Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/200214

Effective MEC assisted Internet of Vehicles Task Offloading Framework with DRL

Sulieman Shelash Mohammad[1,2,*], *Hamza Abu Owida*[3], *Hanan Jadallah*[1], *Asokan Vasudevan*[2,4], *and Mohammad Faleh Ahmmad Hunitie*[5]

[1]Electronic Marketing and Social Media, Economic and Administrative Sciences, Zarqa University, Zarqa 13110, Jordan
[2]Faculty of Business and Communications, INTI International University, Negeri Sembilan 71800, Malaysia
[3]Department of Medical Engineering, Faculty of Engineering, Al-Ahliyya Amman University, Amman, Jordan
[4]School of Management, Shinawatra University, 99 Moo 10, Bangtoey, Samkhok, Pathum Thani 12160 Thailand
[5]Department of Public Administration, School of Business, University of Jordan, Amman, Jordan

Received: 8 Jul. 2025, Revised: 16 Dec. 2025, Accepted: 5 Jan. 2026
Published online: 1 Mar. 2026

Abstract: The Internet of Vehicles (IoV) is a new paradigm that is being driven by advancements in automotive networking and communications. One well-liked method for bolstering delayed applications for entertainment is cloud computing. Nevertheless, cloud computing may cause significant delays for complex applications sensitive to latency, such as autonomous and assisted driving systems and emergency failure management. Mobile edge computing's efficient processing capabilities near devices have led to extensive integration with the IoV.Therefore, this paper proposes an Effective Mobile Edge Computing-based Internet of Vehicles Task Offloading Framework (EMEC-IoVTOF) with deep reinforcement learning to reduce the latency in vehicular communication networks. The first step is to examine the limitations imposed by the car terminals' usage of energy and communication bandwidth. Secondly, this paper uses the mathematical approach for computing the offloading cost of vehicle communication network processing is needed. The next step is to find the best offloading technique by applying particle swarm optimization to job offloading. Additionally, the inertia weight factor is engineered to adaptively vary in response to the objective function value to evade local optimization. In conclusion, the results of the simulation tests demonstrate that the suggested algorithm can efficiently distribute computing workloads in the IoV.

Keywords: IoV, Mobile Edge Computing, Particle Swarm optimization

1 Introduction

The number of automobiles on the road is on the rise because to the expansion of the car industry and rising incomes. The result has been severe traffic congestion and an increase in the frequency of accidents [1,2]. To overcome this scenario IoV have been used. The ability for automobiles to communicate with one another and enhance road safety is a key aspect of the IoV [3,4]. When done right, MEC can boost the efficiency and speed of mobile apps and services by making the most of the resources available at the edge. To lessen the load on the cloud's main servers, preprocess and filter data using edge servers [5,6]. Verify that the latency, throughput, and stability of edge services are up to par with what is needed for mobile apps [7,8]. One area of machine learning that merges deep learning with reinforcement learning is called DRL. An agent gains decision-making skills through reinforcement learning, a subfield of machine learning, when it encounters incentives and penalties in its environment [9,10].

Modern technology like connectivity, big data, and AI are all a part of EMEC-IoVTOF [11,12]. It will play a crucial role in the smart transportation system of the future. Traditionally, computation offloading involves uploading jobs to a cloud centre to meet the computational resource requirements of the workloads [13,14]. A major network transmission delay could occur, though, because the cloud centre is usually quite a distance from cars [15,16]. To offer customers with computer services that are close to their location, edge computing places servers at the network's periphery [17,18]. By skipping the cloud and going straight to the edge servers, workloads can be transferred more quickly with

* Corresponding author e-mail: dr_sliman@yahoo.com

edge computing [19,20]. To meet the time delay limitations of jobs, the IoV can include edge computing [21]. The foundational technology of computing at the edge is computation offloading. To decrease task execution delays and energy consumption in tasks offloading systems, current research on computation offloading is centred on offloading decisions and tasks scheduling [22].

Developing a reliable method for offloading computations and scheduling jobs efficiently is crucial for meeting the demands of users for high-quality computing services [23]. The EMECallows for easier and more effective management of network resources, it is worth considering. The problem of dealing with high-dimensional continuous action space remains a problem for generic reinforcement learning systems, even after taking both energy consumption and time delay into account together [19]. This research follows up by investigating the scheduling and task offloading strategies in (IoVTOF) environments via the lens of DRL [24]. Vehicle edge computing systems can be made more efficient and effective by making better use of computing resources when tasks are offloaded. Adaptively learning the vehicle's operating status and task demands, and then making adjustments to task offloading choices, might boost the scalability and flexibility of automotive edge computing systems [25]. This study primarily contributes to the following areas:

– The purpose of EMEC-IoVTOF including deep reinforcement learning (DRL) is to provide a model for in-vehicle terminal task offloading and scheduling.
– In DRL, the value function or policy function is used in reinforcement learning techniques. Utilizing edge computing, users are able to access computer services that are near to their location by deploying servers at the network's periphery. To move decision-making and task scheduling away from the edge server, a deep reinforcement learning approach based on the pointer network was developed.
– In this paper EMEC-IoVTOF is executed in the edge server and reduce the energy consumption of vehicles and to provide a efficient IoV to the environment.

Here is the outline for the rest of the paper: The relevant research is presented in Section 2. The EMEC-IoVTOF with DRL approach is covered in Section 3, and the IoV's efficiency is described in Section 4.

2 Related study

Several research have investigated on effective MEC assisted Internet of Vehicles Task Offloading Framework. Here are some relevant research works.

A Directed Acyclic Graph (DAG) representing the dependent tasks, and an intelligent task offloading system that uses off-policy reinforcement learning enabled by a Sequence-to-Sequence (S2S) neural network. This paper merge a particular off-policy policy gradient method with a trimmed surrogate objective to enhance training efficiency. After that, use synthetic DAGs to describe heterogeneous applications and run comprehensive simulation tests. During training, it converges quickly and consistently. Under different conditions, it achieves better performance than the current methods while approximating the ideal solution in terms of latency and energy usage by Wang et al. [26].

By combining the Deep Deterministic policy gradient (DDPG) algorithm with mobile network operators' (MNOs') central control system, Kong et al. [27] create a joint computing and caching framework that mobile customers may access.It is centered on the scenario of the Internet of Vehicles, which relies on the mobile network supplied by MNO. In this research, introduce an optimization problem that takes into account the computation and caching energy costs to decrease MNO's energy cost. The inadequate processing resources in automobiles may be compensated for, according to Xu et al. [28], by upgrading vehicle digital twins and offloading services to Edge computing devices (ECDs). Nevertheless, a solution is suggested for DT-empowered IoV in edge computing—a service offloading (SOL) approach that utilizes deep reinforcement learning—because ECD would overload under heavy service demands, thereby reducing the Quality of service (QoS). Optimal offloading choices are obtained by SOL via the use of Deep Q-network (DQN), a combination of DRL that approximates value functions.

To offer a more comprehensive picture of the environment, Gao et al. [29] utilize an LSTM network as an internal state predictor; a BRNN is then applied to learn and improve the features derived from the agents' conversations. To get the desired outcomes, the policy that has been decided by reinforcement learning is put into action as an offloading technique. To meet the needs of users and tasks in real-time One strategy to improve performance in conjunction with cloud computing is MEC, which involves processing data at the network's periphery.

For large-scale, sparsely dispersed user equipment on the mobile edges, unmanned aerial vehicles (UAVs) have been used as aided edge clouds (ECs). A Markov decision method is developed for the mobile edge computing system that is helped by several UAVs. An investigation into a cooperative multi-agent deep reinforcement learning framework is conducted to ascertain the combined approach to trajectory design, task distribution, and power management. In their work, Zhao et al. [30] take into account the high-dimensional continuous action space and use the twin delayed deep deterministic policy gradient technique listed in Table 1.

Table 1: Comparison of Methods, Advantages, and Limitations in Edge Computing and IoV Scenarios

Method	Advantages	Limitations
S2S-NN	–Enhanced training efficiency –Better performance than current methods –Approximates ideal solution in latency and energy usage	–Dependent on synthetic DAGs for simulation tests –May require substantial computational resources for training
DDPG Algorithm	–Minimization of MNO's energy cost –Optimization of computation and caching energy costs	–Focuses specifically on the IoV scenario, limiting generalizability –Dependency on centralized control system for computing and caching resource allocation
SOL Method using DQN	–Complements vehicles' computational resources –Optimizes offloading decisions for DT-empowered IoV in edge computing	–Risk of edge computing overload under excessive service requests –Dependency on real-time updates of digital twins and offloading decisions
LSTM	–Provides more complete environmental state prediction –Enhances features obtained from agent communication –Real-time performance improvement through MEC	–Complexity in managing LSTM and BRNN architectures –Potential challenges in coordinating multiple agents in a dynamic MEC environment
TD3 Algorithm	–Addresses high-dimensional continuous action space –Optimizes joint trajectory design, task allocation, and power management for multi-UAV assisted MEC	–Requires significant computational resources for training –Complexity in coordinating multiple UAVs and managing continuous action spaces

3 Proposed method

Due to the Internet of Vehicles, vehicle data processing and computing demands have expanded tremendously. Effective task offloading is needed to achieve these objectives and optimize system performance. MEC reduces latency and bandwidth by using proximity-based computing with edge servers. By considering communication bandwidth, energy utilization, and system efficiency, DRL makes effective task offloading choices.

The EMEC–IoVTOF has been developed to address the challenge of reducing latency in vehicular communication networks. This is crucial for applications like autonomous driving and emergency management. The framework incorporates crucial elements to enhance the effectiveness of job offloading decisions. To guarantee smart offloading decisions, the system is built on top of deep reinforcement learning. Of the many things being considered, energy usage and automobile terminals' imposed transmission bandwidth limits set the scope for this issue. Figure 1 illustrates a system that backs-up this decision-making process with an all-encompassing offloading cost calculation.

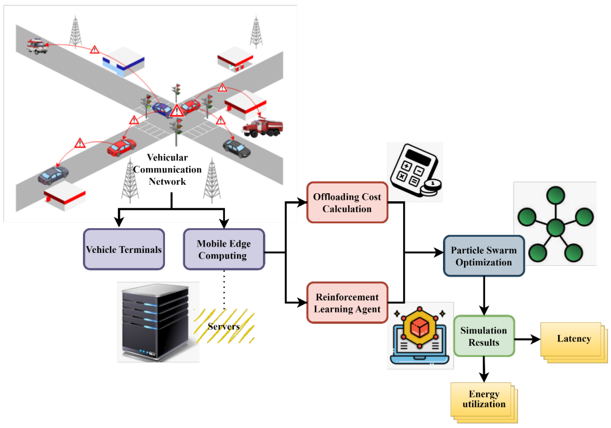

Fig. 1: MEC-IoV

Through this approach, tasks can be divided between mobile edge computing servers and vehicle terminals in an effective manner. Moreover, Particle Swarm

Optimization is employed by the framework to optimize offloading mechanisms thus improving overall efficiency of function distribution. To avoid getting stuck at local optimizations, values of the inertia weight factor should be adjusted depending on objective functions' values. Simulation experiments have shown that suggested method appropriately distributes computing workloads inside IoV. EMEC-IoVTOF reduces latency while optimizing resource utilization which helps seamless integration of complex applications into vehicle networks.

$$JBB = \frac{\sigma \times \sqrt{\forall} - \exists \times log(\rho \times \sigma)}{\int_{\pi}^{\in} \frac{\rho \times \nabla}{\mu + \tau}} + \frac{\delta \times sin(\rho)}{R} \tag{1}$$

Equation 1 shows the suggested framework's cost function for job offloading JBB. For the purpose of trying to find the best offloading technique $\frac{\rho \times \nabla}{\mu+\tau}$, it takes into account variables like data transfer rate δ, computing capabilities $\forall$, delay $\exists$, energy usage ρ, connection bandwidth σ, and system features R.

$$r = -\frac{r_c \cdot k}{S_q} \cdot C \cdot pqf\left(\frac{-S}{PR}\right) \cdot \left(\frac{\beta^{m}_{B,t}}{\rho}\right) \cdot (\varphi) \tag{2}$$

The suggested framework's successful information rates for offloading tasks is represented by Equation 2. It incorporates several elements, including channel reliability r_c, transmit bandwidth r_c, quality of signal s_q, computational capacity C, processing demands pqf, distance S, power for transmission P, transmit path loss PR, fading value $\beta^{m}_{B,t}$, mobility aspect m, packets error rate m, and job urgency φ.

$$BEQ = \frac{\sqrt[3]{\gamma} \times \infty}{\frac{log(\delta \times \tau)}{\pi + \alpha} + \frac{cot(1/2)}{\pi}} + \frac{\sqrt[3]{\gamma} \times \infty}{\frac{log(\nabla \times \exists)}{\gamma + \beta} + \frac{cos(r)}{\tau}} \tag{3}$$

The suggested offloading of tasks paradigm is defined by Equation 3 as the BEQ. When trying to determine the best approach for allocating tasks, it takes into consideration factors like processing capability ∞, job urgency γ, and function trigonometry. Additional variables include system settings $(\delta, \tau, \pi, \alpha, \nabla, \exists, \beta)$.

The model of the vehicular edge computing system's RLF is shown in Figure 2. Intelligent agent cars' interactions with their surroundings are depicted in the model. The agent watches the state and decides what to do based on the trained policy. Following the selection of the action at the state of the environment changes. Afterwards, the intelligent agent cars are rewarded immediately for the change. The action space depicts a collection of possible activities that agents under centralised direction can take within the specified time frame. The agent's responsibilities may include choosing which cars to service, deciding how to offload, and calculating the offload fraction for each vehicle's tasks. The aim of optimizing the reward function of the

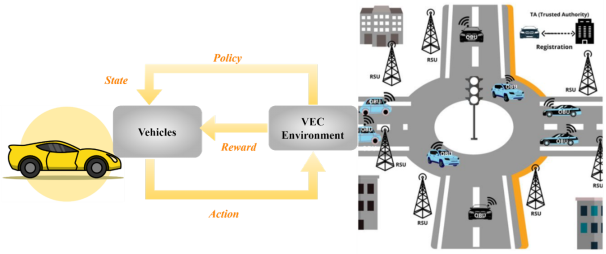

Fig. 2: RLF model based on vehicle edge computing system

vehicular edge computing system is to minimize system cost while training the reinforcement learning agent to maximize long-term benefits. The decision-making technique for job offloading in the VEC system is DDPG, a DRL algorithm based on actor-critic architecture. The policy network and the network (the critic) are trained iteratively through interaction with the environment using the algorithm. Agents can learn optimal loading decisions using this technique, which considers the system's state and action space.

$$\nabla Q_{rst} = \frac{250 \cdot (1-P)}{P^3} + \frac{350 \cdot (1-Q)}{Q^3} - \frac{450 \cdot (1-R)}{R^3} \tag{4}$$

As an expression of the dependability variables (P, Q, R) in the suggested framework, Equation 4 shows the partial inverse of the QoS (Quality of Service) function Q_{rst}. The evaluation of QoS responsiveness to shifts in reliability is accomplished by combining components such as dependability requirements system characteristics (250, 350, 450), and quadratic terms.

$$tan\,\alpha + sin\,\beta = 4\,cos\frac{1}{3}(\tau + \varepsilon)\,sin\frac{1}{4}(\varepsilon - \alpha) \tag{5}$$

A link between trigonometric operations in the suggested structure involving angles $\alpha, \beta, \tau, and \varepsilon$ is shown in Equation 5. It represents the equilibrium between the tangential and sin of the angle α, which is equivalent to the product of the cotangent of half of the total of τ and ε and the sin of a quarter of the gap among ε and α.

$$cos\,\exists \pm tan\,\sigma = 3\,sin\frac{1}{5}(\vartheta \pm \delta)\,tan\frac{1}{4}(\tau \mp \alpha) \tag{6}$$

Within the suggested paradigm, trigonometric equations involving angles are described by Equation 6. This shows how the cosine of angle $\exists$, divided by the direction of angle τ, equals twice the sine value of one-fifth of the sum or variation of ϑ and δ, and the tangential of a quarter of the summation or variation of τ and α equal zero.

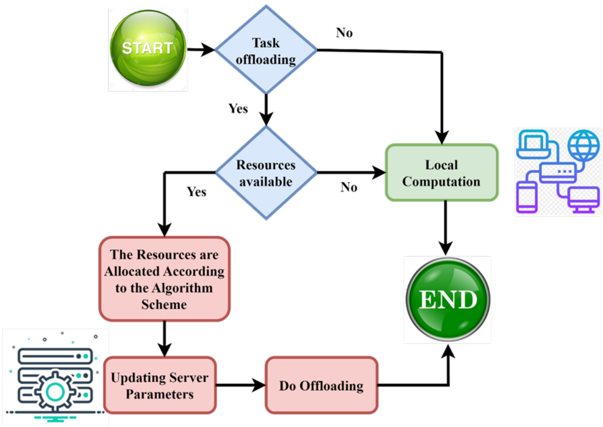

Fig. 3: computing offloading in mobile edge computing environment

To process and aggregate tiny packets created by IoV services prior to their arrival at the core network, the MEC can be utilised which elaborates in figure 3. For IoV devices that run on batteries, this means better scalability and application flexibility. By shortening the time data travels between servers and devices, MEC helps keep devices and services running smoothly for longer, which is good for business in the long run. Assuming all of the data collected by these IoV devices is transferred to the cloud service centre for processing, the remote cloud will be under a great deal of strain. Unfortunately, the majority of IoV devices are either underpowered or unable to analyse data. To process and aggregate tiny packets created by IoV services prior to their arrival at the core network, the MEC can be utilised. For IoV devices that run on batteries, this means better scalability and application flexibility. By shortening the time data travels between servers and devices, MEC helps keep devices and services running smoothly for longer, which is good for business in the long run.

Initializing parameters such as step time, attenuation aspect, and gradient descent variety is the first stage in Algorithm 1 for task offloading utilizing DRL. A random initialization is performed on the neural network's parameters θ, and an experience replay buffer E is established. Initializing the environment state and obtaining its characteristic vector are done for each episode. This softmax output is generated by the neural network processing the characteristic vector throughout each iteration. Based on this output, actions are carried out and rewards are earned accordingly. After that, the replay buffer is where the experience tuples are kept. Employing batch gradient descent, the parameters of the neural network are modified after each episode.

$$g^z = 2 + \frac{t}{2!} + \frac{t^2}{3!} + \frac{t^3}{4!} + \cdots, -a < b < a \tag{7}$$

Algorithm 1 Task Offloading using (DRL)

Initialize Step duration, attenuation aspect, pattern variety of gradient descent
Initialize the parameters θ of the neural network randomly and initialize the revel in replay buffer E.
for every episode **do**
  Initialize the surroundings state, get its characteristic vector (CV).
  for every iteration **do**
    Use $V(x_n)$ as the enter, acquire the softmax output of the neural community.
    Execute the motion xn,take a look at the new surroundings nation i++, and receives the corresponding instant reward n_i.
    Put the quadruple $(a+b)$ into the experience replay buffer
    if i++ is the terminated state **then**
      break
    end if
  end for
  Sample mini-batches from the experience replay buffer $\mathscr{E}$.
  Obtain samples from the experience replay buffer E,and replace the parameters θ of neural community through minimizing the objective feature in equation the usage of batch gradient descent set of rules.
end for

A function g^z, with g being a constant and z being a variable, usually denoting time t, may be expressed as the power chain expansion in Equation 7. To keep the function confined inside the interval $[-a,a]$, the range of converging for the series of values is indicated by the inequality $a < b < a$.

$$z = \frac{-t \pm \sqrt{f^2 - 7rs}}{3p} + \frac{-b \pm \sqrt{b^2 - 7fr}}{4d} \tag{8}$$

In Equation 8 link among z and t, outlined by the set of formulas, is established by applying the quadratic formula to two independent expressions. Variables f, r, and s are used in the initial expression, whereas variables b, f, r, and d are utilised in the second expression. There are two possible answers to each equation, as shown by the $\sqrt{f^2 - 7rs}$ and $\sqrt{b^2 - 7fr}$ terms.

$$r(m) = r_0 + \sum_{l=1}^{\alpha}(b_m \, cot \frac{\delta\gamma\beta}{z} + \tau_w \, sin \frac{\beta\theta\mu}{y}) \tag{9}$$

The function $r(m)$ is defined by Equation 9 and is dependent on the parameter m. The constant starting point r_0 is defined. Each term in the function is calculated as the sum of the two gestures, and the function itself is a summation over α terms. The cotangent functional is involved in the first communication, which contains variables b_m, δ, β, and z, and the sine function is involved in the second communication, containing variables τ_w, θ, μ, and y.

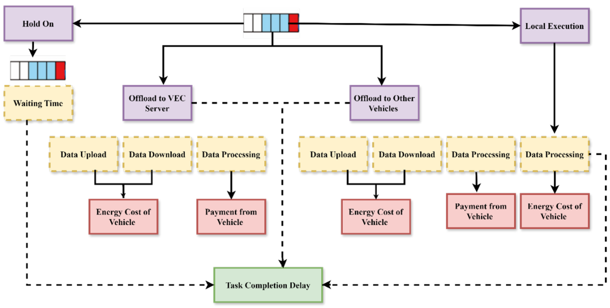

Fig. 4: offloading of vehicle edge computing

The new VEC technology allows data processing and storage to be relocated to the edge of IoT networks. Figure 4 shows how low-power and delay-sensitive mobile apps on the IoV may benefit from offloading computing tasks to the end of the VEC network. Issues with resource management, data security and privacy due to increased mobility, and the unpredictability of the IoV all added to the difficulties of VEC offloading.

This study aims to address this by examining VEC's computing work offloading. To categorise the present offloading of computing workloads and survey the key offloading systems and methodologies in the VEC sector. This paper talk about the possibility of VEC.

To determine the best methods for offloading computing duties in VEC, researchers could use this survey as a reference to identify and comprehend the key features of VEC. Transferring information from a vehicle's internal systems or sensors to outside computing resources, like edge servers or the cloud, is commonly referred to as data upload in the context of vehicle edge computing. For many uses, such as real-time tracking, data analysis, and decision-making, this data transfer is crucial. As shown in Figure 4 edge computing in vehicles entails processing data by analysing, transforming, and manipulating information retrieved from in-vehicle sensors and systems.

$$(2+w)^m = 5 + \frac{qw}{6!} + \frac{q(w-1)r^2}{7!} + \cdots + \frac{q(w-1)r^2}{8!} \quad (10)$$

In equation 10 the statement $(2+w)^m$ is expanded into a power series, with m standing for the exponents and w being a parameter. Each term in the series increases in control, beginning with the constant term 5 and progressing to terms involving q and r.

$$(w+z)^b = \sum_{p=0}^{r} \binom{w}{s} y^e d^{l-y} + \binom{z}{l} d^w z^{r-n} \quad (11)$$

The analysis of communication bandwidth determined with $(w+z)^b$, with b standing for the exponents in Equation 11. By adding up the terms from $p = 0$ to r, the equation captures the combinations of variables w, s, y, e, d, and l.

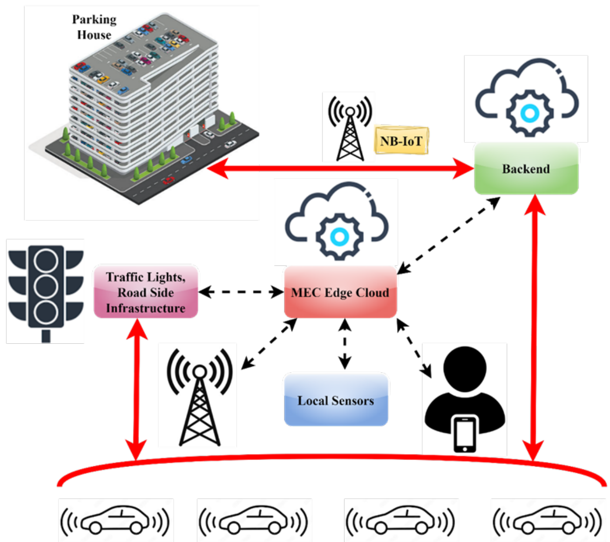

Fig. 5: autonomous vehicle with edge computing

$$M = -\frac{A_d \cdot c}{\alpha_f} \cdot B \cdot exp(\frac{-Q}{JK}) \cdot (\frac{A^e_{w \cdot p}}{T}) \cdot (Z) \quad (12)$$

The equation 12 denotes the analysis of user energy M, which takes into account an array of factors that influence energy use. This expression contains variables that stand for variables associated with device properties, environmental factors, and models of energy consumption, such as A_d, c, α_f, B, Q, JK, and the exponential components for $(\frac{A^e_{w \cdot p}}{T}) \cdot (Z)$.

Sensors are now embedded in all smart devices. The data is handled entirely in the vehicle, but many in-car applications still require data to be sent to the cloud. Users can enjoy lower data transmission costs because to edge computing, which enables processing and control of data at the edge rather than transmitting it to the cloud. An idea in network design known as Multi-access Edge Computing (MEC) allows for an IT service environment and cloud computing to exist at the periphery of a cellular network. Applications may now make choices in real-time using data acquired from mobile devices and IoT sensors due to MEC, which provides high-bandwidth, low-latency access to radio network information.

Deploying cloud infrastructure near end-users and devices, often at the network's periphery, is what's known as MEC edge cloud. Services and applications that rely on fast, reliable connections are housed in this infrastructure. Several potential sites exist for the deployment of MEC edge clouds, including base stations, cell towers, and network aggregation hubs. A low-power wide-area (LPWA) technology standard, Narrowband Internet of

Things (NB-IoT) allows devices in the IoT ecosystem to communicate and interact efficiently. For many IoTuses, NB-IoT's licenced spectrum operation provides a safe and dependable communication channel as shown in Figure 5.

Algorithm 2 Task Offloading to Parked Vehicles

Input: the set of obligations H, the most resources can be assigned to every task $f_{\max}$, and the overall available resources of the parking cluster P_c.
Use P as the current available computing assets of the parking cluster, initialize $P = Pn$.
while $H = 0$ **do**
  if $P > 0$ **then**
    for every mission $n \in H$ **do**
      for every i such that $i \leq f_{\max}$ and $i < H$ **do**
        Select the n and i with minimized, denoted as n_0 and i_0.
      end for
    end for
  else
    break
  end if
  Allocate i_0 resources to task n_0.
  $H := H - n_0$.
  $P := P - i_0$.
end while

When starting Algorithm 2 to offload work to parked vehicles, the following information is input: the set of tasks H, the maximum resources that may be provided to each task f_{max}, and the total resources available in the parking cluster P_c. After that, the procedure sets the starting state of the computing resources P to equal the total resources P_n. The method will iteratively pick the task and resource allocation that minimizes a given measure while there are tasks left in set H. After a task is chosen, resources are assigned to it, and then the task and its resources are no longer considered.

$$LPO = \frac{\sqrt[3]{\nabla} \times \varepsilon}{\frac{log(\tau \times \delta)}{\beta + \tau} + \frac{tan(\mu + \vartheta)}{Q}} + \frac{\sqrt[3]{\gamma} \times \infty}{\frac{log(\varepsilon + \rho)}{\tau + \alpha} + \frac{cos(\exists + 1)}{{}^{\circ}F}} \quad (13)$$

The latency analysis, including all the parameters that affect latency in a system, is given by Equation 13 as LPO. The following words are used to describe parameters relating to network features, computing abilities, and environmental circumstances: ∇, ε, τ, δ, μ, ϑ, Q, γ, ∞, ρ, α, $\exists$, and ${}^{\circ}F$.

$$(X + Y)^Z = \sum_{l=0}^{p} \binom{s}{w} x^k \mu^{z-y} + \binom{a}{q} w^n a^{r-p} \quad (14)$$

The analysis of efficiency Z as the exponents and X and Y as variables reflects the effectiveness in a system, as seen in Equation 14 which describes the system. From $l = 0$ to p, the formula incorporates terms that consist of combinations of the variables p, w, x^k.

$$r^s = w + \frac{v}{7!} + \frac{v^2}{8!} + \frac{v^3}{9!} + \cdots, -\sigma < f < \sigma \quad (15)$$

The analysis of task offloading for the $functionr^s$, with r^s as the base and s as the exponent, via the variables w and v, is shown in Equation 15. Regarding task offloading, the inequality $-\sigma < f < \sigma$ may serve as a limit or cutoff that limits the variable f to a particular range.

DRL and MEC are used to create a structure for offloading tasks in IoV systems. According to the proposed design, this should be done in such a way that it maximizes the use of energy, communication bandwidth, latency as well as system efficiency which will lead into better performance and reliability of IoV systems. Future works need to focus on scalability improvement where larger networks should be dealt with by the system while at the same time utilizing more advanced methods in DRL.

4 Result and Discussion

IoV has opened up many possibilities for improving transportation through the adoption of advanced algorithms and communication technologies. This research looks at various aspects of an IoV system such as energy utilization, efficiency, communication bandwidths among others. Knowing these features will enable researchers come up with smarter collision detection systems, reduce power consumption, lower latencies enhance overall performance through intelligent task offloading. Each factor is analysed in relation to its effect on IoV ecosystems before suggesting areas for further investigation.

Dataset description: This dataset contained attributes that affect how cars crash in Internet of vehicles [31]. It can be employed by researchers to build collision detection alarm systems using smart algorithms or any other AI related study within this domain.

4.1 Communication Bandwidth

To analyze communication bandwidth within mobile edge computing (MEC), one needs to know data transmission requirements between mobile devices and edge servers. Mobile devices have apps that generate different amounts of data traffic which directly affects processing needs as well as required bandwidths. For instance streaming high definition videos consumes more resources than sending raw sensor readings over network links. In addition there is direct proportionality between bandwidth requirements with number of devices connected on edges server sides.

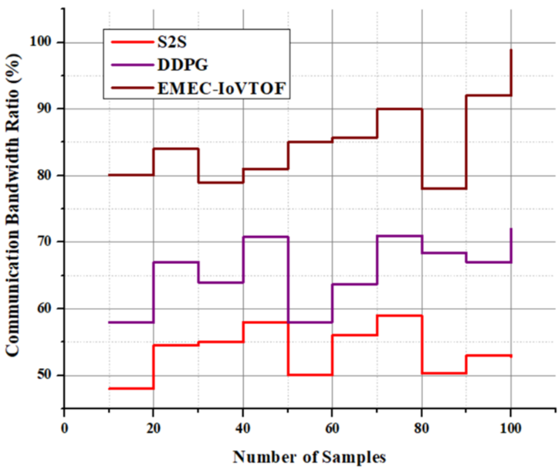

Fig. 6: Analysis of Communication Bandwidth

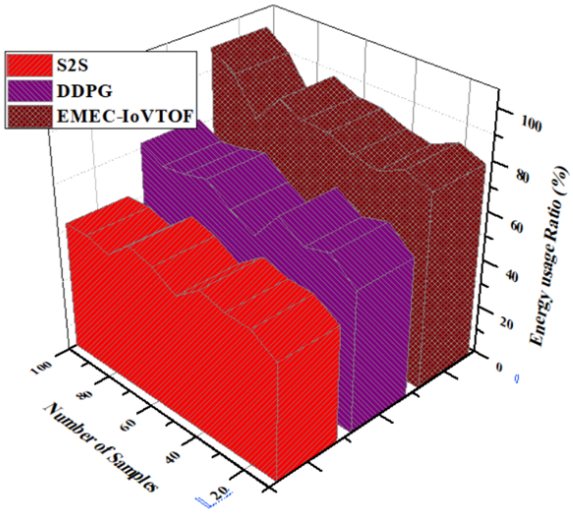

Fig. 7: Analysis of Energy Usage

Broadband capacity increases linearly depending on the number of connected things as shown in figure 6 equation 11. However if location where edge servers are deployed is close to mobile devices they serve then latency will be reduced together with bandwidth needed sometimes. The amount of information that can travel from one device to another through an edge server is influenced by factors like network reliability, signal strength and congestion levels within it. So data compression or optimization techniques may be used for reducing data transmission load hence lowering bandwidth requirement. These considerations are done during MEC communication bandwidth analysis to ensure network architecture is capable of supporting desired data transfer speeds coupled with latency thresholds for planned applications.

4.2 Energy Usage

When studying energy consumption within IoV, power requirements for vehicles, communication networks and other infrastructure in the ecosystem should come into play. Additionally energy required to charge EVs has become a critical concern recently.

The protocols and technologies chosen for communication affect the energy consumed by vehicle and infrastructure communication devices as shown in figure 7 and equation 12. Computers on board use power to process information collected by sensors, cameras, and other communication gadgets while servers at the edge or in the cloud do so too. Energy is needed by the infrastructure that supports IoV such as base stations, routers, servers among others for its operation and maintenance. One needs to collect and analyze data about energy consumption from different components, predict energy use under various scenarios and find ways of optimizing as well as increasing efficiency so as to carry out an energy analysis in IoV systems. This method can be used to build IoV systems which are energy efficient, reliable and environmentally friendly.

4.3 Latency

To perform latency analysis within IoV, it is important that to create DRL algorithms which will optimize latency-sensitive task and reduce communication delays. Assigning RL problem description towards optimization of latency in IoV. Tasks that need optimization due to latency should be identified e.g., collision avoidance systems or traffic light control mechanisms.

Think of the IoV environment as a RL ecosystem where there are buildings, cars, roads among others connected through networks like any other ecosystem would be described. Use measures for latency to define state space, action space and rewards is shown in figure 8 and equation 13. DRL agent can learn optimal policies for minimizing latencies within IoVs through simulation or historical data Train stability can be achieved through experience replay techniques coupled with target networks Determine how much lower does trained DRL agent brings down latencies compared old fashioned methods when evaluated under realistic scenarios of IoVs Evaluating how well does trained DRL agent perform in terms lowering down latencies against earlier approaches during realistic Iovs Take advantage of this trained DRL agent by making it capable of adapting to changing network conditions and traffic patterns in real-time so as to optimize latency in IoV.

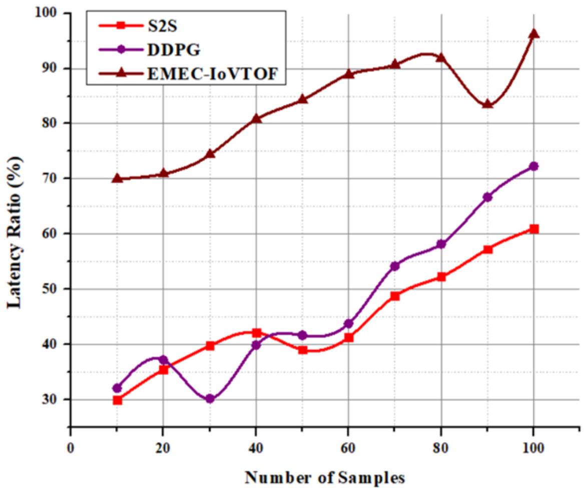

Fig. 8: Latency Analysis

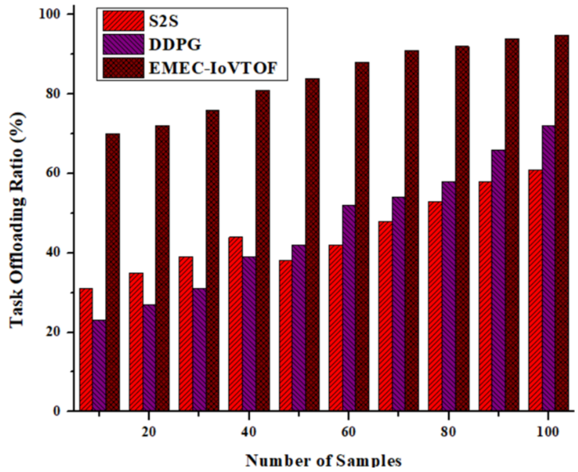

Fig. 10: Analysis of Task Offloading

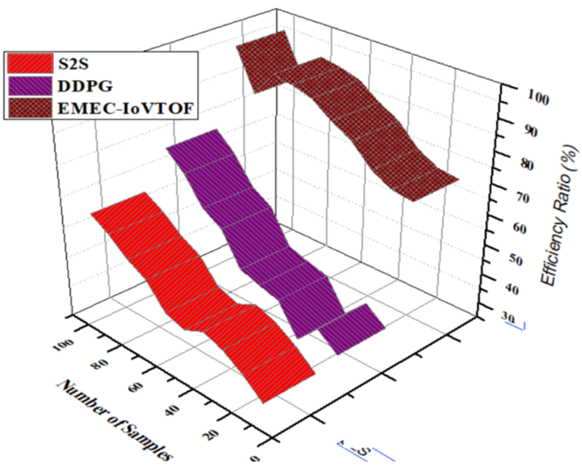

Fig. 9: Efficiency Analysis

4.4 Efficiency

It is possible to analyze different parts of the IoV systems for efficiency using deep reinforcement learning (DRL) algorithms. Energy consumption, traffic flow, resource utilization and overall system performance are some key measures that need attention when looking at efficiencies within an IoV.

Create an RL problem formulation for optimization of efficiency is shown in figure 9 and equation 14. The state space, action space and reward function could be defined using established efficiency metrics which this paper can adopt. A RL model should be created representing all vehicles, buildings and networks connecting them within Internet of Vehicles (IoV) setting Train DRL agent with historical data or simulation to learn good strategies for traffic management, route optimization, reduction of energy use among others Resources should be utilized better while still being able to cope up with changes by letting trained DRL agent make decisions in real time injected into IoVs systems. There is potentiality that exists which could enable one improve on system performances, lower down energy consumptions as well as increase overall efficiencies within IoVs through DRL based analysis of efficiency.

4.5 Task Offloading

Task offloading refers to studying how computing tasks can be moved from vehicles onto edge or cloud servers in the context of Internet-of-Vehicles (IoVs). Find out use cases that can benefit from task offloading such as applications requiring low-latency data processing; computations demanding many resources etc. Determine how task offloading affects the latency, dependability, and bandwidth of communications.

Think about various offloading mechanisms and communication systems is expressed in figure 10 and equation 15. Research how much power is required to process task locally vs when they are offloaded. the amount of energy used for data transmission, calculation, and idle time. Test various task offloading mechanisms and rank them according on how well they reduce latency, save energy, and improve system efficiency. Create optimisation methods to enhance the efficacy of offloading tasks, including dividing tasks, balancing workloads, and allocating resources. Improving the performance, efficiency, and reliability of IoV systems is attainable by studying task offloading in IoV, which considers the specific problems and requirements of vehicular contexts.

The complex dynamics of IoV systems are better understood due to this paper, which opens up the possibilities to smart algorithms and technologies that may improve vehicle dependability, efficiency, and safety. To fully realize the Internet of Vehicles' (IoV) revolutionary potential in transportation system transformation and urban mobility quality improvement, more research in these areas is highly encouraged.

5 Conclusion

Due to the ever-increasing capabilities of IoV technology, new service applications are continually popping up. Smart cities and smart transportation, two areas of intense study, are as reliant on driving networking technologies as the explosion of IoV data. The existing IoV and core networks are therefore under heavy load. To reduce the burden on the central network while still meeting the demanding latency requirements of IoV applications, a solution to task offloading in IoV edge computing architecture that uses reinforcement learning is proposed. In this study, EMEC-IoVTOF builds the computation, communication, interference, and privacy protection models of the task offloading technique according to the anticipated IoV with DRL system architecture. The goal function works to reduce the user's cost as much as possible. Cost stays cheap in relation to other offloading systems as user numbers, vehicle speeds, and MEC processing capacity all increase.

The offloading rate is higher for the suggested technique like a deep reinforcement learning model for optimising trajectory control and task offloading from start to finish. To optimise several criteria, such as compute latency and energy consumption of the MEC network, the suggested model regulates the fraction of offloading job. The future work of this paper involve researching the feasibility of multi-UAV collaboration with task segmentation. The real road scene is extremely complex, the situation under consideration is a straight road with one direction of travel and no intersections. More study is required to fully understand the complexities of real-life traffic scenarios. On top of that, IoV users are constantly on the go, and network topology is always evolving. Communication disruptions might occur when user links are changed often. Choosing the right relay can increase communication dependability, which is something to think about for future work that involves integrating relay communication.

Acknowledgement

This research was partially funded by Zarqa University.

References

[1] Z. Ning, K. Zhang, X. Wang, L. Guo, X. Hu, J. Huang and R.Y. Kwok. Intelligent edge computing in internet of vehicles: A joint computation offloading and caching solution. *IEEE Transactions on Intelligent Transportation Systems*,**22**, 2212-2225 (2020).

[2] A.A. Mohammad, S.I.S. Al-Hawary, A. Hindieh, A. Vasudevan, H.M. Al-Shorman, A.S. Al-Adwan and I. Ali. Intelligent data-driven task offloading framework for Internet of vehicles using edge computing and reinforcement learning. *Data and Metadata*,**4**, 521 (2025).

[3] J. Hu, Y. Li, G. Zhao, B. Xu, Y. Ni and H. Zhao. Deep reinforcement learning for task offloading in edge computing assisted power IoT. *IEEE Access*,**9**, 93892-93901 (2021).

[4] A.A.S. Mohammad, K.I. Al-Daoud, M.A. Rusho, A. Alkhayyat, H. Doshi, P. Dey and M. Kiani. Modeling polyethylene glycol density using robust soft computing methods. *Microchemical Journal*,**210**, 112815 (2025).

[5] H. Zhao, J. Hua, Z. Zhang and J. Zhu. Deep Reinforcement Learning-Based Task Offloading for Parked Vehicle Cooperation in Vehicular Edge Computing. *Mobile Information Systems*,**2022**, 9218266 (2022).

[6] A.A.S. Mohammad. The impact of COVID-19 on digital marketing and marketing philosophy: evidence from Jordan. *International Journal of Business Information Systems*,**48**(2), 267-281 (2025).

[7] B. Lin, K. Lin, C. Lin, Y. Lu, Z. Huang and X. Chen. Computation offloading strategy based on deep reinforcement learning for connected and autonomous vehicle in vehicular edge computing. *Journal of Cloud Computing*,**10**, 33 (2021).

[8] A.A.S. Mohammad, K.I. Al-Daoud, S.I.S. Mohammad, T. Samarah, A. Vasudevan and M. Li. Content marketing optimization: A/B testing and conjoint analysis for engagement strategies in Jordan. *Journal of Ecohumanism*,**3**(7), 3086-3099 (2024).

[9] J. Zhang, H. Guo and J. Liu. Adaptive task offloading in vehicular edge computing networks: a reinforcement learning based scheme. *Mobile Networks and Applications*,**25**, 1736-1745 (2020).

[10] A.A.S. Mohammad, K.I. Al-Daoud, B. Al Oraini, S.I.s. Mohammad, A. Vasudevan, J. Zhang and M.F.A. Hunitie. Using Digital Twin Technology to Conduct Dynamic Simulation of Industry-Education Integration. *Data and Metadata*,**3**, 422 (2024).

[11] G. Qu, H. Wu, R. Li and P. Jiao. DMRO: A deep meta reinforcement learning-based task offloading framework for edge-cloud computing. *IEEE Transactions on Network and Service Management*,**18**, 3448-3459 (2021).

[12] A.A.S. Mohammad, S.I.S. Mohammad, B. Al Oraini, A. Vasudevan, A. Hindieh, A. Altarawneh and I. Ali. Strategies for applying interpretable and explainable AI in real world IoT applications. *Discover Internet of Things*,**5**(1), 71 (2025).

[13] L. Geng, H. Zhao, J. Wang, A. Kaushik, S. Yuan and W. Feng. Deep Reinforcement Learning Based Distributed Computation Offloading in Vehicular Edge Computing Networks. *IEEE Internet of Things Journal*,**10**, 12416-12433 (2023).

[14] A.A.S. Mohammad, S.I.S. Mohammad, K.I. Al Daoud, B. Al Oraini, A. Vasudevan and Z. Feng. Building Resilience in Jordan's Agriculture: Harnessing Climate Smart Practices and Predictive Models to Combat Climatic Variability. *Research on World Agricultural Economy*,**6**(2), 171-191 (2025).

[15] S. Li, X. Hu and Y. Du. Deep reinforcement learning for computation offloading and resource allocation in unmanned-aerial-vehicle assisted edge computing. *Sensors*,**21**, 6499 (2021).
[16] A.A.S. Mohammad, Y. Nijalingappa, S.I.S. Mohammad, R. Natarajan, L. Lingaraju, A. Vasudevan and M.T. Alshurideh. Fuzzy Linear Programming for Economic Planning and Optimization: A Quantitative Approach. *Cybernetics and Information Technologies*,**25**(2), 51-66 (2025).
[17] W. Zhan, C. Luo, J. Wang, C. Wang, G. Min, H. Duan and Q. Zhu. Deep-reinforcement-learning-based offloading scheduling for vehicular edge computing. *IEEE Internet of Things Journal*,**7**, 5449-5465 (2020).
[18] A.A.M. Shlash, S.M. Shelash, B. Al Oraini, H.A. Ayman, A. Vasudevan and M.A. Turki. Decoding Consumer Behaviour: Leveraging Big Data and Machine Learning for Personalized Digital Marketing. *Data & Metadata*,**4**, 700 (2025).
[19] Y. Wu, J. Xia, C. Gao, J. Ou, C. Fan, J. Ou and D. Fan. Task offloading for vehicular edge computing with imperfect CSI: A deep reinforcement approach. *Physical Communication*,**55**, 101867 (2022).
[20] A.A. Mohammad, A.M. Al-Ramadan, S.I. Mohammad, B. Al Oraini, A. Vasudevan, N. Alshdaifat and M.F.A. Hunitie. Customer Sentiment Analysis for Food and Beverage Development in Restaurants using AI in Jordan. *Data and Metadata*,**4**, 922 (2025).
[21] J. Huang, J. Wan, B. Lv, Q. Ye and Y. Chen. Joint computation offloading and resource allocation for edge-cloud collaboration in internet of vehicles via deep reinforcement learning. *IEEE Systems Journal*,**17**, 2500-2511 (2023).
[22] X. Li. A computing offloading resource allocation scheme using deep reinforcement learning in mobile edge computing systems. *Journal of Grid Computing*,**19**, 35 (2021).
[23] Z. Xue, C. Liu, C. Liao, G. Han and Z. Sheng. Joint service caching and computation offloading scheme based on deep reinforcement learning in vehicular edge computing systems. *IEEE Transactions on Vehicular Technology*,**72**, 6709-6722 (2023).
[24] Z. Lv, D. Chen and Q. Wang. Diversified technologies in internet of vehicles under intelligent edge computing. *IEEE transactions on intelligent transportation systems*,**22**, 2048-2059 (2020).
[25] J. Zhao, H. Quan, M. Xia and D. Wang. Adaptive resource allocation for mobile edge computing in internet of vehicles: A deep reinforcement learning approach. *IEEE Transactions on Vehicular Technology*,**73**, 5834-5848 (2023).
[26] J. Wang, J. Hu, G. Min, W. Zhan, A.Y. Zomaya and N. Georgalas. Dependent task offloading for edge computing based on deep reinforcement learning. *IEEE Transactions on Computers*,**71**, 2449-2461 (2021).
[27] X. Kong, G. Duan, M. Hou, G. Shen, H. Wang, X. Yan and M. Collotta. Deep reinforcement learning-based energy-efficient edge computing for internet of vehicles. *IEEE Transactions on Industrial Informatics*,**18**, 6308-6316 (2022).
[28] X. Xu, B. Shen, S. Ding, G. Srivastava, M. Bilal, M.R. Khosravi and M. Wang. Service offloading with deep Q-network for digital twinning-empowered internet of vehicles in edge computing. *IEEE Transactions on Industrial Informatics*,**18**, 1414-1423 (2020).
[29] H. Gao, X. Wang, W. Wei, A. Al-Dulaimi and Y. Xu. Com-DDPG: Task offloading based on multiagent reinforcement learning for information-communication-enhanced mobile edge computing in the internet of vehicles. *IEEE Transactions on Vehicular Technology*,**73**, 348-361 (2023).
[30] N. Zhao, Z. Ye, Y. Pei, Y.C. Liang and D. Niyato. Multi-agent deep reinforcement learning for task offloading in UAV-assisted mobile edge computing. *IEEE Transactions on Wireless Communications*,**21**, 6949-6960 (2022).
[31] https://www.kaggle.com/datasets/harunachiromagombe/internet-of-vehicles-dataset

Sulieman Shelash Mohammad is a Professor of Business Management at Al al-Bayt University, Jordan (currently at Zarqa University, Jordan), with more than 17 years of teaching experience. He has published over 100 research papers in prestigious journals. He holds a PhD in Financial Management and an MCom from Rajasthan University, India, and a Bachelor's in Commerce from Yarmouk University, Jordan. His research interests focus on supply chain management, Marketing, and total quality (TQ). His ORCID ID is orcid.org/0000-0001-6156-9063.

Hanan Jadallah is affiliated with Electronic Marketing and Social Media, Economic and Administrative Sciences at Zarqa University, Jordan. Her research interests include digital marketing strategies, blockchain applications in business, and consumer behavior in digital environments. She has contributed to multiple research projects on technological integration in marketing. Her ORCID ID is orcid.org/0009-0005-7138-1167.

Asokan Vasudevan is a distinguished academic at INTI International University, Malaysia. He holds multiple degrees, including a PhD in Management from UNITEN, Malaysia, and has held key roles such as Lecturer, Department Chair, and Program Director. His research, published in esteemed journals, focuses on business management, ethics, and leadership. Dr. Vasudevan has received several awards, including the Best Lecturer Award from Infrastructure University Kuala Lumpur and the Teaching Excellence Award from INTI International University. His ORCID ID is orcid.org/0000-0002-9866-4045.