Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

SOFE: SDN-based Task Offloading Technique for Fog-IoEV Network

Ahmed Jawad Kadhim Al-Shaibany
Ministry of Education, General Directorate for Education in Al- Qadisiyah
Jaber Ibrahim Naser
Ministry of Education, General Directorate for Education in Al- Qadisiyah
Mustafa Noaman Kadhim
University of Al-Qadisiyah
Dhulfiqar Zoltán Aalwahab
alwahab.zoltan@nik.uni-obuda.hu

Obuda University - John von Neumann Faculty of Informatics

Research Article

Keywords: Offloading Technique, IoEV, Fog Computing, SDN, Parked Electric Vehicles, Deadline

Posted Date: September 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9802073/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

SOFE: SDN-based Task Offloading Technique for Fog-IoEV Network

Ahmed Jawad Kadhim Al-Shaibany[1], Jaber Ibrahim Naser[1], Mustafa Noaman Kadhim[3] and Dhulfiqar Zoltán Aalwahab [3, *]

[1] Ministry of Education, General Directorate for Education in Al- Qadisiyah, Al- Qadisiyah, Iraq
ahmed.kadhim@mail.um.ac.ir, jaberalidami@gmail.com

[2] College of Computer Science and Information Technology, University of Al-Qadisiyah, Al Diwaniyah, 58001, Iraq. mustafa.noaman@qu.edu.iq

[3] Obuda University - John von Neumann Faculty of Informatics, 1034 Budapest, Bécsi út 96/b, Hungary.
alwahab.zoltan@nik.uni-obuda.hu

Abstract: Internet of Electric Vehicles (IoEV) is the next-generation version of Internet of Vehicles (IoV) networks. It is created by connecting electric vehicles using the Internet. The IoEV vehicles equipped with sensors and devices generate various types of tasks that must be completed within a predefined time interval. Unfortunately, due to the resource and energy limitations of electric vehicles, some of them fail to execute and perform all the storage and processing tasks. Thus, they are forced to find alternative processing nodes with available resources. The nearest fog nodes and parked vehicles appear to be the best choice. However, if the selected fog node is overloaded, new challenges will occur, and an additional delay will be added to the task completion time. Moreover, if the selected parked vehicle has a low level of residual energy, problems related to delay and the vehicle's lifetime will appear. In this paper, a new SDN-based fog-IoEV architecture and an SDN-based task Offloading technique for Fog-IoEV (SOFE) are proposed. SOFE aims to prolong the lifetime of the electric vehicles and reduce the response time by offloading the task to a suitable processing node (fog node or parked vehicle). It depends on a new binary linear programming model that takes into account the load of the processing nodes and the movement direction of the client vehicle. Moreover, it focuses on the constraints of the deadline, the user approval, and the residual energy. To evaluate the performance of SOFE, a simulation is performed. The results showed that SOFE outperforms RF-BinBRO-DARL and QAFT-SDVN in terms of average response time, percentage of tasks that meet the deadline, and number of dead electric vehicles with varying numbers of parked electric vehicles and tasks.

Keywords: Offloading Technique, IoEV, Fog Computing, SDN, Parked Electric Vehicles, Deadline.

1. Introduction

An electric vehicle is a new type of transportation mode. The development in the battery industry has prompted many governments to adopt the idea of electric vehicles which reduce the use of fossil fuels and the emission of carbon dioxide [1, 2]. Like fuel vehicles, the recent versions of electric vehicles have connection equipment (i.e., On-Board Units). With the advent of real-time computing and wireless communication techniques, vehicles are becoming more intelligent. Connecting the electric vehicles and roadside units (RSUs) creates a new type of vehicular network called Electrical Vehicular Ad hoc Network (E-VANET) [3]. Adding the Internet as a connection mode and setting a set of sensors on the electric vehicles generates an Internet of Electric Vehicles (IoEV) network [4].

The vehicles and their sensors in IoEV monitor roads, accidents, weather conditions, passengers, etc., and generate various types of requests that differ in type, time sensitivity, volume, etc., to make traveling safer and comfortable [4]. Such requests need different types of processing and storage services and consume an amount of energy to be completed within a time period (i.e., deadline). The electric vehicles can provide services for some of them. Unfortunately, they cannot respond to all requests due to the limitations of energy, storage, and processing capabilities. A popular solution is to transmit requests and tasks to cloud computing which utilizes high-quality servers and can process all tasks efficiently. However, sending large volume tasks to the remote cloud servers consumes a lot of power from the electric vehicle's battery. Moreover, bandwidth limitations, network collisions, and long distances add additional delay to the response time. All that makes cloud computing not the best solution for IoEV [5]. To overcome the issues of conventional cloud computing, fog computing represents a magic solution [6].

Fog computing is a new paradigm of distributed systems produced by Cisco in 2015 to move the services of cloud computing to the network edge. It aims to bring the storage, networking, data, and processing services closer to the end devices such as vehicles, sensors, and Internet of Things devices [7]. In fog computing, different services and functions are hosted at the end devices such as routers, servers, access points, RSUs, etc. It can support the mobility and enhance security, data analysis, and quality of services (QoS) represented by reducing the service latency, and bandwidth consumption [8]. Using the principles of fog computing systems within the vehicular network is known as

vehicular fog computing (VFC) [9]. In VFC, the moving and parked vehicles represent assistant fog nodes, and their resources provide networking and processing services [10].
In IoEV, sensors, vehicles, drivers, and passengers generate different requests and tasks that need responses. As it is known, electric vehicles have limited energy and computation resources, and as a result, they cannot respond to all tasks. Thus, some tasks are sent to the closest fog node to execute them with low service time. However, there is no guarantee that the closest fog node (or as it is called, the source fog node in this paper) is always available or under-loaded due to its limited capabilities and waiting tasks in its queue. Entering new tasks into the queue of the overloaded source fog node will increase the delay, and as a result, pass the deadlines of some tasks. On the other side, another fog node (i.e., far fog node) is idle or has better capabilities compared to the closest one. Thus, it can be utilized to produce a good service for the tasks. At this point, there are two questions. The first question is, what is the solution if all fog nodes are full? In this paper, we try to solve this issue by utilizing the concept of electric vehicular fog computing (EVFC) and using the resources of the parked vehicles which can provide a reasonable service for some tasks. However, not all parked electric vehicles can produce service due to energy limitations. Moreover, some owners of electric vehicles and due to the cost of battery recharging don't give permission to use the resources of their vehicles, or they give a limited permission to execute a number of tasks using the resources of their vehicles. The second question is how to select the best fog node or parked vehicle for each task. To determine the best one, we simply need a complete information of all available processing nodes (i.e., fog nodes and parked vehicles). moreover, we need an efficient offloading technique that exploits this knowledge to make a correct selection decision. Software-defined network (SDN) is a powerful approach to control the network with a high level of dynamicity, flexibility, and programmability compared to a traditional network. It uses one or more controllers to gather fresh information of the network such as mobile and fixed nodes, bandwidth, delay, data paths, resources, tasks, etc. [11]. Therefore, we produce a new offloading technique that exploits the strong points of the SDN controller to select the optimal processing node for each task.
The contributions of this paper are summarized as follows:

- Building a new SDN-Fog-IoEV architecture by merging concepts of SDN and EVFC with IoEV. It helps in collecting fresh information of the whole network and provide a global view to utilize the network resources efficiently to produce services for launched requests and tasks.
- Formulating the problem of selecting the best processing unit for each task as a mathematical optimization model. This model is in the form of binary linear programming to determine the processing node that responds within the deadline. Moreover, it focuses on the load status of the processing node (i.e., under-loaded or overloaded) and movement direction of the client vehicle (i.e., the vehicle that generated the task).
- Designing a new SDN-based task offloading technique. It benefits from the capabilities and complete knowledge of the SDN controller in solving the proposed mathematical optimization model to determine which fog node or parked electric vehicle is the optimum one for each task.
- Using residual energy and user approval constraints to determine how to utilize the resources of parked electric vehicles efficiently without emptying their battery or switching off these vehicles. The user approval (i.e., vehicle owner permission) constraint is used to determine the number of tasks that can be executed by each parked electric vehicle.

The organization of rest sections of this paper is as follows: section 2 shows a review of the related works in the SDN-Fog-IoV network, section 3 explains the system model and problem definition, section 4 illustrates the proposed offloading technique, section 5 displays simulation and results, and final section summarizes the conclusions and future directions.

2. Related Works

The utilization problem of fog computing resources was taken into account in a set of previous works. Moreover, several resource management schemes, task offloading policies, and optimization models were suggested to solve it. They can be categorized as follows:

2.1. Traditional Resource Management

A set of researchers in [12-17, 5] produced various resource management schemes to utilize the resources at the network edge to produce different services for IoV vehicles. Z. Zhou et al. [12] focused on the delay and inefficient resource utilization of VFC and proposed a task assignment and resource allocation technique based on contract theoretical modeling. First, they formulated the optimization problem of task assignment as a two-sided matching. After that, they used dynamic updated information and a pricing-based stable matching method to solve it. G. Li et al. [13] combined IoV with mobile edge computing to improve the response time of vehicles' requests. They proposed

a system composed of three tiers: vehicles, data service agents, and fog servers to manage the resources using a Stackelberg game model. Moreover, they applied a dynamic partitioning strategy to balance the workload and enhance QoS. Y. Zhang et al. [14] designed a deep reinforcement learning based deep Q-learning network to allocate the resources of edge computing to execute the tasks and requests launched from IoV vehicles with minimum latency and overhead. It takes the mobility speed of vehicles and the computing capabilities of edge nodes as constraints in the mathematical model used to solve the allocation problem. A.M. Farooqi et al. [15] proposed an efficient fog computing model to enhance QoS and reduce the response time of IoV critical applications. It depends on using a set of edge computing processing points connected using a 5G network to reduce the required time to respond to the requests received from the vehicles. H. Xi et al. [16] suggested a reinforcement learning policy to reduce the idle time of the resources of RSUs, cloud, and edge computing layers. They applied the caching concept and multi-layer allocation approach to produce services for the IoV requests. To do that, they transferred the problem of resource allocation to an optimization model and solved it using a two-layer deep Q network model. U. Awada et al. [17] presented a task offloading method that considers the available resource, delay, and vehicle mobility in IoV safety applications. They used a control layer to gather real-time information about the nearby roadside units and vehicles to make a selection decision. In addition, they proposed an intelligent multi-task scheduling policy that depends on the dependency of tasks to increase utilization of the available resources and reduce the processing time. A.J. Kadhim [5] produced an energy-aware clustering method for the Internet of Hybrid Vehicle (IoHV)-fog network to connect fog units and fuel and electric vehicles. Moreover, a new energy-based resource management technique was proposed. It plays a big role in balancing the load of processing requests generated by electric vehicles to the fuel vehicles and fog nodes to save the energy of electric vehicles.

2.2. Parked Vehicles as Processing Nodes

The parked vehicles were considered as fog processing units in several research papers [18-23]. A.J. Kadhim et al. [18] focused on how to distribute IoV tasks amongst the fog nodes fairly. To do that, they applied a load balance policy that works proactively in the local and global SDN controllers. Moreover, they used parked vehicles as secondary processing fog nodes to increase the utilization of the available resources as much as possible and enhance the ratio of meeting the deadline of critical IoV applications. A.A. Khadir et al. [19] proposed an offloading approach applied in the SDN controller to reduce the processing time by determining the optimal processing node (i.e., cloud server, fog node, moving or parked vehicle) for the offloaded task from the vehicles. The problem of selecting the best processing point is solved using a binary linear programming model which takes the load, processing cost, and service time into account. Q. Peng et al. [20] proposed a hybrid task assignment method in IoV that depends on a ternary search-based algorithm and a Stackelberg game model to reduce the delay and total cost of executing the tasks. They assumed that the parked vehicles have embedded processing and storage resources and can be utilized as edge execution units. X. Hu et al. [21] proposed a hybrid resource management scheme that consists of two stages which are balancing the workload and offloading the tasks to reduce the delay in mobile edge computing. The workloads are distributed amongst the edge computing nodes and a set of parked vehicles using the proposed load balancing technique. The selection of a suitable processing node for a workload is done using a dynamic game-based allocation approach. Y. Cheng et al. [22] combined a heuristic algorithm with deep adaptive reinforcement learning to analyze the load and status of fog computing nodes and study the behavior of IoV vehicles according to their mobility and parking. It aims to allocate the resources of fog computing and parked vehicles efficiently to improve the resource utilization and service rate of IoV applications. N. Saeed et al. [23] focused on the resource allocation problem in VFC. Therefore, they used particle swarm optimization, subspace minimization, and multi-objective grey wolf optimizer algorithms to allocate the processing and storage resources of slow-moving and parked vehicles to increase their utilization and reduce the processing delay.

2.3. SDN-based Resource Management

In addition to [18, 19], a set of SDN-based resource management approaches were proposed to enhance the performance of the fog-IoV environment. X. He et al. [24] improved the constrained optimization particle swarm optimization algorithm using the reverse of the flight of mutation particles and a linear decrease in inertia weight. After that, they applied their improved algorithm in the central SDN controller to distribute the IoV tasks among the fog nodes to enhance the latency and QoS. A.J. Kadhim et al. [25] proposed a hierarchical load balancing scheme applied in central and local SDN controllers to provide fog resources to serve real-time applications of the IoV network. To do that, they proposed a mathematical model with maximum resource utilization of fog nodes as the objective function and network bandwidth and task deadline as constraints. M. Ibrar et al. [26] presented an artificial intelligence-based task assignment and resource allocation for fog-IoV networks using a set of SDN controllers and agents. Their proposed scheme distributes the available fog processing units according to their load and processing

capabilities and uses a Markov decision model to assign tasks to the most appropriate one. It aims to decrease the delay and response time of sensitive tasks of IoV vehicles and utilize the fog computing resource in an efficient manner. U. Maan et al. [27] proposed an SDN-based offloading technique to select the best fog node for each request to increase the total performance. It works depending on deep Q network reinforcement learning and Kalman filter prediction approach. It predicts the next position of the vehicle that generated the request, and based on the available fog resources, it selects the best fog node. E. Qafzezi et al. [28] produced a new scheme to manage the network resources in Cloud-Fog-VANET efficiently using the strong points of SDN and artificial intelligence. It collects information about some parameters such as available resources, task size, application's requirement, and network conditions using the SDN controller and applies an integrated fuzzy logic system to assign the launched task to the most appropriate fog or cloud processing node. S.A. Syed et al. [29] applied the SDN concept in the fog computing node and used a central SDN controller to manage the whole network. The SDN controller assigns priority to the tasks and messages based on their type, size, and deadline and forwards them after classifying them into safety and non-safety. Moreover, they proposed heuristic algorithms to improve QoS and guarantee message delivery with a minimum ratio of failure. P. Li et al. [30] designed a hierarchical SDN framework to manage fog-IoV networks, utilize the resources, and distribute the computation requests efficiently. In this framework, the SDN controller applied a task offloading technique based on the Deep Q-Network model to distribute and balance the load of the edge computing nodes with minimum mean square deviation. S.A. Darade et al. [31] applied an integrated whale optimization policy on the fresh information of the SDN controller to balance the generated workload from IoV applications. Moreover, it reduces the delay and number of transmitted tasks to the remote cloud processing servers.
Table 1 shows a summary and comparison of the related works according to a set of parameters such as network type, exploitation of fog computing and parked vehicles, residual energy, user approval, allowed number of tasks that can be executed by the parked electric vehicles, deadline, etc. The next paragraph explains the limitations and challenges of the previous research works and why they are not appropriate for IoEV applications.
The above research works focused mostly on the QoS of IoV and how to utilize the resources of fog nodes and reduce their idle time as much as possible. However, they suffer from a set of problems and limitations that make them inappropriate for IoEV. First, they did not pay attention to the energy parameter and its effect on the performance of the proposed approaches in terms of the lifetime of vehicles. Second, some of them [12-17, 24-31] ignored the vast idle resources of parked vehicles which can be used as assistant computing nodes. Third, the authors of some research works assumed that the vehicles offload their tasks to the closest fog node when their processing capabilities are unable to respond to these tasks. But, if the closest fog node is overloaded at that moment, it will ignore these tasks or add them to the end of the waiting queue. Thus, the deadline of critical tasks will be passed. Fourth, the authors of [18-23] aimed to exploit the idle resource of parked vehicles in executing the increased number of tasks of other vehicles. But, they didn't pay any attention to the residual battery power and amount of consumed energy from the battery of parked vehicles during task processing. Moreover, they did not limit the number of tasks that can be processed by each parked vehicle depending on its residual energy and battery capacity. All these limitations mean that the previous load balance, task offloading, and resource allocation approaches are not perfect for IoEV and the limited possibilities of electric vehicles. This is the drive to propose a new suitable offloading strategy to save energy and increase the lifetime of IoEV electric vehicles.

Table 1: Summery of the related works (+ it is taken into account, - it is not taken into account)

Reference	Network type	Vehicle type	SDN-based?	Fog or edge computing	Parked vehicles	Load Status	Residual Energy	User approval	Turnaround time	Communication time	Deadline	Allowed tasks	Direction
[12]	IoV	Diesel	No	+	-	+	-	-	+	-	-	-	-
[13]	IoV	Diesel	No	+	-	+	-	-	+	+	-	-	+
[14]	IoV	Diesel	No	+	-	-	-	-	+	+	-	-	-
[15]	IoV	Diesel	No	+	-	+	-	-	+	-	-	-	-
[16]	IoV	Diesel	No	+	-	+	-	-	-	-	-	-	-
[17]	IoV	Diesel	No	+	+	+	-	-	+	-	-	-	-

[5]	IoHV	Diesel & Electric	No	+	-	+	+	-	-	-	-	-	-
[18]	IoV	Diesel	Yes	+	+	+	-	-	-	-	+	-	-
[19]	IoV	Diesel	Yes	+	+	-	-	-	+	+	+	-	-
[20]	IoV	Diesel	No	+	+	+	-	-	-	-	-	-	-
[21]	IoV	Diesel	No	+	+	+	-	-	+	+	-	-	-
[22]	IoV	Diesel	No	+	+	+	-	-	+	-	-	-	-
[23]	IoV	Diesel	No	+	+	+	-	-	-	-	+	-	-
[24]	IoV	Diesel	Yes	+	-	+	-	-	-	-	-	-	-
[25]	IoV	Diesel	Yes	+	-	+	-	-	-	-	+	-	-
[26]	IoV	Diesel	Yes	+	-	+	-	-	+	-	-	-	-
[27]	VANET	Diesel	Yes	+	-	+	-	-	-	+	-	-	-
[28]	VANET	Diesel	Yes	+	-	-	-	-	-	-	-	-	+
[29]	IoV	Diesel	Yes	+	-	-	-	-	+	+	+	-	-
[30]	IoV	Diesel	Yes	+	-	+	-	-	+	-	-	-	-
[31]	IoV	Diesel	Yes	+	-	+	-	-	-	-	-	-	-
Proposed	IoEV	Electric	Yes	+	+	+	+	+	+	+	+	+	+

3. System Model and Problem Formulation

This paper presents a system architecture (shown in Fig. 1) includes four layers as follows:

3.1. IoEV Layer: In addition to parked electric vehicles that found in parking lots, there are several electric vehicles moving at different velocities and directions in this layer. Each one, whether it is moving or parked, has a rechargeable battery and a set of smart devices like a camera and sensors that generate computation tasks and storage demands differ according to type, size, and importance level. Moreover, each electric vehicle has limited energy, computation resources, and storage memory. Such resources can be exploited to produce services for some types of tasks locally within the deadlines. Each electric vehicle represents a single processing node, and as a result, it can be modeled as M/M/1 queuing model with arrival rate λ, service time $1/\mu$, and service rate μ. Table 2 shows the symbols used in this work. The electric vehicle j can work as an assistant processing node and execute a task t if it can meet the following condition:

$$(R_j > \mathcal{E}_j) \wedge (\mathcal{S}_j < L_j) \wedge (T_j < \mathcal{D}_t) \ , \quad (1)$$

Where

$$\mathcal{E}_j = \beta_j * 0.20 \ , \quad (2)$$

$$\mathcal{S}_j = \sum_{t=1}^{n} S_t \ , \quad (3)$$

$$\mathrm{T}_j = \frac{\sum_{t=1}^{n}(P_t + W_t)}{n} \ , \quad (4)$$

This condition (equation 1) consists of three sub-conditions which are residual energy ($R_j > \mathcal{E}_j$), load status ($\mathcal{S}_j < L_j$), and deadline ($T_j < \mathcal{D}_t$). Therefore, the vehicle adds the new task to its waiting queue if it is under-loaded, has residual energy more than the energy threshold, and can produce service before the deadline.

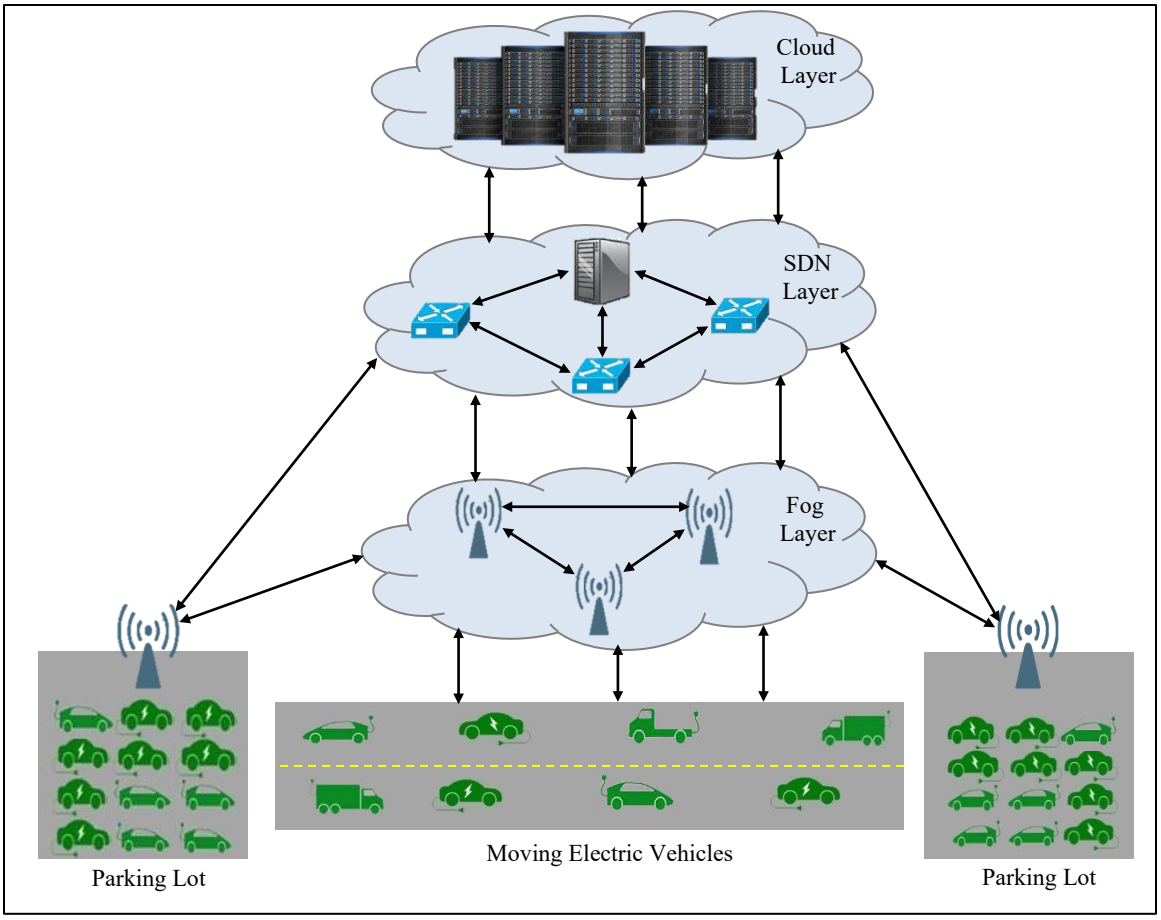


Fig. 1: System architecture

In this layer, the moving vehicles communicate during mobility with different types of nearby fixed nodes, which communicate with the SDN controller and can help in processing tasks. The moving vehicles in this paper can be used to execute only some of their tasks while the parked vehicles, in addition to their tasks, can be utilized to process some tasks of other parked and moving vehicles. The question is, why are the moving electric vehicles not appropriate for executing the tasks of other vehicles? The moving vehicle may move from one location to another during the processing operation of the tasks of other vehicles. Thus, it requires routing paths to deliver the response to those vehicles, and as a result, it must send route requests to the SDN controller which computes the best routes. This procedure consumes an amount of battery power and takes time, and as a result, adds additional delay which may affect the probability of meeting the deadlines. Moreover, to utilize the resources of moving electric vehicles, these vehicles must send periodic information to each fixed node and communicate with it during their mobility. This procedure also consumes a large amount of energy which is limited in electric vehicles.

Each set of parked vehicles found in a parking lot communicates with a fixed gateway. The parked vehicles represent assistant processing nodes and can be utilized to execute the tasks of other moving or parked electric vehicles (i.e., the vehicles that do not meet the above condition). However, exploiting the resources of parked electric vehicles without limitations drains the battery power. It is not acceptable to the electric vehicle owners. Thus, to exploit the resources of parked electric vehicles, we need user approval to save energy and prevent the dead situations of these vehicles. Each parked electric vehicle without a user approval cannot be used and does not send information to the gateway to prevent energy wastage. On the other side, the resources of the rest vehicles (i.e., parked electric vehicles that have the user's approval) can be utilized to process the tasks. Thus, they send information like battery capacity, residual energy, workload, average turnaround time, and workload threshold to the gateway. Based on residual energy and battery capacity of each vehicle, the gateway computes the allowed number of tasks that can be processed by that vehicle using equation (5). After that, the gateway transmits the collected information of parked electric vehicles to the SDN controller.

$$\Upsilon_j = \left\lfloor \frac{(R_j - \text{'E}_j) * \alpha}{r} \right\rfloor \qquad (5)$$

Table 2: Notations

Symbol	Definition
i, j	Indexes represent a fog node and a parked electric vehicle.
N	It is a set of all network nodes (i.e. vehicles, fog nodes, SDN controller, OpenFlow switches, and cloud servers)
R_j	The residual energy of an electric vehicle j.
E_j	The energy threshold of an electric vehicle j. In this paper, it represents 0.20 of the total battery capacity.
S_j	It is the total size of all processing tasks in the waiting queue of the electric vehicle j.
L_j	Load threshold of electric vehicle j.
T_j	Average turnaround time of electric vehicle j.
$\mathcal{D}_t$	It is the deadline of task t.
β_j	Total battery capacity of electric vehicle j.
S_t	It is the size of task t.
P_t	It is the processing time of task t.
W_t	It is the waiting time of task t in the queue.
Υ_j	The allowed number of tasks that can be processed by the parked electric vehicle j.
α	It is a weight that represents the percentage of available energy that can be used to execute the tasks. In this paper, its value is 0.15 of the available energy.
r	The average of the energy required to execute a task.
S_i	It is the total size of all processing tasks in the waiting queue of fog node i.
L_i	Load threshold of fog node i.
T_i	Average turnaround time of fog node i.
s	It is the source fog node. It is the closest fog node to the electric vehicle that generated the task.
C_{si}	It is the communication time from the source fog node s to the selected fog node i.
C_{sj}	It is the communication time from source fog node s to the selected parked electric vehicle j.
f_i , v_j	Variables with 0 and 1 values represent fog node i and parked electric vehicle j.
x_i , x_j	Variables with 0 and 1 values represent the load status of fog node i and parked electric vehicle j.
y_i , y_j	Variables with 0 and 1 values refer to the direction of the fog node i parked electric vehicle j, respectively, compared to the movement direction of client vehicle.
η_j	It is the number of offloaded tasks to the parked electric vehicle j.
R_t	It is the response time.
T_k	Average turnaround time of cloud node k.
C_{sk}	It is the communication time from source fog node s to the selected cloud server k.

3.2. Fog Layer: In addition to gateway nodes in the parking lots, there are different types of fog processing nodes such as routers, servers, roadside units, etc. at the network edge with a short distance to the electric vehicles. Each one has heterogeneous resources and covers a set of electric vehicles found in its domain. Each under-loaded fog node can provide processing and storage resources to the vehicles that are found in its coverage domain and the radio range of other fog nodes and execute the launched tasks from these vehicles. Moreover, each fog node has an agent to perform duties of the OpenFlow protocol to connect to the SDN controller in the next layer and send it different information about its workload, load threshold, and average turnaround time. It performs SDN controller's commands like offloading and executing the tasks, and working as an intermediate node to forward tasks and responses to other fog nodes. The fog node is considered a single server. Thus, it can be modeled as M/M/1 queuing model with arrival rate λ and service rate μ. The fog node i can execute a task t if it is not overloaded and its turnaround time is less than the deadline of that task, as shown in the following condition:

$$(\mathcal{S}_i < L_i) \wedge (T_i < \mathcal{D}_t)\,, \qquad (6)$$

3.3. SDN Layer: In this layer, there is a set of SDN switches (i.e., OpenFlow switches) and an SDN controller that works as a network leader to manage the whole network. The SDN controller has a complete updated view about the number and type of network nodes, available resources, residual energy of the parked vehicles, available bandwidth, and routing paths. It communicates with the nodes of the fog layer directly using the OpenFlow protocol and has connections with cloud nodes using a multi-hop connection mode. Also, it can communicate with moving and parked

vehicles un-directly via the fog nodes and gateways. Moreover, it can make the task offloading decision and configure the best route to transmit a task from a fog node to a fog or cloud node.

3.4. Cloud Layer: The nodes of this layer are better and faster than those of the fog layer. Each cloud node has large processing and storage resources and can execute time-sensitive tasks quickly. However, the long distance and constrained network bandwidth are the largest problems that increase the communication delay and lead to missing the deadline of some tasks. The resources of each cloud server is divided into a set of virtual machines, and each one represents a server and can be modeled using M/M/c with service rate μ of each virtual machine and total service rate $c\mu$ where c is the number of virtual machines.

In this paper, the architecture consists of a large number of nodes which are SDN controller, cloud nodes $K = \{1, 2, \ldots, k\}$, fog nodes $F = (1, 2, \ldots, n)$, moving electric vehicles $M = \{1, 2, \ldots, m\}$, parked electric vehicles $P = \{1, 2, \ldots, p\}$, and OpenFlow switches $O = \{1, 2, \ldots, o\}$. Selection of the processing unit is performed by the SDN controller which has complete knowledge of the network nodes and links. Moreover, it computes the routing path and communication time from the source fog node s to the selected processing node. To do that, the SDN controller builds an undirected graph with a set of nodes N and a set of links L and transforms the selection problem of optimal processing unit to a Binary-Linear Programming with two binary variables as follows:

$$Max \quad f_i\left((x_i + x_i y_i)\frac{1}{T_i + C_{si}}\right) + v_j\left((x_j + x_j y_j)\frac{1}{T_j + C_{sj}}\right) \quad \forall i \in F,\ j \in P \tag{7}$$

S.T.

$$f_i + v_j = 1, \quad \forall i \in F, \quad \forall j \in P, \tag{8}$$

$$f_i, v_j \in \{0,1\}, \quad \forall i \in F, \forall j \in P, \tag{9}$$

$$f_i (T_i + C_{si}) + v_j (T_j + C_{sj}) < \mathcal{D}_t, \forall i \in F, \forall j \in P, \tag{10}$$

$$v_j \eta_j < v_j \Upsilon_j, \quad \forall j \in P, \tag{11}$$

$$v_j R_j > v_j E_j, \quad \forall j \in P, \tag{12}$$

$$x_i, y_i \in \{0,1\}, \quad \forall i \in F, \tag{13}$$

$$x_j, y_j \in \{0,1\}, \quad \forall j \in P, \tag{14}$$

$$T_i, C_{si} > 0, \quad \forall i \in F, \tag{15}$$

$$T_j, C_{sj} > 0, \quad \forall j \in P, \tag{16}$$

The objective function (equation 7) is maximization and aims to select the best processing unit (fog node or parked vehicle) that is under-loaded and located in the same mobility direction of the source electric vehicle and has minimum turnaround time and communication time. We wrote $1/(T_i + C_{si})$ because the objective function is maximization and the minimum values of turnaround time and communication time are the best. The first constraint (equation 8) explains that one and only one node from fog nodes and parked vehicles is selected to execute a task. The second constraint (equation 9) indicates that the variable f_i takes the value 1 if the fog node i is selected and 0 if it is not selected. Moreover, the value of v_j is assigned in the same way. The third constraint (equation 10) is the deadline constraint. It explains that the turnaround time and communication time to send the tasks t from the source fog node to the selected processing node must be less than the deadline of that task. The fourth constraint (equation 11) refers to the number of offloaded tasks to parked electric vehicle j must be less than the allowed number of tasks that can be processed by the parked vehicle j. The fifth constraint (equation 12) indicates that the residual energy of the parked vehicle must be more than the energy threshold. The sixth constraint (equation 13) refers to the value of x_i is equal to 1 if the fog node i is not overloaded. Otherwise, its value is set to 0. Moreover, the value of y_i is equal to 1 if the fog node i is located in the same direction as the movement of the client vehicle. Otherwise, its value is set to 0. The seventh constraint (equation 14) explains that the value of x_j is equal to 1 if the parked electric vehicle j is not overloaded. Otherwise, its value is set to 0. Moreover, the value of y_j is equal to 1 if the parked electric vehicle j is located in the same direction as the movement of the client vehicle. Otherwise, its value is set to 0. The last two constraints (equations 15 and 16) show that the values of turnaround time and communication time must be more than 0.

Example:

In addition to the source fog node, there are 10 processing nodes with different IDs and capabilities as shown in Table 3. In this example, we assume that all parked vehicles have residual energy more than the energy threshold (fifth constraint (equation 12) is true) and the number of executed tasks is less than the threshold of allowed tasks (fourth constraint (equation 11) is true). Let's say there is a request to process a task with a deadline 7.3 seconds reached the SDN controller from the source fog node. The SDN controller must select the best processing node from fog nodes and parked vehicles. In this example, SDN applied the proposed model to the information in Table 3 and solved it using CPLEX. Moreover, we applied and solved the proposed model to this example manually. The processing nodes with IDs 55, 60, 91, and 99 are overloaded, and the nodes with IDs 75, 81, and 90 do not meet the task's deadline. The nodes with IDs 61, 62, and 95 can be selected to process the task. But, the best one with maximum solution of the objective function is the fog node (ID: 61) with average turnaround time and communication time equal to 3.6 and 2.5, respectively.

Table 3: Example's data

Processing node type	**ID**	**Load status (x_i or x_j)**	**Direction (y_i or y_j)**	**Turnaround time**	**Communication time**	**Solution of objective function (equation 7)**
Fog node	55	0	1	5.3	4.1	= 1 * ((0+ 0*1) * 1/ (5.3 + 4.1)) + (0) = 0
Parked vehicle	60	0	0	7.2	1.2	= 1* ((0+ 0*0) * 1/ (7.2 + 1.2)) + (0) = 0
Fog node	**61**	**1**	**1**	**3.6**	**2.5**	**= (0) + 1* ((1+ 1*1) * 1/ (3.6 + 2.5)) = 0.3278**
Fog node	62	1	0	2.9	3.4	= 1* ((1+ 1*0) * 1/ (2.9 + 3.4)) + (0) = 0.1587
Fog node	75	1	1	6.6	3.3	= 1* ((1+ 1*1) * 1/ (6.6 + 3.3)) + (0) Third constraint (equation 10) is false
Parked vehicle	81	1	0	5.2	11.6	= (0) + 1* (1+ 1*0) * 1/ (5.2 + 11.6)) Third constraint (equation 10) is false
Parked vehicle	90	1	1	4.7	2.8	= (0) + 1* ((1+ 1*1) * 1/ (4.7 + 2.8)) Third constraint (equation 10) is false
Fog node	91	0	0	6.0	1.2	= 1 * ((0+ 0*0) * 1/ (6.0 + 1.2)) + (0) = 0
Fog node	95	1	1	4.4	2.0	= 1 * ((1+ 1*1) * 1/ (4.4 + 2.0)) + (0) = 0.3125
Parked vehicle	99	0	1	3.6	2.7	= (0) + 1* ((0+ 0*1) * 1/ (3.6 + 2.7)) = 0

4. Proposed Offloading Technique

Electric vehicles have limited energy, and they cannot process the generated tasks when their residual energy reaches a certain level. Such tasks must be processed and their generators must receive results during a predefined time interval. Therefore, the electric vehicle can get an assistant to process them from the other processing nodes such as fog nodes and parked vehicles. To do that in an efficient way, a new offloading approach depends on SDN capabilities, and a new optimization model named SOFE is proposed. In SOFE, each fog node computes the average turnaround time every time period and determines its load status (under-loaded or overloaded). After that, it sends information related to average turnaround time and load status to the SDN controller periodically. In addition, the parked vehicles in each parking lot and have the user approval, compute their average turnaround time and load. After that, they send some information about the battery capacity, residual energy, workload, average turnaround time, and load threshold to the gateway which forwards it to the SDN controller.

As shown in Fig. 2, when an electric vehicle or one of its sensors generates a task, it tries to utilize its resources and capabilities to process that task. **Note**, the electric vehicle that generates a task is called the client vehicle in this paper. First, the client vehicle checks the conditions of its residual energy, load status, and deadline, and if these conditions are true, it will enter the task into the waiting queue and process it. Otherwise, it forwards the task to the closest fog node (CFN).

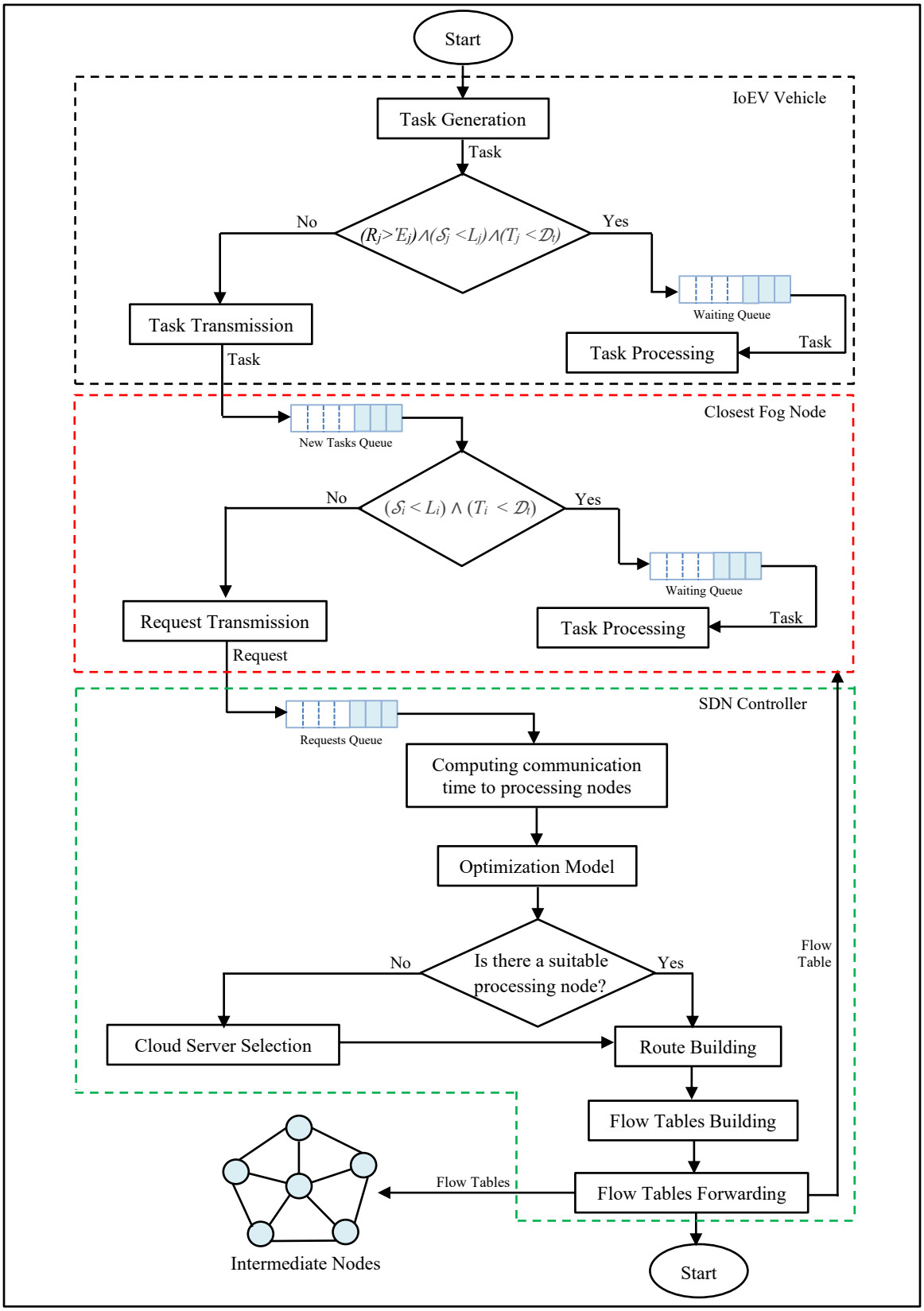


Fig. 2: The proposed SDN-based offloading approach

CFN also checks its capabilities like load status and deadline and compares them with the task requirements such as deadline and size using equation (6). If CFN is under-loaded and its average turnaround time is less than the task deadline, it adds the task to its waiting queue. Otherwise, it sends a request with information about the movement direction of the client vehicle and the task such as size and deadline to the SDN controller.

To select the best processing node, the SDN controller computes the communication time from CFN to each processing node depending on its global knowledge of the network. After that, it applies the proposed optimization model, and if there is a suitable processing node at the edge of the network, it will build the best route from CFN to the selected processing node using Dijkstra's algorithm with minimum delay. After that, it builds the flow tables and forwards them to the CFN and the inter mediate nodes to forward the data of that task to the selected processing node. Otherwise (If all processing nodes at the network edge are not able to execute the task due to overloading or deadline constraints), the SDN controller offloads that task to the cloud layer. In this case, the response time of task t includes the turnaround time (i.e., execution time and waiting time) of cloud server k and communication time from source fog node s to the cloud server C_{sk}. It can be defined as follows:

$$\acute{R}_t = T_K + C_{sk} \ , \tag{17}$$

The turnaround time of all cloud servers is very short due to their high capabilities. Thus, it is not an important factor and all focus should be on the communication time which is long due to the distance, network bandwidth, and network congestion. The communication time is more important and sensitive compared to turnaround time and must be reduced as much as possible to meet the deadline. Therefore, the SDN controller must select the closest cloud server and give the highest priority to how to find the best route to the selected cloud server among the available routes. At this point, the best route from the viewpoint of the SDN controller is one that has minimum delay. To do that, Dijkstra's algorithm with minimum delay is applied. This algorithm selects the shortest route with minimum total communication time. After that, the SDN controller builds the flow tables and forwards them to CFN and the intermediate nodes to send the data of that task to the selected cloud server.

5. Simulation and Results

In this paper, to build the proposed architecture and investigate the efficiency of SOFE, the network simulators SUMO 0.19.0 and OMNeT++ 5.0 are used. Moreover, they are used to compare performance results of SOFE with those of RF-BinBRO-DARL (Revised Fitness-based Binary Battle Royale Optimizer combined with Deep Adaptive Reinforcement Learning) [22], and QAFT-SDVN (QoS aware and fault tolerance-based SDN-IoV network) [29]. To do this comparison efficiently, percentage of tasks that meet the deadline (PTMD), average response time (ART), and the number of dead electric vehicles (DEV) are considered as evaluation metrics. In this paper, the electric vehicle with remaining energy less than 5 % is considered a dead vehicle. To increase the realism of simulation scenarios, the Baghdad city map is imported from openstreetmap website and added to SUMO. In the simulation environment, there are general parameters with fixed values across all simulation scenarios, as shown in Table 4. Moreover, there are special ones that have changed values in each scenario, as shown in the following:

5.1. Number of Parked Electric Vehicles

The idle resources of the parked electric vehicles can be utilized to process the tasks at the network edge. In this scenario, the effect of increasing the number of parked electric vehicles on the performance of SOFE, RF-BinBRO-DARL, and QAFT-SDVN according to ART, PTMD, and DEV is investigated. The number of parked electric vehicles in the environment of this scenario is 100, 200, 300, 400, and 500 with various battery capacities and residual energy levels. Moreover, 90% of the parked electric vehicles have user's approval and their resources can be used to execute the tasks. The number of launched tasks from electric vehicles is 300 with various volumes ranging from 25 to 250 MB and different deadlines ranging from 3 to 7 seconds.

Fig. 3 shows the values of ART using SOFE, RF-BinBRO-DARL, and QAFT-SDVN with increasing the number of parked electric vehicles. From this figure, we notice the effect of increasing the number of parked vehicles on reducing the response time. The reason for the response time reduction is increasing the available stable resources. Moreover, it shows that SOFE enhances ART by approximately 6.7% and 12.69% compared with RF-BinBRO-DARL and QAFT-SDVN, respectively, as the number of parked electric vehicles increases. In SOFE, some tasks are processed locally at the vehicle itself and the nearest fog nodes. Moreover, SOFE focused on selecting the best processing node that responds with minimum time by considering the load status, mobility direction, turnaround time, and communication time. In addition, employing the SDN concept in SOFE plays a very large role in selecting the processing nodes efficiently and computing the best routes quickly.

Fig. 4 explains the efficiency of SOFE, QAFT-SDVN, and RF-BinBRO-DARL in processing the tasks within their deadlines in a simulation environment with small and large numbers of parked electric vehicles. Increasing the number

of parked vehicles means increasing the available processing resources which run the tasks with low delay. We notice improvements in the results of PTMD of about 3.4% and 7.2% when increasing the number of parked vehicles, compared with RF-BinBRO-DARL and QAFT-SDVN, respectively. In this figure, we see that RF-BinBRO-DARL is better than QAFT-SDVN according to PTMD because it benefits from the resources of parked vehicles to process the tasks quickly. In addition to utilizing the parked vehicles, SOFE focuses on the deadline during selecting the processing node for each task. Moreover, the proposed optimization model gives a high priority to the processing nodes that are located in the same mobility direction as the client vehicle. This procedure reduces the communication time and increases the percentage of tasks that are executed before their deadlines. Thus, the PTMD of SOFE is better than that of RF-BinBRO-DARL.

Table 4: General parameters

Parameter	**Value**
Computer Type	HP
Random access memory	16.0 GB
CPU	13th Gen Intel(R) Core(TM) i7-1355U, 1.70 GHz
Operating system	Windows 11
Simulators	SUMO and OMNeT
Map	Baghdad city
Offloading techniques	SOFE, RF-BinBRO-DARL, and QAFT-SDVN
SDN controller	POX
Number of SDN switches	15
Number of cloud servers	5
Number of fog nodes	30
Number of parking lots	10
Number of gateways	10
Number of moving vehicles	2500
Average velocity of moving vehicles	20m/s
Battery capacity of electric vehicles	Different (ranging from 35-75 kWh)
Remaining power of electric vehicles	Different (ranging from 25-95 %)
CPU frequency of each vehicle	10^9 cycles/second
CPU frequency of each fog node	2 * 10^9 cycles/second
CPU frequency of each cloud node	5 * 10^9 cycles/second
Memory of vehicles	Different (ranging from 128 MB to 1 GB)
Memory of fog nodes	Different (ranging from 1 to 5 GB)
Memory of each cloud node	1 TB
Task sizes	Different (ranging from 25 to 250 MB)
Iterations of simulation	15
Time of simulation	15 minutes

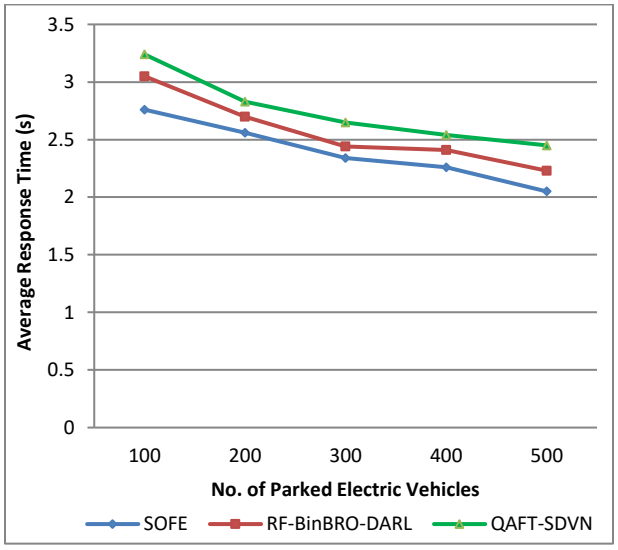


Fig. 3: Average response time

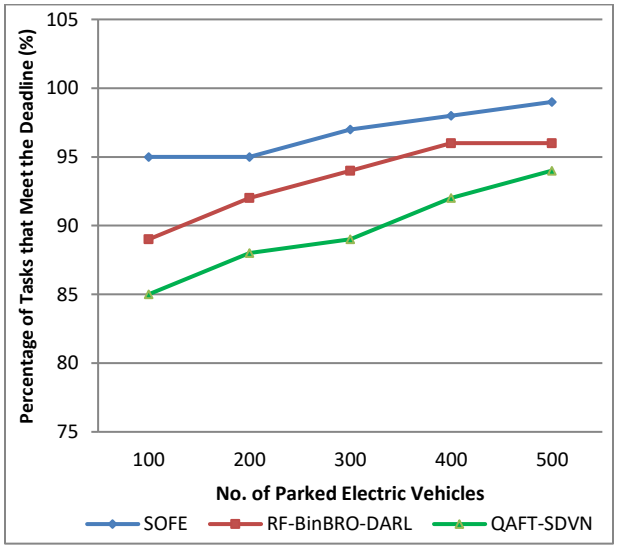


Fig. 4: Percentage of tasks that meet the deadline

Fig. 5 illustrates the effect of increasing the number of parked electric vehicles on reducing the power consumption and the number of dead electric vehicles. RF-BinBRO-DARL and QAFT-SDVN were proposed for IoV and did not pay any attention to the energy. Thus, when applying them in IoEV, they fail to produce good results in terms of DEV. On the other side, SOFE and the optimization mathematical model focus on the residual energy of the electric vehicles. It prevents the electric vehicles with low levels of remaining power from participating in executing the generated tasks. Moreover, they allow the parked vehicles to process a limited number of tasks depending on the user approval, battery capacity, and remaining power. This can reduce the number of dead electric vehicles. As a result, SOFE decreased the values of DEV by about 80.76% and 73.68% compared to RF-BinBRO-DARL and QAFT-SDVN, respectively, with different numbers parked vehicles.

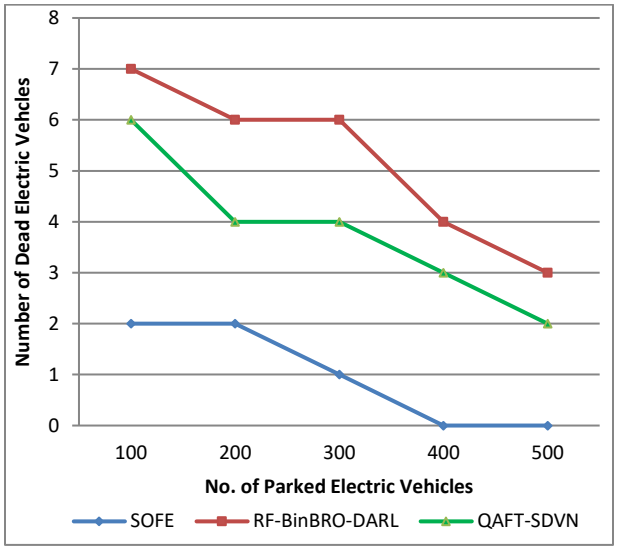


Fig. 5: Number of dead electric vehicles

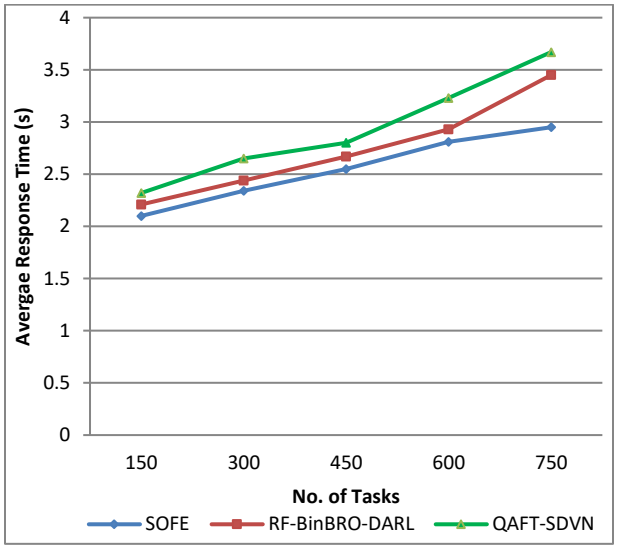


Fig. 6: Average response time

5.2. Number of Tasks

This scenario investigates and explains how the number of generated tasks affects the efficacy and performance of SOFE, RF-BinBRO-DARL, and QAFT-SDVN according to ART, PTMD, and DEV. The simulation environment of this scenario has 300 parked electric vehicles, and 90% of them have the user's approval. The number of launched tasks is 150, 300, 450, 600, and 750 with various volumes ranging from 25 to 250 MB and different deadlines ranging from 3 to 7 seconds.

Fig. 6 displays the values of ART with small and large numbers of generated tasks. With small numbers of tasks, the processing nodes respond with a reasonable response time because they are under-loaded and most resources are available. With increase the number of tasks, the available resources are reduced gradually. Therefore, the waiting time will be increased due to the overloaded cases of nearby processing units. However, SOFE improved ART by about 6.93% and 13.08%, compared with RF-BinBRO-DARL and QAFT-SDVN, respectively, as the number of tasks increased. SOFE performed good ART because it focuses on selecting the closest available processing nodes with minimum turnaround time. Moreover, it selects the processing node with low workload and located in the same movement direction as the client vehicle.

Fig. 7 shows the performance of SOFE, RF-BinBRO-DARL, and QAFT-SDVN in terms of PTMD with different numbers of tasks. As illustrated in this figure, increasing the number of tasks that are required to be processed within a predefined time interval affects the efficiency and performance of offloading techniques negatively. This is due to the increasing load and waiting time of the processing nodes, and thus, the deadline of some tasks will be passed. SOFE uses the knowledge of the SDN controller and focuses on the deadline during selection of the processing node for each task. Therefore, it increased the percentage of tasks that meet the deadline by about 6.2% and 11.6%, respectively, when compared with RF-BinBRO-DARL and QAFT-SDVN.

Processing the tasks consumes an amount of the vehicle's power. Increasing the number of tasks increases the consumed energy and number of dead vehicles when SOFE, RF-BinBRO-DARL, and QAFT-SDVN are used as shown in Fig. 8. However, SOFE performance isn't affected significantly with increasing the number of tasks because it limits the number of tasks that can be executed by the parked electric vehicles. In addition, the residual energy constraint is taken into account when assigning tasks to the parked electric vehicles. Thus, it improved the values of

DEV by approximately 77.77% and 69.23% with different numbers of tasks compared to RF-BinBRO-DARL and QAFT-SDVN, respectively. In this figure, we notice that the number of dead electric vehicles of RF-BinBRO-DARL is larger than that of SOFE and QAFT-SDVN because it executes some tasks using the parked vehicles without paying attention to the battery capacity and residual energy of electric vehicles. Moreover, it did not impose a limitation on the number of tasks that can be processed by the parked vehicles.

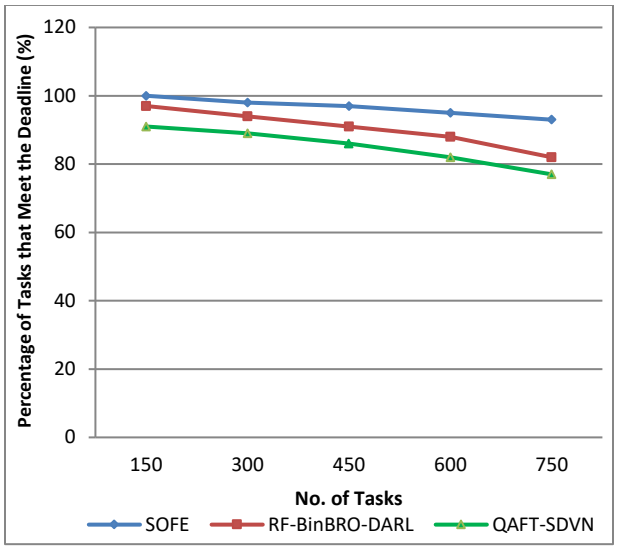


Fig. 7: Percentage of tasks that meet the deadline

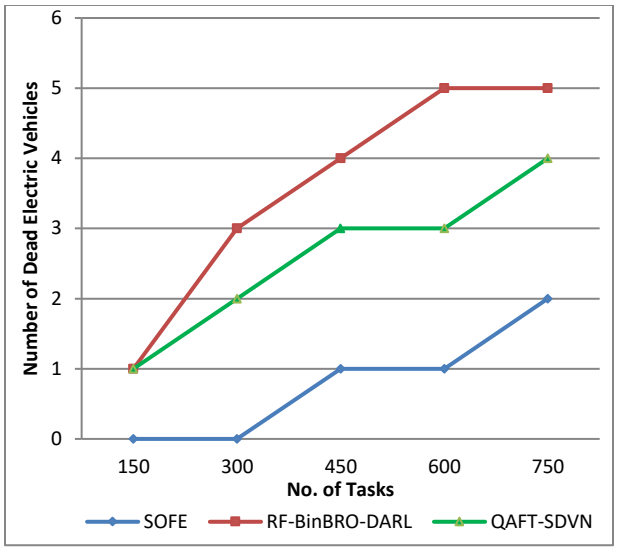


Fig. 8: Number of dead electric vehicles

6. Conclusions and Future Works

Energy limitation of electric vehicles is the biggest challenge in IoEV networks. Moreover, the vehicles have limited capabilities and fail to process all tasks and demands. Therefore, some tasks are transferred to the closest fog nodes to be processed. However, this solution is not optimal one especially when the closest fog nodes are overloaded. Thus, it is better to utilize the idle resources of parked electric vehicles to overcome this challenge. Again, some problems related to the energy consumption and vehicle lifetime will appear if there is no efficient resource management approach. In this paper, a new SDN-based offloading approach for IoEV was proposed. The selection problem of a suitable processing node was transformed into a binary optimization model that aims to select the optimal processing node that is under-loaded and located in the same mobility direction as the client vehicle. Moreover, it focuses on the average turnaround time and communication time to reduce the delay. In addition, it considers the deadline, residual energy, and allowed number of tasks that can be executed by parked electric vehicles as constraints to avoid the dead statuses of the parked vehicles. Compared with RF-BinBRO-DARL and QAFT-SDVN, the simulation results showed that SOFE enhanced ART by about 6.7% and 12.69%, increased the values of PTMD by about 3.4% and 7.2%, and reduced the values of DEV by approximately 80.76% and 73.68% with various numbers of parked vehicles. Moreover, SOFE improved ART by about 6.93% and 13.08%, increased the values of PTMD by approximately 6.2% and 11.6%, and decreased the values of DEV by about 77.77% and 69.23% with various numbers of tasks. The future plan of this research work can be summarized as follows:

- Using 5G technology in SDN-fog-IoEV to establish the communications among fog nodes layer and with SDN controller and OpenFlow switches to offload the data of IoEV tasks quickly.
- Designing an energy and mobility aware routing approach for IoEV to exchange the data among electric vehicles with a low amount of energy consumption and a high level of route stability.
- Designing a reinforcement learning approach to balance the load and allocate the resources of fog processing units and parked electric vehicles efficiently to reduce the idle time and improve the resource utilization.
- Producing a new proactive protection policy to detect and avoid the risks of some attacks that aim to consume the resources and energy of electric vehicles such as distributed denial of service.

Funding

This work is supported by Obuda University.

Author Contributions

A.J.K.A.-S., J.I.N., M.N.K., and D.Z.A. designed the study. A.J.K.A.-S. collected and processed the data. J.I.N. and M.N.K. implemented the proposed method and drafted the manuscript. D.Z.A. analyzed the results. All authors reviewed and approved the final manuscript.

Ethical approval

Not Applicable

Consent to Participate declaration

Not applicable.

Consent to Publish declaration

Not applicable.

Code availability

Not applicable (custom code)

Acknowledgment

This research was supported by Al-Qadisiyah University and Obuda University, whose assistance the authors deeply appreciate.

Conflict of Interest

The authors declare no conflicts of interest.

Data Availability

Not applicable

References

[1] D.B. Richardson, "Electric Vehicles and the Electric Grid: A Review of Modeling Approaches, Impacts, and Renewable Energy Integration," Renewable and Sustainable Energy Reviews, vol. 19, pp. 247–254, Mar. 2013.
[2] K. Mahmud, G.E. Town, S. Morsalin, and M.J. Hossain, "Integration of Electric Vehicles and Management in the Internet of Energy," Renewable and Sustainable Energy Reviews, vol. 82, no. 3, pp. 4179–4203, Feb. 2018.
[3] Y. Feng, Y. Huang, B. Li, H. Peng, J. Wang, and W. Zhou, "Connectivity Enhancement of E-VANET based on QL-mRSU Self-learning Energy-saving Algorithm," IEEE Access, vol. 11, pp. 3810–3825, Jan. 2023.
[4] A.J. Kadhim and J.I. Naser, "Toward Electrical Vehicular Ad Hoc Networks: E-VANET," Journal of Electrical Engineering and Technology, vol. 16, pp. 1667–1683, Mar. 2021.
[5] A.J. Kadhim, "An Energy Resource Management for Cluster based IoHV Supported by Fog Computing," Karbala International Journal of Modern Science, vol. 11, pp. 35–55, Dec. 2024.
[6] M.H. Alsharif, A. Jahid, R. Kannadasan, M.K. Singla, J. Gupta, K.S. Nisar, A. Abdel-Aty, and M. Kim, "Survey of Energy-Efficient Fog Computing: Techniques and Recent Advances," Energy Reports, vol. 13, pp. 1739–1763, Jun. 2025.
[7] Q. He, Z. Xi, Z. Feng, Y. Teng, L. Ma, and Y. Cai, "Telemedicine Monitoring System Based on Fog/Edge Computing: A Survey," IEEE Transactions on Services Computing, vol. 18, no. 1, pp. 479–498, Feb. 2025.
[8] W. C. Chuan, S. U. A. Laghari, S. Manickam, E. Ashraf, and S. Karuppayah, "Challenges and Opportunities in Fog Computing Scheduling: A Literature Review," IEEE Access, vol. 13, pp. 14702–14726, Jan. 2025.

[9] B. Hu, J. Du, J. Zhang, and X. Chu, “Computation Offloading and Resource Allocation in Mixed Cloud/Vehicular-fog Computing Systems,” IEEE Transactions on Mobile Computing, Early Access, pp. 1–13, Mar. 2025.
[10] N. Keshari, D. Singh, and A. K. Maurya, “A Survey on Vehicular Fog Computing: Current State-of-the-Art and Future Directions,” Vehicular Communications, vol. 38, pp. 1-16, Dec. 2022.
[11] A.J. Kadhim, S.A.H. Seno, J.I. Naser, and J. Hajipour, “DMPFS: Delay-Efficient Multicasting Based on Parked Vehicles, Fog Computing and SDN in Vehicular Networks,” Vehicular Communications, vol. 36, pp. 100–118, Oct. 2022.
[12] Z. Zhou, P. Liu, J. Feng, and Y. Zhang, “Computation Resource Allocation and Task Assignment Optimization in Vehicular Fog Computing: A Contract-Matching,” IEEE Transactions on Vehicular Technology, vol. 68, no. 4, pp. 3113–3125, Apr. 2019.
[13] G. Li, Y. Zhang, M. Wang, J. Wu, Q. Lin, and X. Sheng, “Resource Management Framework based on the Stacklberg Game in Vehicular Edge Computing,” Complexity, vol. 1, pp. 1-11, Jan. 2020.
[14] Y. Zhang, F. Li, and B. Li, “Computing Resource Allocation Scheme of IoV Using Deep Reinforcement Learning in Edge Computing Environment,” EURASIP Journal on Advances in Signal Processing, vol. 21, pp. 1–19, Jun. 2021.
[15] A.M. Farooqi, M.A. Alam, S.I. Hassan, and S.M. Idrees, “A Fog Computing Model for VANET to Reduce Latency and Delay Using 5G Network in Smart City Transportation,” Applied Sciences, vol. 12, no. 4, pp. 1–15, Feb. 2022.
[16] H. Xi and H. Sun, “Resource Allocation Strategy of Internet of Vehicles Using Reinforcement Learning,” Journal of Information Processing Systems, vol. 18, no. 3, pp. 443–456, Aug. 2022.
[17] U. Awada, J. Zhang, S. Chen, S. Li, and S. Yang, “Resource-Aware Multi-Task Offloading and Dependency-Aware Scheduling for Integrated Edge-Enabled IoV,” Journal of Systems Architecture, vol. 141, 130–143, Aug. 2023.
[18] A.J. Kadhim and J.I. Naser, “Proactive Load Balancing Mechanism for Fog Computing Supported by Parked Vehicles in IoV-SDN,” China Communications, vol. 18, no. 2, pp. 271–289, Feb. 2021
[19] A.A. Khadir and S.A.H. Seno, “SDN-based Offloading Policy to Reduce the Delay in Fog-Vehicular Networks,” Peer-to-Peer Network Applications, vol. 14, pp. 1261–1275, Feb. 2021.
[20] Q. Peng, Y. Jia, L. Liang, and Z. Chen, “A Task Assignment Scheme for Parked-Vehicle Assisted Edge Computing in IoV,” in: *93rd Vehicular Technology Conference (VTC2021-Spring)*, IEEE, Finland, pp. 113–119, Jun. 2021.
[21] X. Hu, X. Tang, Y. Yu, S. Qiu, and S. Chen, “Joint Load Balancing and Offloading Optimization in Multiple Parked Vehicle Assisted Edge Computing,” Wireless Communications and Mobile Computing, vol. 1, pp. 1–13, Nov. 2021.
[22] Y. Cheng, A. Vijayaraj, K.S. Pokkuluri, T. Salehnia, A. Montazerolghaem, and R. Rateb, “Vehicular Fog Resource Allocation Approach for VANETs Based on Deep Adaptive Reinforcement,” Learning Combined With Heuristic Information, IEEE Access, vol. 12, pp. 139056–139075, Sep. 2024.
[23] N.K. Saeed, K.A. Asaad, and A.A. Saffer, “Optimized Resource Allocation in Vehicular Fog Computing Environments Using Hybrid MOSP Algorithm,” ZANCO Journal of Pure and Applied Sciences, vol. 36, no. 6, pp. 118–131, Dec. 2024
[24] X. He, Z. Ren, C. Shi, J. Fang, “A Novel Load Balancing Strategy of Software-Defined Cloud/Fog in the Internet of Vehicles,” China Communications, vol. 13, no. 2, pp. 140–149, Jan. 2016.
[25] A.J. Kadhim and S.A.H. Seno, “Maximizing the Utilization of Fog Computing in Internet of Vehicle Using SDN,” IEEE Communications Letters, vol. 23, no. 1, pp. 140–143, Jan. 2019.
[26] M. Ibrar, A. Akbar, S.R. Jan, M.A. Jan, L. Wang, and H. Song, ”ARTNet: AI-based Resource Allocation and Task Offloading in a Reconfigurable Internet of Vehicular Networks,” IEEE Transactions on Network Science and Engineering, vol. 9, no. 1, pp. 67–77, Feb. 2022.
[27] U. Maan and Y. Chaba, “Deep Q-network based Fog Node Offloading Strategy for 5 G Vehicular Adhoc Network,” Ad Hoc Networks, vol. 120, pp. 178–193, Sep. 2021
[28] E. Qafzezi, K. Bylykbashi, P. Ampririt, M. Ikeda, K. Matsuo, and L. Barolli, “An Intelligent Approach for Cloud-Fog-Edge Computing SDN-VANETs Based on Fuzzy Logic: Effect of Different Parameters on Coordination and Management of Resources,” Sensors, vol. 22, no. 3, pp. 1-20, Jan. 2022.
[29] S.A. Syed, M. Rashid, S. Hussain, F. Azim, H. Zahid, A. Umer, A. Waheed, M. Zareei, and C. Vargas-Rosales, “QoS Aware and Fault Tolerance Based Software-Defined Vehicular Networks Using Cloud-Fog Computing,” Sensors, vol. 22, pp. 1–17, Jan. 2022, https://doi.org/10.3390/s22010401
[30] P. Li, W. Xie, Y. Yuan, C. Chen, and S. Wan, “Deep Reinforcement Learning for Load Balancing of Edge Servers in IoV,” Mobile Networks and Applications, vol. 27, pp. 1461–1474, May 2022.
[31] S.A. Darade, M. Akkalakshmi, and N. Pagar, “SDN based Load Balancing Technique in Internet of Vehicle using Integrated Whale Optimization Method,” AIP Conf. Proc, vol. 2469, no. 1, pp. 31–43, Mar. 2022.