

Multi-Objective Immune Algorithm for Internet of Vehicles for Data Offloading

B. Gomathi¹, S. T. Suganthi^{2,*}, T. N. Prabhu³ and Andriy Kovalenko⁴

¹Department of Information Technology, Hindusthan College of Engineering and Technology, Coimbatore, 641050, India

²Department of Computer Engineering, Lebanese French University, Erbil, 44001, Iraq

³Department of Information Technology, Sri Ramakrishna Engineering College, Coimbatore, 641022, India

⁴Department of Electronic Computers, Kharkiv National University of Radio Electronics, Kharkiv Oblast, 61000, Ukraine

*Corresponding Author: S. T. Suganthi. Email: suganthi@lfu.edu.krd

Received: 04 January 2022; Accepted: 14 February 2022

Abstract: On the Internet of Vehicle (IoV) devices, offloading data is the major problem because massive amounts of data generate energy consumption, and the execution cost is high. At present, accidents traffic management is highly prominent due to increased vehicles among the population. IoV is the only technology to help the transport system effectively. This data outreach the memory also has high energy consumption, and the storage cost is high. To overcome these issues, a Mobility aware Offloading scheme with Multi-Objective Immune Optimization algorithm (MOS-MOIO) is used in the cloud storage. The data is generated from the online sensor system. The generated information must alert the IoV users with a useful data offloading scheme. The result of the data offloading is evaluated with various performance metrics.

Keywords: Internet of vehicles; offloading; multi-objective immune optimization; offloading scheme; mobility aware offloading scheme

1 Introduction

Vehicles have become an indispensable aspect of people's daily lives as transportation has progressed. Vehicle use is increasing, making it more difficult to operate the current transportation system the likelihood of traffic collisions in metropolitan areas. As a result, the Internet of Vehicles (IoV) has gotten many bride attentions under the Internet of Things. Already, it has been deployed primarily in urban traffic environments. In contrast to conventional transportation systems, the Internet of Vehicles (IoV) focuses on information sharing between vehicles, humans, and roadside units (RSUs), with the purpose of enabling users to obtain real-time traffic information readily. On the Internet of Vehicles, sensors and microcontrollers in roadside equipment build an intelligent vehicle grid.

The resource-constraint such as storage, battery, central processing unit (CPU) are important issues in internet of things (IoT) devices. These devices work with the remote cloud sensors and helps to offload the requested tasks and complete the tasks [1]. The real-time routing helps the user and helps to improve their driving experiences. This routing also improves the efficient operation of traffic [2]. The data



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

collected from these devices were stored in the cloud platform. Data offloading is the central issue in the Cloud platform because there is data traffic in the networks.

The primary distinction between edge computing and cloud computing is that edge computing moves the jobs to the edge server close by, which can incredibly lessen the postponement of the job and decrease the capacity on the cloud terminal, and at a similar time has better security. In the dumping system, the allocation of unstable tasks might lead to specific servers' unnecessary capacity. When the capacity of Mobile Edge Computing (MEC) servers is temperamental, it will influence the server execution and even lead to server disappointment. Then again, the energy burned-through in the activity measure is likewise one of the estimation models [3]. Hence, while assigning resources, it is fundamental to upgrade the capacity and energy utilization of many MEC servers to lessen the cost of edge hardware [4–6].

Moreover, a computation job needs to process a set of data that the user must analyze, with some (or all) of the data being offloaded to the MEC server. Data offloading is the major problem in the existing system, and there are energy consumption and computational cost problems. To overcome this problem, the proposed method, Multi-objective immune optimization for data offloading, helps to focus on the challenge of effective resource management and intelligent (partial) data offloading techniques. The complete offloading decision-making process is carried out by incorporating and implementing a realistic cognitive-behavioral paradigm for the involved IoT entities. The proposed technique helps to reduce uploading delay, server energy consumption, and load balancing.

The rest of our research article is written as follows: Section 2 consists of a brief study of existing IoV offloading and mobility aware offloading scheme (MOS) with multi-objective immune optimization methods. Section 3 describes the working principle of the proposed model. Section 4 evaluates the result and gives a comparison of different algorithms. Section 5 concludes the research work.

2 Related Works

Vehicular ad hoc network (VANET) and Mobile ad hoc network (MANET) are based on mobility portable applications [7]. The vehicle network 5G Dynamic Spectrum Access (DSA) is used to improve the act. The communication used in this network runs the program in the fast and reliable mode [8]. A data communication scheme based on a safe and mobility-sensitive device and task scheduling schemes assisted by mobile drones in MANET has been the subject of in-depth studies in bibliographical studies [9–11]. Based on MANET, these studies proposed first come first serve (FCFS), Opportunistic Load Balancing, and min Max schemes to solve the problem of unloading and scheduling with security constraints. The Software-Defined Network helps to distribute to compute node services geographically. The knowledge task of distributed fog node services supporting mobility is based on the Software-Defined Network scheduling schemes studied by this work [12–15].

The primary purpose is to deploy road unit computing and communication nodes for vehicle applications in MANET. The goal was to minimize the latency and response time of user-level applications. However, the above studies have individually addressed the offloading or scheduling of IoT applications in their considered issues. In addition, the problems of joint unloading and planning are analyzed [16–19] with security constraints in the onboard vehicle network. Methods based on linear simulated annealing search, local search, and the standard and various earliest finishes are suggested. However, some methods provide centralized security for particular nodes (for example, on local nodes and computing/communication nodes), regardless of the overall network situation.

Power allocation is sent in Edge Computing systems that have many jobs done independently. However, a low-complexity sub-optimal method [20] was devised that optimize the task offloading scheduling and

resource allocation. The algorithm reduced the weighted sum of the device’s execution delay and energy consumption, which leads to minimizing the energy consumption and execution latency simultaneously.

The above studies have looked into multi-objective optimization for offloading the computing tasks in IoV for Edge Computing. Still, load balancing is considered an issue while considering both time and energy consumption. The proposed MOS-MOIO algorithm has a precise global search capacity to solve the Multi-optimization problem. As a result, this paper presents a method for data offloading based on the MOS-MOIO approach. Thus, the Immune method is an intellectual exploration method stimulated by the natural immune method. For optimizing the issues, the immune method is used with antigens. Every antibody in the immune method gives a possible outcome.

3 Proposed MOS with Multi-Objective Immune Optimization for Data Offloading

In recent days, IoV has been used among vehicles. It collects the data from the vehicles, and then it is monitored. In IoV systems, data offloading plays a significant role [21]. In the existing systems, data offloading is very challenging. The proposed method helps to reduce the offloading of data energy consumption and provides security to the data. Fig. 1 shows the architecture model of MOS with Multi-Objective Immune optimization.

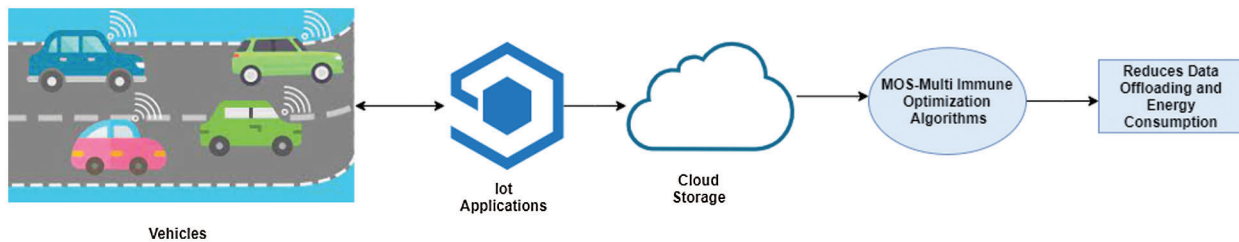


Figure 1: Architecture model of MOS with a multi-objective immune optimization algorithm

IoT devices help to collect the data with number of sensors. Then the data are stored in the cloud, in this stage huge collection of data was given to the cloud, therefore the data offloading is caused. To reduce this, Mobility aware offloading scheme (MOS) is used. The architecture model Fig. 1 shows that MOS with a multi-objective immune optimization algorithm efficiently reduces the data offloading. It gives security mobility to the system. Multi-Objective Immune optimization algorithm reduces the delay of the mobile user; reduces the server’s energy consumption. Therefore, the user can easily access the data by mobile applications.

3.1 MOS

$$T_o = \sum_{i \in A} \sum_{k=1}^k \alpha_{A,I}.(DS_{A,I}/b_a + ET_{A,I}^c + t_b) \leq \sum_{a=1}^a \sum_{i=1}^n D_{A,I} \quad (1)$$

$$T'_o = \sum_{i \in A} \sum_{k=1}^k \alpha_{A,I}.(DS_{A,I}/b_a + ET_{A,I}^c + t_b) + 1 - \alpha_{A,I}.(DS'_{A,I}/b'_a + (1 - ET_{A,I}^c) + t_b) \leq \sum_{a=1}^a \sum_{i=1}^n D_{A,I} \quad (2)$$

The data is transferred between various nodes during mobility, which is cared for by mobility aware offloading scheme (MOS). In the computing node, the task is scheduled. Before encryption, the hash data is computed with task i and node K1. When task i is moved to another node, K2, the K2 decrypts the data. The process continues till it reaches the optimal solution. To offload the task, the online offloading

engine takes a proper decision. It is also a part of the encryption and decryption process for security in Eqs. (1) and (2).

Algorithm 1: Mobility task offloading.

Input: i, a, K \ Number of tasks, applications, and Nodes

Output: optimized node.

```

for each ( $i \in a \in A$  as  $N$  &  $K \in k$ ) do
  If( $T_o = True$ ) then
     $\alpha_{A,I} = 1$ ; \ Apply encryption to the task
     $BCH [T_o, S_i] \leftarrow hash(S_i, I_A)$ ; \ BCH stores the hash values
    Calculate the costs of the system by using Eq. (1)
     $BCH [T_o, S_i] \leftarrow hash(S_i, I_k)$ ;
    P1 is optimized; \ partially initialized
  If( $1 - T'_o$ ) then
    Calculate the costs of the system by using Eq. (2)
     $BCH [1 - T'_o, S_i, K] \leftarrow K$ ;
    P1 is optimized; \ partially initialized
  If( $K \in k \in 0$ ) then
     $K = 0$  \ crashed nodes;
     $K = 1$  \ Stable nodes;
     $(S_i, K1) \leftarrow K2$ ; \ decrypt the task data and store it in the nodes
end-loop

```

Algorithm 1 first depicts that the task data is decrypted and then re-calculates the hash data. After that, it is compared with hash value and stored in the cloud. Then the offloaded task data is taken from the sender. The task is calculated using Eqs. (1) and (2). Tab. 1 provides details about the parameters of Eqs. (1) and (2).

Table 1: Description and parameters used in the equation

Parameters	Description
A	Number of applications
I	Tasks
k	Number of hybrid nodes
$DS_{A,I}$	Data size
$D_{A,I}$	Deadline
b_a	Bandwidth of network
$ET_{A,I}^c$	Execution time

3.2 Model of Time Consumption

The user can face unloading delay issues; it is one of the important criteria. It depends on the size of the unloading delay. The upload delay can be given in Eq. (3) as:

$$TA_{i\text{off}} = c_i / \theta v_2 e \quad (3)$$

Thus $v_2 e$ is used to move byte rate from vehicle to V2E. Task processing delay is evaluated by the following Eq. (3):

$$TA_{i\text{Exe}} = z_i / \frac{Z_j}{v * u_j} \quad (4)$$

The delay of processing of task is $task_i$ shown in the Eq. (4).

Algorithm 2: MOS with Multi-Objective immune optimization

Input: $|C|$, $|VE|$, np , G_{\max} , P , M , Z_j , C_j

Output: Possible outcome l , e

begin

C , $n \leftarrow$ Initialize tasks to server

$pop(0) \leftarrow$ assigns values to the population

verify the server system steps by using Algorithm 1

$Pop(1) \leftarrow$ Perform the process of Non-dominant sort

for $i = 1$ to G_{\max} do

$q(i) \leftarrow$ Choose the process($Pop(i)$, $np/2$)

$t(i) \leftarrow$ Immune process $q(i)$ by using Algorithm 2

$p(i+1) \leftarrow$ Finish the size of the population np

Process the functions

$r(i) \leftarrow (p(i+1)) \setminus$ non-dominant sorting to the process

end for

End begin

Algorithm 2 shows the process of the algorithm. Initially, the population is allocated, and using constraints helps compute the fitness worth. The algorithm works with the non-dominated sorting genetic algorithm III (NSGA-III), producing a possible outcome.

4 Experimental Evaluation

In this section, a series of complex simulations and tests are carried out to assess the performance of the proposed MOS-MOIO approach. The simulation process is explored with the help of Parameter settings for simulation and the statement of the relative offloading techniques. The proposed MOS-MOIO approach is evaluated with the help of various vehicle scales on the metrics of load balancing, execution cost, execution time, and energy consumption of the relative offloading techniques.

The five different vehicle scales are used in our simulation, with the number of vehicles set to 30, 60, 90, and 120. In parallel to our MOS-MOIO method, we use additional fundamental offloading techniques such as

MABOS and MOIA to conduct the comparison study. The Mobility Aware Blockchain Enabled offloading scheme (MABOS) tackled blockchain-enabled and motion-aware cross task offloading and scheduling problems with a vehicular cloud network. It cuts the price of communication and processing in vehicle applications. During formulation, the problem constraints, such as network capacity, job deadlines, and node resource load, are considered. The terminals are heterogeneous, decentralized fog devices and a centralized cloud to handle user-generated queries. The multi-objective immune algorithm (MOIA) established a vehicle-to-vehicle optimization model to handle the job unloading target problem whenever the vehicle drove all over the region. The server energy demand and load balancing were optimized to improve minimum delay. It made multi-objective optimization of unloading latency, server energy demand, and load attempting to balance feasible.

The task is first offloaded to the nearest edge computing devices in most scenarios. If the task to be offloaded requires more resources than the current devices provide, it is offloaded through the algorithm to find a minimum path on edge devices close to the present one. This method is repeated until the devices are offloaded of all tasks. On a personal computer (PC) with two AMD Ryzen-7 3700U 2.3 to 4 GHz processors and 4 GB RAM, the methods are implemented using CloudSim simulation tools. In the succeeding sections, the evaluation outcomes are displayed in detail.

4.1 Result Evaluation of Execution Cost

Execution cost is based on the number of tasks and the number of seconds to complete the task. Initially, the data are offloaded to the server. It executes each task. In the proposed system, it is minimized. Fig. 2 shows the evaluation of execution cost. The proposed method takes 58.25 s to execute 700 tasks. Compared with the existing system, the proposed method efficiently executes the tasks, and the execution cost is reduced by using an intellectual exploration ability of the Immune method.

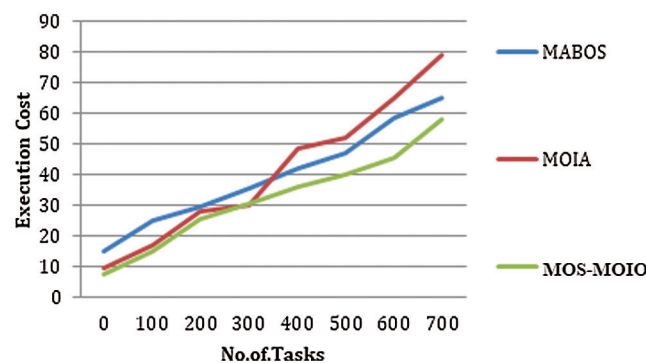


Figure 2: Evaluation of execution cost

4.2 Evaluation of Unloading Delays

In each task, the unloading delay is a significant constraint. It differs according to the number of Vehicles and number of tasks. There is a transmission delay in the existing systems; therefore, to process the task, there is a delay. The proposed methods help to reduce the unloading delays. Using MOS with a multi-objective immune optimization algorithm efficiently reduces the unloading delays and calculates the execution, time of response, offloading time, and transmission time.

Fig. 3 shows the execution time evaluation using the number of vehicles; 150 vehicles are used for evaluation, and the execution time is reduced.

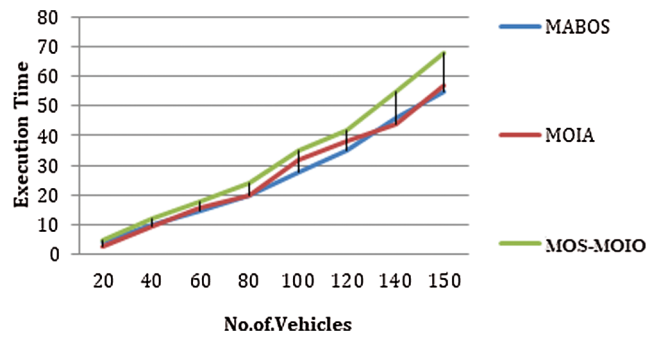


Figure 3: Evaluation of time of execution

4.3 Evaluation of Resource Utilization

The virtual machine (VM) instances in each device are occupied after offloading all tasks to the edge devices using computational offloading methods. Fig. 4 shows the resource utilization of edge devices with different vehicle scales with MABOS, MOIA, and MOS-MOIO. The number of working edge devices and the number of occupied VM instances in each edge device are used to calculate resource consumption. The resource consumption increases when fewer employed devices and more occupied VM instances. Our suggested approach MOS-MOIO achieves greater and steadier resource usage, as shown in Fig. 4. MOS-MOIO uses fewer resources and minimizes the number of idle VM instances than other offloading techniques.

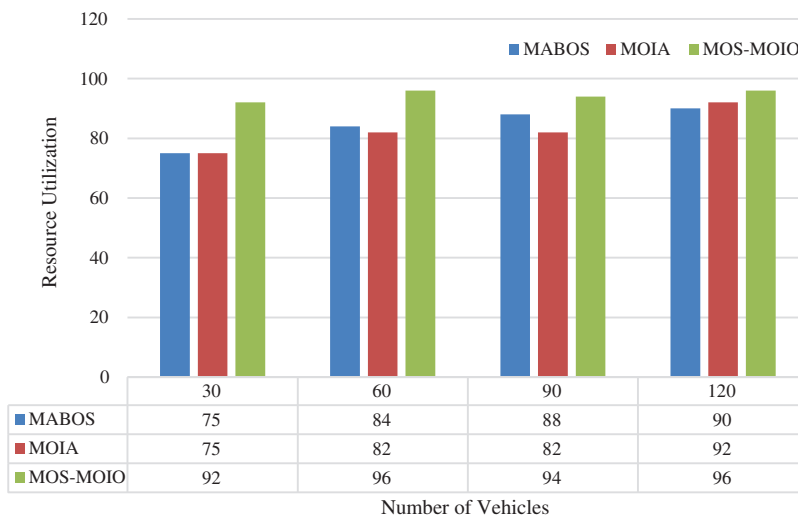


Figure 4: Evaluation of resource utilization

4.4 Evaluation of Load Balancing

The load balancing efficiency is compared in Fig. 5 to avoid the circumstance where some edge devices degrade due to overload or some resources are squandered due to underload. It demonstrates that our proposed approach, MOS-MOIO works better at load balancing at a lower rate. In contrast to the MABOS and MOIA schemes, the load balancing capability of the MOS-MOIO scheme improves as the number of vehicles increases. The load balancing rate, on the other hand, rises as the vehicle scales grow larger, implying that load balancing becomes more difficult as the number of vehicles rises. In

comparison to MABOS and MOIA, the suggested MOS-MOIO technique spreads jobs more uniformly across servers.

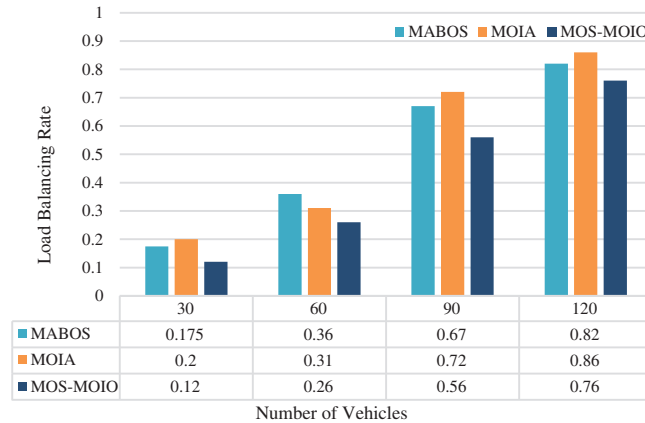


Figure 5: Evaluation of load balancing

4.5 Evaluation of Energy Consumption

We analyze three distinct offloading mechanisms for energy consumption efficiency with varying vehicle scales in Fig. 6. MOS-MOIO uses less energy than other offloading methods, as illustrated in Fig. 6 since it consumes less energy from idle VM instances. This indicates that MOS-MOIO wastes fewer resources than other offloading methods. All offloading strategies increase the energy consumption of active VM instances as the number of tasks rises.

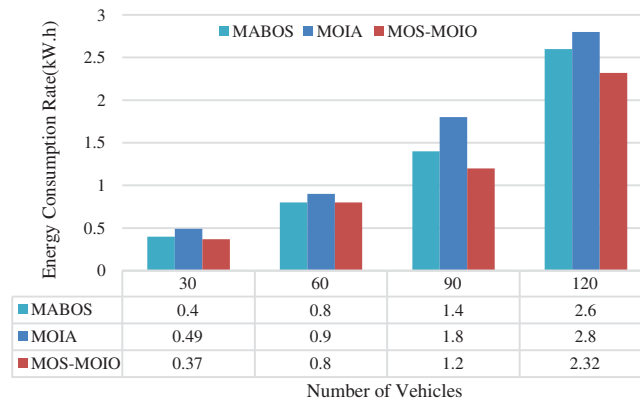


Figure 6: Evaluation of energy consumption

5 Conclusion

The Internet of Vehicles (IoV) is a promising technology; it is widely used nowadays. In this technology, minimizing its cost and reducing the data offloading is more complicated. The existing methods do not provide a resourceful data offloading. This proposed methodology, MOS with Multi-Objective Immune Optimization algorithm, minimizes the data offloading efficiently. Also, it reduces the cost of execution, unloading delays, and energy utilization. Cloud storage is used to enhance the security of the data; therefore, data security is increased. The proposed method executes all the tasks with a low execution

cost. It attains 58.25 s for executing the tasks. The machine learning technology will give more efficient data offloading in future research. It gives a more user-friendly environment to the user.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] S. N. Atluri and S. Shen, "Global weak forms, weighted residuals, finite elements, boundary elements & local weak forms," in *The Meshless Local Petrov-Galerkin (MLPG) Method*, 1st ed., vol. 1. Henderson, NV, USA: Tech Science Press, pp. 15–64, 2004.
- [2] A. Lakhan, M. Ahmad, M. Bilal, A. Jolfaei and R. M. Mehmood, "Mobility aware blockchain enabled offloading and scheduling in vehicular fog cloud computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4212–4223, 2021.
- [3] Z. Si-feng, C. Jiang-hao, S. En-lin and Z. Qing-hua, "Multi-objective optimized immune algorithm for computing offloading problem in edge computing scenes of internet of vehicles," *Wireless Personal Communications*, 2021. <https://doi.org/10.21203/rs.3.rs-656949/v1>.
- [4] H. Sun, H. Yu and G. Fan, "Contract-based resource sharing for time effective task scheduling in fog-cloud environment," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 1040–1053, 2020.
- [5] R. M. Abdelmoneem, A. Benslimane and E. Shaaban, "Mobility-aware task scheduling in cloud-fog IoT-based healthcare architectures," *Computer Networks*, vol. 179, pp. 1–17, 2020.
- [6] X. Zhao and C. Huang, "Microservice based computational offloading framework and cost efficient task scheduling algorithm in heterogeneous fog cloud network," *IEEE Access*, vol. 8, pp. 56680–56694, 2020.
- [7] M. Huang, Q. Zhai, Y. Chen, S. Feng and F. Shu, "Multi-objective whale optimization algorithm for computation offloading optimization in mobile edge computing," *Sensors*, vol. 21, no. 8, pp. 2628, 2021.
- [8] A. Lakhan and X. Li, "Transient fault aware application partitioning computational offloading algorithm in microservices based mobile cloudlet networks," *Computing*, vol. 102, no. 1, pp. 105–139, 2020.
- [9] M. Usman, M. A. Jan, X. He and P. Nanda, "QASEC: A secured data communication scheme for mobile Ad-hoc networks," *Future Generation Computer Systems*, vol. 109, pp. 604–610, 2020.
- [10] K. Kumar, S. Kumar, O. Kaiwartya, P. K. Kashyap, J. Lloret *et al.*, "Drone assisted flying ad-hoc networks: Mobility and service oriented modeling using neuro-fuzzy," *Ad Hoc Networks*, vol. 106, pp. 102242, 2020.
- [11] A. Agarwal, R. Pal and A. Prakash, "A scheduling algorithm including deadline of messages in vehicular ad hoc network," in *Proc. AVCSF*, Springer, Allahabad, Prayagraj, India, pp. 115–123, 2020.
- [12] B. Ko, K. Liu, S. H. Son and K. -J. Park, "RSU-Assisted adaptive scheduling for vehicle-to-vehicle data sharing in bidirectional road scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 977–989, 2020.
- [13] R. Vatambeti, "A novel wolf based trust accumulation approach for preventing the malicious activities in mobile ad hoc network," *Wireless Personal Communications*, vol. 113, no. 4, pp. 2141–2166, 2020.
- [14] C. Lin, G. Han, X. Qi, M. Guizani and L. Shu, "A distributed mobile fog computing scheme for mobile delay-sensitive applications in SDN-enabled vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5481–5493, 2020.
- [15] C. Tang, X. Wei, C. Zhu, Y. Wang and W. Jia, "Mobile vehicles as fog nodes for latency optimization in smart cities," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9364–9375, 2020.
- [16] M. Zhu, Y. Hou, X. Tao, T. Sui and L. Gao, "Joint optimal allocation of wireless resource and MEC computation capability in vehicular network," in *Proc. of 2020 IEEE Wireless Communications and Networking Conf. Workshops (WCNCW)*, Seoul, Korea (South), pp. 1–6, 2020.

- [17] Y. Wu, J. Wu, L. Chen, G. Zhou and J. Yan, "Fog computing model and efficient algorithms for directional vehicle mobility in vehicular network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2599–2614, 2020.
- [18] K. Liu, K. Xiao, P. Dai, V. Lee, S. Guo *et al.*, "Fog computing empowered data dissemination in software defined heterogeneous vanets," *IEEE Transactions on Mobile Computing*, vol. 20, pp. 3181–3193, 2020.
- [19] Q. Yuan, J. Li, H. Zhou, T. Lin, G. Luo *et al.*, "A joint service migration and mobility optimization approach for vehicular edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 9041–9052, 2020.
- [20] A. A. Al-Habob, O. A. Dobre, A. G. Armada and S. Muhaidat, "Task scheduling for mobile edge computing using genetic algorithm and conflict graphs," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8805–8819, 2020.
- [21] Y. Mao, J. Zhang and K. B. Letaief, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," in *2017 IEEE Wireless Communications and Networking Conf. (WCNC)*, San Francisco, USA, pp. 1–6, 2017.