 1\}$, and $\alpha \in \left\{0, \frac{1}{L}\alpha_{\max}, \frac{2}{L}\alpha_{\max}, \ldots, \alpha_{\max}\right\}$. Here, L denotes the number of discrete stages of maximum transmit power.

4.1.3 Transition Probability. The probability of transition from a present state $s_n^t \in \mathcal{S}_n$ to a next state $s_n^{t+1} \in \mathcal{S}_n$ after executing an action $a_n^t \in \mathcal{A}$ is given as $P\left(s_n^{t+1} \mid s_n^t, a_n^t\right)$.

4.1.4 Reward. The n^{th} DGU receives the following immediate reward for the t^{th} time slot.

$$
\begin{aligned}
r_n^t\left(s_n^t, a_n^t\right) &= \sum_{n=1}^{N} \Omega_1 \left[\overline{EE}_n^t + \bar{D}_n^t\right]\\
&+ \sum_{n=1}^{N} \Omega_2 \mathcal{G}\left(\bar{R}_{n,1}^t - \bar{R}_{n,1}^{t,\min}\right) + \sum_{n=1}^{N} \Omega_3 \mathcal{G}\left(\bar{R}_{n,2}^t - \bar{R}_{n,2}^{t,\min}\right)\\
&+ \sum_{n=1}^{N} \Omega_4 \mathcal{G}\left(D_n^t - \lambda\right) \qquad (20)
\end{aligned}
$$

Here, $\mathcal{G}(x)$ is a piecewise function and its value is given as follows:

$$
\mathcal{G}(x) = \begin{cases} Z, & x \geq 0 \\ \mathbf{x}, & x < 0 \end{cases}
$$

where Z is set as a positive constant to indicate revenue.

4.2 D2PG and Federated learning

D2PG is based on a widely used RL technique called deep deterministic, which blends policy-based and value-based techniques. D2PG interacts with the environment concurrently through the utilization of multiple processes. The learning outcomes are combined by each process and sent to the global model for the gradient update. As can be seen from the above, D2PG offers the best course of action for every state-activity pair to produce the ideal power and time. However, the implementation of the D2PG framework might not be appropriate for the following reasons: (i) Training each agent takes a significant amount of time. (ii) It takes a significant amount of energy to train each agent separately. We propose the FL-based strategy, which allows a loose federation of participating users under the supervision of a central server, to overcome these issues. In FL, the user's raw data is divided into the model training data, and a sporadic average of the local models is sent to the centralized server. This technique improves the distributed Deep Neural

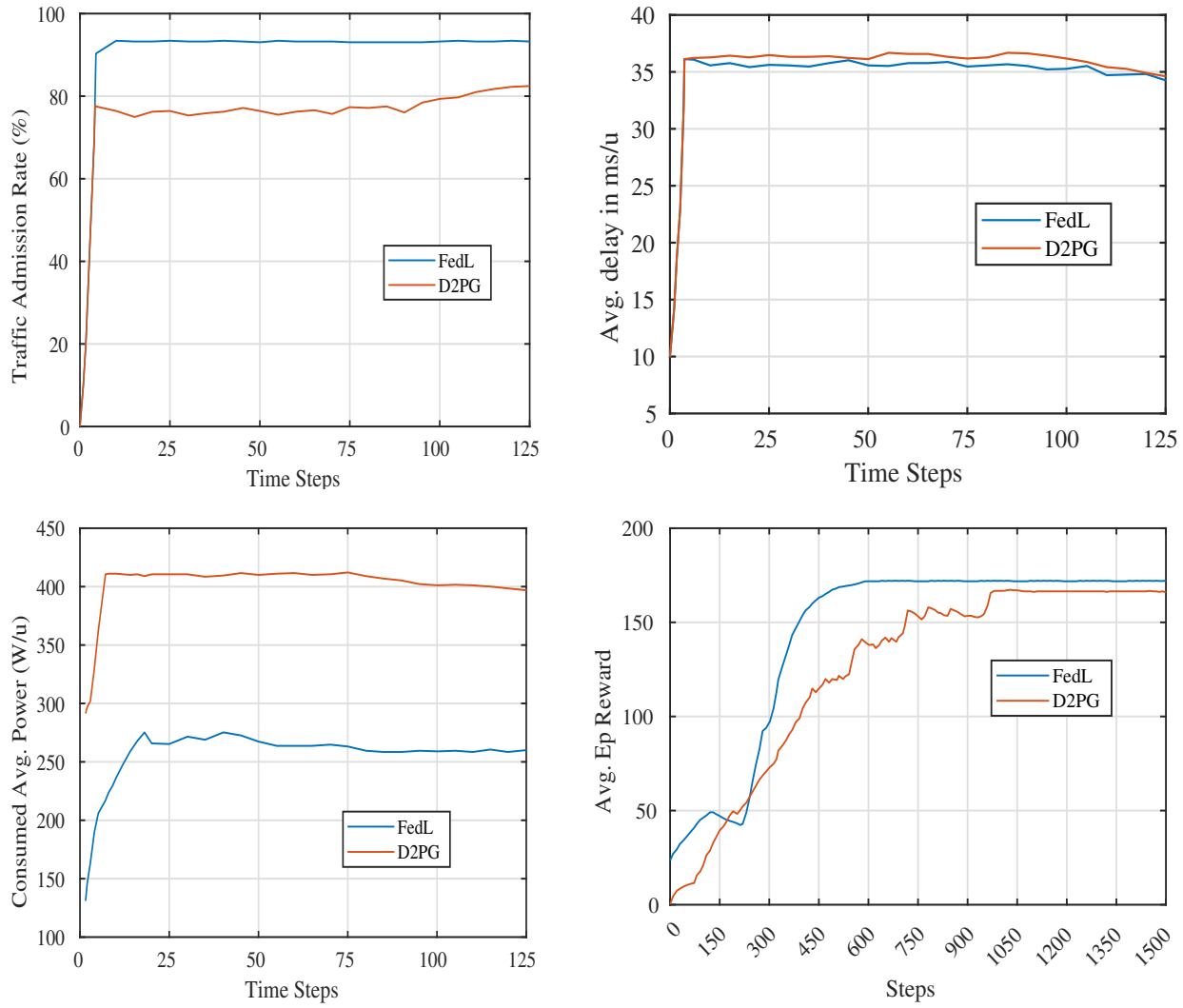

Figure 2: Comparative Results (a) Energy Consumption (b) Network Latency (c) Traffic Admission Rate & (d) Average Reward

Network (DNN) and D2PG training performance. It is also determined that in FL, the uploading overheads are almost negligible in comparison to centralized learning for two main reasons: (i) the training models' size is smaller than the raw models' size, and (ii) the averaging time frame is much longer than the training time frame.

5 SIMULATION RESULTS

We have established two service locations in the simulation. There is one convergence server per service region. Within a service area, there is a 250 m-long two-way road. Every car starts at a random location and travels at a 30 km/h average. There are 120 communication RB in a service region, and each communication has a 20kHz bandwidth. The edge, convergence, and cloud servers have processing power consumptions of 1, 10 & 100, respectively. Table 1 represents the simulation parameters used in this article. We view

Table 1: Simulation Parameters

Parameters	Values
Maximum Length of Road	250 m
Vehicle's Avg Speed	30 km/h
Power Consumed at Cloud, BS and Edge Node	100, 10, 1
Frequency at Cloud, BS and Edge Node	10, 1 & 0.1 GHz
Transmission Rate at Cloud & BS	4 Mbps, 10 Mbps
Transmission Power at Cloud & BS	1 W each

slices and Wireless Service Providers (WSP) as a single, separate network where there is a trade-off with WSP. For the entire environment, average cumulative and comparative performance is

displayed in Figures 2 (a), (b), and (c). The cumulative performance with D2PG settings is shown by one curve, whereas the cumulative performance for FedL settings is indicated by the other. As shown, FedL may achieve better than D2PG outcomes depending on various traffic conditions and network states, meeting the requirements of a dependable vehicular network in terms of admission rate, latency, and energy use. Since more slices directly affect cost functions and meet network constraints and thresholds, the number of slices increases, which in turn results in an increase in admission rate, as shown in Fig. 2 (a). It also represents that compared to D2PG, FedL offers a higher traffic admission rate.

Figure 2 (b) illustrates that D2PG has a higher delay than our suggested approach. It illustrates how the agent uses predefined weights to distinguish between different network costs. Figure 2 (c) illustrates how D2PG uses more power across the network when compared with our approach. Both curves slightly fall as soon as an agent uses trade-offs to pursue a multi-objective strategy. For instance, increasing vehicle traffic increases admission rates but also increases power consumption; as a result, we cannot anticipate a significant improvement in all measures as traffic increases since metrics are trade-offs. The FedL method, which is the one we suggested in the article, is contrasted with D2PG in Figure 2 (d). D2PG is a reinforcement learning system that uses off-policy data to constantly train a Q-function and a policy that associates states with actions. D2PG, which is based on gradient-based optimization, is utilized in situations with continuous action spaces. There are ten workers in each of FedL and D2PG. The system's reward increases as predicted during the training process. The reward typically remains at the ideal value once the goal solution is discovered. As can be observed, our suggested solution outperforms the other in terms of stability and convergence speed.

6 CONCLUSION

In this study, we suggested a zero-trust environment trustworthy method for slicing and scheduling URLLC resources for 6G vehicle networks. This strategy combined a scoring system with edge, convergence, and cloud servers, the three layers of the infrastructure. We have presented a logical model to calculate the trust score of edge nodes to shield cars from malicious node assaults. The problem was then optimized using an asynchronous fedL approach. The outcomes of the simulation demonstrated that our suggested approach may effectively distribute the resources needed for offloading active duties while safeguarding the vehicle's information security. Blockchain technology may be utilized in ZTA for vehicular 6G services in future research to better address information security concerns.

REFERENCES

[1] Ishan Budhiraja, Vineet Vishnoi, Neeraj Kumar, Deepak Garg, and Sudhanshu Tyagi. 2022. Energy-Efficient Optimization Scheme for RIS-Assisted Communication Underlaying UAV with NOMA. In *ICC 2022 - IEEE International Conference on Communications*. 1–6. https://doi.org/10.1109/ICC45855.2022.9838872

[2] Ahmad Hammoud, Hani Sami, Azzam Mourad, Hadi Otrok, Rabeb Mizouni, and Jamal Bentahar. 2020. AI, Blockchain, and Vehicular Edge Computing for Smart and Secure IoV: Challenges and Directions. *IEEE Internet of Things Magazine* 3, 2 (2020), 68–73. https://doi.org/10.1109/IOTM.0001.1900109

[3] Xumin Huang, Rong Yu, Jiawen Kang, Zhuoquan Xia, and Yan Zhang. 2018. Software Defined Networking for Energy Harvesting Internet of Things. *IEEE Internet of Things Journal* 5, 3 (2018), 1389–1399. https://doi.org/10.1109/JIOT.2018.2799936

[4] Yiming Huo, Xiaodai Dong, Wei Xu, and Marvin Yuen. 2019. Enabling Multi-Functional 5G and Beyond User Equipment: A Survey and Tutorial. *IEEE Access* 7 (2019), 116975–117008. https://doi.org/10.1109/ACCESS.2019.2936291

[5] Neeraj Joshi, Ishan Budhiraja, Deepak Garg, Sahil Garg, Bong Jun Choi, and Mubarak Alrashoud. 2024. Deep reinforcement learning based rate enhancement scheme for RIS assisted mobile users underlaying UAV. *Alexandria Engineering Journal* 91 (2024), 1–11. https://doi.org/10.1016/j.aej.2024.01.039

[6] Jiawen Kang, Zehui Xiong, Dusit Niyato, Shengli Xie, and Junshan Zhang. 2019. Incentive Mechanism for Reliable Federated Learning: A Joint Optimization Approach to Combining Reputation and Contract Theory. *IEEE Internet of Things Journal* 6, 6 (2019), 10700–10714. https://doi.org/10.1109/JIOT.2019.2940820

[7] Jianhui Liu and Qi Zhang. 2018. Offloading Schemes in Mobile Edge Computing for Ultra-Reliable Low Latency Communications. *IEEE Access* 6 (2018), 12825–12837. https://doi.org/10.1109/ACCESS.2018.2800032

[8] Nguyen Cong Luong, Dinh Thai Hoang, Shimin Gong, Dusit Niyato, Ping Wang, Ying-Chang Liang, and Dong In Kim. 2019. Applications of Deep Reinforcement Learning in Communications and Networking: A Survey. *IEEE Communications Surveys and Tutorials* 21, 4 (2019), 3133–3174. https://doi.org/10.1109/COMST.2019.2916583

[9] Mayra Samaniego and Ralph Deters. 2018. Zero-Trust Hierarchical Management in IoT. In *2018 IEEE International Congress on Internet of Things (ICIOT)*. 88–95. https://doi.org/10.1109/ICIOT.2018.00019

[10] Ramesh Sekaran, Rizwan Patan, Arunprasath Raveendran, Fadi Al-Turjman, Manikandan Ramachandran, and Leonardo Mostarda. 2020. Survival Study on Blockchain Based 6G-Enabled Mobile Edge Computation for IoT Automation. *IEEE Access* 8 (2020), 143453–143463. https://doi.org/10.1109/ACCESS.2020.3013946

[11] Himanshu Sharma, Ishan Budhiraja, Prakhar Consul, Neeraj Kumar, Deepak Garg, Liang Zhao, and Lie Liu. 2022. Federated learning based energy efficient scheme for MEC with NOMA underlaying UAV. In *Proceedings of the 5th International ACM Mobicom Workshop on Drone Assisted Wireless Communications for 5G and Beyond* (Sydney, NSW, Australia) *(DroneCom '22)*. Association for Computing Machinery, New York, NY, USA, 73–78. https://doi.org/10.1145/3555661.3560867

[12] Vineet Vishnoi, Ishan Budhiraja, Suneet Gupta, Neeraj Joshi, Anushka Nehra, and Haneef Khan. 2023. Deep Reinforcement Learning Based Energy Efficiency Maximization Scheme for Uplink NOMA Enabled D2D Users. In *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*. 4728–4734.

[13] Maoqiang Wu, Dongdong Ye, Jiahao Ding, Yuanxiong Guo, Rong Yu, and Miao Pan. 2021. Incentivizing Differentially Private Federated Learning: A Multidimensional Contract Approach. *IEEE Internet of Things Journal* 8, 13 (2021), 10639–10651. https://doi.org/10.1109/JIOT.2021.3050163

[14] Dongdong Ye, Rong Yu, Miao Pan, and Zhu Han. 2020. Federated Learning in Vehicular Edge Computing: A Selective Model Aggregation Approach. *IEEE Access* 8 (2020), 23920–23935. https://doi.org/10.1109/ACCESS.2020.2968399