MEKALA, M.S., DHIMAN, G., PARK, J.H., JUNG, H.-Y. and VIRIYASITAVAT, W. 2024. ASXC2 approach: a service-X cost optimization strategy based on edge orchestration for IIoT. *IEEE transactions on industrial informatics* [online], 20(3), pages 4347-4359. Available from: https://doi.org/10.1109/TII.2023.3315744

ASXC2 approach: a service-X cost optimization strategy based on edge orchestration for IIoT.

MEKALA, M.S., DHIMAN, G., PARK, J.H., JUNG, H.-Y. and VIRIYASITAVAT, W.

2024

ASXC^2 Approach: A Service-X Cost Optimization Strategy Based on Edge-Orchestration for IIoT

M S Mekala, *Senior Member, IEEE*, Gaurav Dhiman, *Senior Member, IEEE*, Ju H. Park *Senior Member, IEEE*, Ho-Youl Jung, Wattana Viriyasitavat, *Senior Member, IEEE*

***Abstract*—Most computation-intensive industry applications and servers encounter service-reliability challenges due to the limited resource capability of the edge. Achieving quality data fusion and accurate service reliability in edge computing for IIoT requires continuous attention to design novel methods to optimize the service-x execution cost. While existing systems have taken into account factors such as device service execution, residual resource ratio, and channel or link condition, the service execution time, cost, and utility ratios of requested services from devices and servers also have a significant impact on service-x execution cost. In order to enhance service quality and reliability, we design a 2-step Adaptive Service-X Cost Consolidation (ASXC^2) approach. This approach is based on the node-centric Lyapunov method and distributed Markov mechanism, aiming to optimize the service execution error rate during offloading. The node-centric Lyapunov method incorporates cost and utility functions, along with node-centric features, to estimate the service cost prior to offloading. Additionally, the design of the Markov mechanism-inspired service latency prediction model assists in mitigating the ratio of offload-service execution errors by establishing a mobility-correlation matrix between devices and servers. In addition, the non-linear programming multi-tenancy heuristic method design help to predict the service preferences for improving the resource utilisation ratio. The simulations show the effectiveness of our approach. The model performance enhance with 0.13% service offloading efficiency, 0.82% rate of service completion when transmit data size is 400 kb, and 0.058% average service offloading efficiency with 40 CPU Megacycles when the vehicle moves 60 Km/h speed in around the server communication range. Our model simulations indicate that our approach is highly effective and suitable to light-weight complex environments.**

***Index Terms*—Edge computing, Industry 4.0, non-linear programming mutli-tenancy method, node-centric Lyapunov method.**

M S Mekala is with School of Computing, Robert Gordon University, Garthdee Road, Aberdeen, AB10 7AQ, Scotland, UK. (E-mail: msmekala@yu.ac.kr & ms.mekala@rgu.ac.uk

Gaurav Dhiman is with Department of Electrical and Computer Engineering, Lebanese American University, Byblos, Lebanon; University Centre for Research and Development, Chandigarh University, India; Department of Computer Science and Engineering, Graphic Era Deemed to be University, India; Division of Research and Development, Lovely Professional University, Phagwara, India; School of Sciences and Emerging Technologies, Jagat Guru Nanak Dev Punjab State Open University, Patiala, Punjab, India (E-mail: gdhiman0001@gmail.com), https://orcid.org/. 0000-0002-6343-5197.

Ju H. Park, Department of Electrical Engineering, Yeungnam University, Gyeongsan 38544, Korea. (E-mail: jessie@ynu.ac.kr).

Ho-Youl Jung is with department of Information and Communication Engineering,Yeungnam University, Gyeongsan 38544, Korea as well as RLRC for Autonomous Vehicle Parts and Materials Innovation, Yeungnam University, Gyeongsan 38544, Korea (E-mail: hoyoul@yu.ac.kr).

Wattana Viriyasitavat is with Business Information Technology Division, Department of Statistics, Faculty of Commerce and Accountancy, Chulalongkorn University, Bangkok 10330, Thailand (E-mail: hardgolf@gmail.com).

I. Introduction

THE AUTOMOBILE industry is undergoing a significant transformation through the use of Digital-Twin Technology in computation-communication to meet the expectations of Industry 5.0. This technology is facilitating the indeed-development of an intelligent transportation system through multi-functional sensors equipped with on-board computational units (OBU) and Road-side Units (RSU) [1] that cooperate for effective communication and computation, resulting in a new era of Edge-Vehicle technology. Additionally, Transport-as-a-service (TaaS) has become increasingly prominent in the vehicle service industry, with predictions suggesting that by 2030, 95% of U.S passengers will utilize TaaS services through autonomous electric vehicles [2].

Motivation: Assuming an Jetson Xavier NX GPU and lidar are built into an automated vehicle, shown in Fig. 1, the Jetson is programmed with Tensorflow to perform computation and communication on sensor data input. The embedded model is trained and tested on sensor data for further processing, using two different modalities: centre-point (two-stage) and SECOND (one-stage models), which achieved speeds of 39.2 Hz and 18.0 Hz, respectively. The primary challenge lies in achieving inference accuracy within a reasonable time frame for object detection and tracking services, given the limited resources of edge devices. To address this issue, one possible solution is to offload computation-sensitive services to potential devices via WiFi or cellular networks provided by mobile edge computing. While current systems have considered factors such as device service execution, residual resource ratio, and channel/link condition, resulting in moderate accuracy, we believe that service execution time, cost, and utility ratios of requested services from devices and servers are also critical in determining the cost of service-x for effective optimization of service cost. Vehicular edge computing (VEC) is highly sought after in the automotive industry due to its ability to offload computation-intensive services in distributed

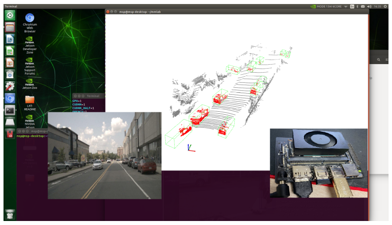


Fig. 1: Tracking simulation analysis on Jetson Xavier NX

computing scenarios. This strategy reduces computation load and improves system performance while maintaining adaptive service reliability [3]. VEC has several advantageous subdomains, including autonomous driving [4], driver assistance, and accident prevention systems. Let's discuss a few basic scenarios illustrated in Fig. 2 regarding the major concern of high mobility in vehicles within Intelligent Transportation System (ITS).

1) Scenario 1: Involves two vehicles equipped with camera and Lidar sensors respectively to record infrastructure for surveillance. Both vehicles are associated with RSU-1 within the region, and if one vehicle fails to receive a response to its service request (object detection or object tracking) within a synchronized time, it causes a delay in service execution
2) Scenario 2: Involves a vehicle associated with RSU-2 within the coverage region, which has raised a service request but not received a response, and has crossed the current coverage region, entering the RSU-1 coverage region. Both scenarios highlight issues with executing the service and handover.

In this regard, an artificial intelligence techniques are commonly used to meet the technology requirements of vehicles, but resource limitations often hinder effective data processing. In such cases, computation offloading is a viable alternative for achieving desired targets of vehicular orchestration, where vehicles/devices allows to offload computation-intensive services to servers which can observe in Fig. 3, subject to limitations on cost and latency-dependent processing capabilities. Some of the challenges include:

1) Choosing a server to handle the service request.
2) Making a decision on whether to offload a portion of the data that the current server is unable to process.
3) Estimating and distributing the necessary resources for executing the service.

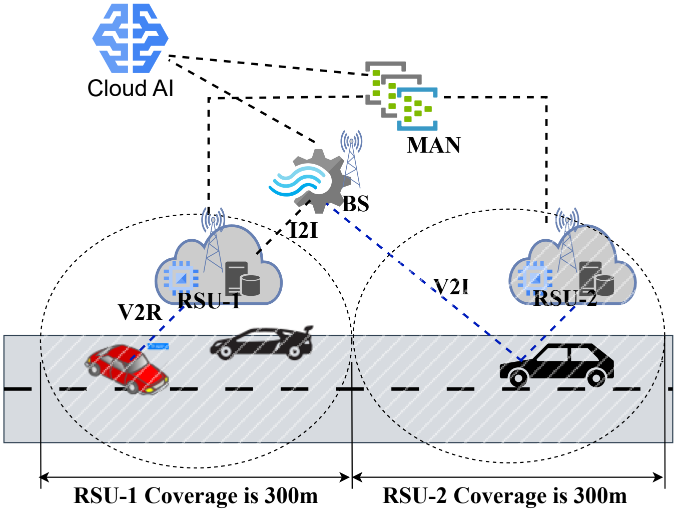


Fig. 2: Cloud-assisted Vehicle Edge-Orchestration

However, the service offloading issue has received global attention from researchers and authors. It is similar to NP-Hard and is addressed through a channel quality and service offload optimization process [5], [6], utilizing Deep Reinforcement Learning (DRL)-influenced techniques and methodologies. Moreover, many existing methods assume that each service request execution occurs within a window slot, whereas in reality, devices and nodes are not static and may leave the server range before completing their requested service. As a result, servers must be conditionally cooperative to offload and migrate computation-intensive services to achieve high performance, posing two global challenges.

1) Optimizing data transmission overhead by reducing offloading decisions.
2) Choosing a potential device or server.

In order to enhance service quality, improve reliability, and minimize data transmission rate overhead, we design a ASXC^2 approach. This approach entails the option to either locally execute the service or select an appropriate device or server for offloading the service and executing it synchronously with the request. The main contributions are as follows

1) Develop a 2-step ASXC^2 approach utilizing the node-centric Lyapunov method and distributed Markov mechanism to optimize the rate of service and offload execution errors.
2) Design a node-centric Lyapunov method that incorporates cost and utility functions, as well as node-centric features, to estimate the service cost prior to offloading to the selected node.
3) Construct a Markov mechanism-inspired service latency prognosticate model for each application request, which includes constructing a correlation matrix between devices and servers, to mitigate the ratio of errors in executing offloaded services.
4) Employ a non-linear programming multi-tenancy heuristic method to predict the service preferences for improving resource utilisation ratio.

The manuscript is organized as follows: Section II provides a brief explanation of the research gaps and problem statements of existing approaches. Section III presents preliminaries of the proposed model. Section IV presents detailed descriptions of the proposed system and its mathematical models, including algorithm. Section V evaluates the simulation outcomes, and Section VI concludes the manuscript.

II. Related Work

This section discusses approaches for offloading services that are both context-aware and Energy cost-efficient, as well as their measurement factors.

A. *Context-aware offloading:*

The authors of [7] developed a joint offloading method that utilizes both cellular and unlicensed channels to minimize communication costs during offloading. They divided the offloading communication process into two steps: vehicle-to-RSU scheduling and Vehicle-to-Infrastructure scheduling. To facilitate the service, they employed a two-sided acknowledgement method that uses the unlicensed spectrum. For the latter, they utilized a DRL model for cellular channel utilization, and while making service offloading decisions, they employed a Double Deep Q-Network (DDQN). In [8], a knowledge-based decision-making method is developed using three parameters: targeted device status, speed, and performance status. In this actor-centric architecture, the agent plays an important role, like where the service should execute, by considering the outcome of the method. However, the drawback is that reward estimation of every timestamp causes the processing delay. In [9], a deep deterministic policy gradient (DDPG) model is developed for optimizing the offloading decisions based on Signal Noise Ratio (SNR) quality as significant criteria and minor is restricted to two-RSU count for service support to avoid delayed communication. Eventually, in [10], the authors designed an IoT-based traffic control environment to map lightweight edge computation requirements. They optimized resource allocation using the DDPG model.

B. Resource-aware Offloading

In [11], a mechanism for reducing data size is developed based on CNN models. It prioritizes offloading intermediate data to an edge server to maximize the quality of service experience. However, this approach does not take into account the challenges posed by real-time environments.

In [12], [13], a joint scheduling and computation offloading (JSCO) method is designed using game theory, considering service execution time, delay, and precedence order as its components. A utility mechanism is employed to sort devices in a predetermined order, resulting in a 54% reduction in resource consumption for local computation and a 37% reduction for offloading-based service computation. However, this strategy is not considered the signal quality during the offloading process.

In [14], [15], a Lyapunov-based service offloading method is designed, considering CPU/cycles requirement and power consumption as its components. It achieves a 64% energy preservation rate compared to state-of-the-art models. However, this approach neglects the information regarding service and communication channels. Some models also incorporate offloading strategies based on energy harvesting [16] and Dynamic Voltage and Frequency Scaling (DVFS) [17] to minimize computation and communication overhead.

In [18], a multi-edge environment is considered to evaluate the performance of the developed offloading method, focusing on energy usage, service reliability, and delay. However, this model has limitations in terms of vertical offloading and mainly concentrates on vehicle-to-vehicle communication. An alternate offloading technique is designed to trade off execution delay and energy usage. A collaborative service offloading model called greedy offloading and resource allocation (GORA) method is proposed in [19], leveraging Q-learning theory to optimize service cost while considering delay and resource constraints. Traditional solutions tend to be nonconvex and result in high complexity. To address this challenge, the author incorporates queuing theory, resulting in a 49% cost reduction. However, the model not be well-suited for complex environments.

C. Channel-aware offloading

In [20], a three-tier architecture is proposed to ensure satisfactory service reliability for end users by leveraging an LTE-backhaul network with the goal of maximizing profit. While the bandwidth allocation problem is convex, the optimization process involving two-steps introduces unusual complexity.

A software-defined network is developed in order to enhance the device service experience, taking into account service utility, edge utility, and channel security components. Moreover, through privacy-aware channel selection [21], the inherent uncertainty is effectively minimized, resulting in a low ratio.

In [22], the authors tackle the NP-hard multi-vehicle service offloading problem and employ game theory to address the distribution challenge. Convergence time and device performance play crucial roles in achieving the Nash equilibrium, but should pay attention for concerns while designing offloading strategy.

1) How many vehicles would benefit through offloading strategy.
2) Offloading strategy should be implemented when the combined energy consumption and execution delay are below the threshold value.

D. Energy-aware Offloading

In [23], a DRL-based scheduling strategy is developed to meet the requirements of RSUs within communication range, while also being energy-efficient. However, the use of non-secure channels can lead to unreliable scheduling and inadequate energy consumption. In [24], a service scheduling policy is developed for optimizing energy usage among multiple RSUs, which showed significant improvements in network lifetime, service execution rate, and delay reduction. Global agents or base stations performed service scheduling using this approach. In [25], developed a Energy-efficient mode selection and resource allocation (EMSRA) model using deterministic policy gradient method to optimise the joint computation-communication issues based on queuing theory, still require attention to meet the requirements of complex-light-weight networks. In [26], a cost-effective server localization (CESL) approach utilizing spectral clustering theory is developed to address the server localization challenge, thereby enhancing network performance. Additionally, they introduce a cost-aware collaborative task execution (CACTE) strategy, leveraging reinforcement learning, to minimize service execution costs while conserving resources (as mentioned in [27]). However, both methods demonstrate remarkable performance ratios and prove optimal for heterogeneous lightweight environments.
Therefore, we design a ASXC^2 approach tackle the concerns at hand. It aims to enhance service quality and reliability by utilizing the node-centric Lyapunov method to break down the cost optimization problem into sub-issues based on node cost and utility functions, as well as node-centric features. Furthermore, a distributed Markov model employment helps to formulate the service-x cost strategy, considering the mobility-correlation between devices and servers. A detailed description

is added in the following sections with theoretical and mathematical proofs.

III. Preliminaries of Proposed Work

In this section, the preliminaries of the proposed method is defined. Let's consider an edge computing orchestration shown in Fig.2, where $k = 1, \cdots, K$ RSU-based servers to meet the service latency constraints of application services denoted as $i = 1, \cdots, I$ from vehicles/devices $j = 1, \cdots, J$. Offloading the computation-intensive services to servers based on resource-constraints and the frequency of invocations is a challenging task to meet their application deadlines. The further constraints are described in bellow sections.

A. Service Tuple

Lets consider two scenarios for service offloading by following Little's law [28] based on M/M/1 queuing method. The first scenario involves searching for nearby potentially computation-rich vehicles to accommodate the services, while the second scenario involves searching for a nearby server that is not overloaded. In both cases, the search is based on three resource factors: computation capacity A_k, residual bandwidth B_k, and storage capacity C_k, which are formulated as a tuple A_k, B_k, C_k. Each service is defined by A_i, T_i, D_i, where A_i is the required computation capacity to execute the requested service i, T_i is the maximum time slots, and D_i is the data size.

B. Communication Cost

Let us assume the communication cost is denotes $\eta\left(i\right)$ which calculates based on amount of offloading input data cost $v^t_{j,j+1}$ (OBU-2-OBU), bandwidth cost $\hbar^t_{j,j+1}$ (OBU-2-OBU) as follows. Similarly, $v^t_{j,k}$ (OBU-2-RSU), bandwidth cost $\hbar^t_{j,k}$ (OBU-2-RSU).

$$\eta\left(i\right) = \rho_{j,k} \cdot \frac{v^t_{j,j+1}}{\hbar^t_{j,j+1}} + \left(1 - \rho_{j,k}\right) \cdot \frac{v^t_{j,k}}{\hbar^t_{j,k}}$$

where $\rho_{j,k}$ offloading decision binary variable, $\rho_{j,k} = 0$ refer the service execution at cloud, else the execution at server.

C. Computation Cost

Let us assume the computation cost is denotes $\psi\left(i\right)$ which calculates based on service resource requirements $\varphi^t_{j,j+1}$ and device resource capacity $\aleph^t_{j,j+1}$ as follows

$$\psi\left(i\right) = \rho_{j,k} \cdot \frac{\varphi^t_{j,j+1}}{\aleph^t_{j,j+1}} + \left(1 - \rho_{j,k}\right) \cdot \frac{\varphi^t_{j,k}}{\aleph^t_{j,k}}$$

The service latency cost is defined as follows

$$\Gamma\left(i\right) = \eta\left(i\right) + z_i \cdot \psi\left(i\right) + \left(1 - z_i\right) \cdot \psi\left(v^t_{j,j+1|j,k}\right)$$

Similarly, the complete service execution cost including latency and energy usage cost is defined as follows

$$\Upsilon\left(i\right) = \zeta_i \cdot \Gamma\left(i\right) + \left(1 - \zeta_i\right) \cdot \mathbb{C}_i$$

where ζ_i is the weight of service completion time which is the tuning parameter plays important role in maintaining the trade-off between latency and energy usage. A larger ζ refers the service i is latency sensitive; otherwise, the service is more energy usage sensitive. $\mathbb{C}_i$, $\Gamma\left(i\right)$ refer energy usage and latency cost, respectively.

Note: The communication between devices and servers utilizes the Orthogonal Frequency-Division Multiple Access (OFDMA) digital modulation scheme. In cases where multiple vehicle devices are connected to a single server, the communication channel is divided equally among the devices to prevent latency. To address service latency, each channel slot handles service requests, and inter-connected communication utilizes service offloading strategies, such as partial or full offloading (illustrated in Fig. 3), when the device requesting the service goes beyond the communication range.

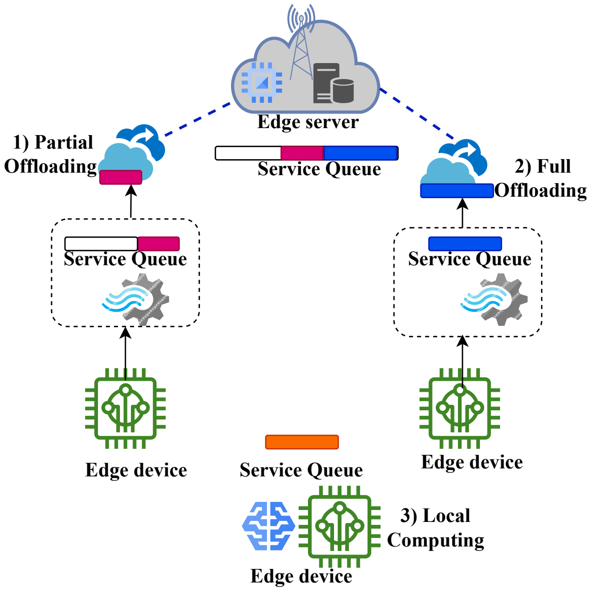


Fig. 3: Service offloading strategies

D. Problem Formulation

Definition 1. *Service offloading occurs when the cost of executing the service is below the threshold value while still meeting latency constraints. This is expressed in the following formulation.*

$$\min \sum_{i \in I} \Upsilon\left(i\right) \tag{1}$$

Subject to

$$C_1 : \textit{Energy cost} \sum_{k \in K} \mathbb{C}^{j,k}_i \leq \mathbb{C}^{j,k}_{i,thr}$$

$$C_2 : \textit{Computation cost} \sum_{k \in K} \varphi^{j,k}_i \leq \aleph^{j,k}_{i,thr}$$

$$C_3 : \textit{Communication cost} \sum_{k \in K} v^{j,k}_i \leq \hbar^{j,k}_{i,thr}$$

$$C_4 : \textit{Binary offloading variable } \rho_{j,k} \epsilon \{0, 1\} \;\; \forall i \epsilon I, \forall j \epsilon J$$

The aim of designing adaptive offloading strategy is to minimise the service execution cost subject to the listed constraints and the same theme is illustrated in Fig.4. This model is derived based on two factors like energy cost and latency. The server capacity must be higher than the requested service requirement capacity to ensure proper functioning. The probability of service offloading $\rho_{j,j+1|j,k}$ is used to derive the local computing queue, which is based on the service arrival rate χ_j and can be calculated as $\chi_j^{lc} = (1-\rho_{j,j+1}) \cdot \chi_j$. To estimate the offloading queue χ_j^{o} of the server, which is based on the service arrival rate χ_k, the probability of service acceptance by the server $\rho_{\hat{j,k}}$ is taken into account. This probability is determined by considering factors such as latency, resource, and storage capacity, and can be expressed as $\hat{\rho}_{j,k}$.

Possible constraints, including cost and latency, that may arise when executing a service request, are illustrated in Fig. 4. The decision threshold level is denoted as *Max*. In line with the aspiration of edge computing to achieve high system performance and service reliability, *local computing* is always preferred, *even if the energy cost violates the energy threshold value. Partial offloading* is preferred when both constraints exceed the threshold value and the average latency violation rate is high. On the other hand, *full offloading* is preferred when both factors show abnormal violation rates. Generally, the characteristics of the policy for offloading service requests by vehicles and the policy for accepting offloaded service requests by servers are partially similar.

Theorem 1. *Minimising the offload-service completion cost issues (1) is similar to NP-hard.*

$$\min \sum_{i \in I} \Upsilon\left(i\right)$$

Proof. Our aspiration is to optimise the service execution window time with low cost. Let's assume, the offloading cost can be projected as $0-1$ Knapsack problem, and the knapsack problem is NP-complete likewise, the services treated as items, and the offloading gain which we treated as item rank or weight (recommend to refer our previous contributions [29]) should be less than the server rank and weight with respect to the capacity. Therefore, targeted issue is NP-hard since $0-1$ Knapsack problem is NP-complete. □

IV. Adaptive Service-X Cost Consolidation based service offloading

In this section, the objective function is formulated in detail by splitting it into subsections as follows.

A. Lyapunov based Cost Optimization

In this subsection, the selection of local computing, full, and partial offloading is derived based on the status of vehicle device including consistent cost, utility factors evaluated with the base of node-centric lyapunov method as follows. Moreover, the same communication model [3] is used for communication between devices and servers.

$$\phi_{r,v}^{t} = \hbar \log_2 \left(\frac{1+\phi_{po}^{v} g_{r,v}^{t}}{(np)^2} \right) \tag{2}$$

The energy consumption of the requested service is estimated based on several factors, including the channel bandwidth capacity represented by $\hbar$, the transmission energy of the vehicle device denoted by ϕ_{po}^{v}, as well as the Gaussian noise power and channel gain, denoted by np and $g_{r,v}^{t}$ respectively.

$$\xi_{j,j+1|j,k}^{t} = \rho_{j,k} \cdot \left(\xi_i^{\text{lc}} + \xi_{j,k}^{\text{o}}\right) + \left(1-\rho_{j,k}\right) \xi_{j,k}^{\text{fo}} \tag{3}$$

Moreover,

$$\xi_i^{\text{o}} = x_i \cdot \xi_i^{\text{lc}} + (1-x_i) \cdot \left(\xi_i^{\text{lc}} + \xi_i^{\text{po}}\right), \xi_i^{\text{lc}} = \xi_{uc} \cdot \frac{D_i}{\aleph_{j,j+1}^{t}}$$

where $x_i \epsilon \{0,1\}$ indicates the binary decision variable for reusing the local computing data during the partial offloading execution process, here 0 indicates reusing the locally executed data; else 1. Therefore, the partial offloading is defined as follows

$$\xi_{j,j+1|j,k}^{\text{po}} = \xi_{\text{uc}} \cdot \left(\frac{D_i - v_{j,j+1|j,k}^{t}}{\aleph_{j,j+1}^{t}} \right) + \phi_{\text{po}}^{v} \cdot \frac{v_{j,j+1}^{t}}{\hbar_{j,j+1}^{t}}$$

Similarly for full offloading,

$$\xi_{j,j+1|j,k}^{\text{fo}} = \xi_{\text{uc}} \cdot \frac{D_i}{\aleph_{j,j+1}^{t}} + \phi_{\text{po}}^{v} \frac{v_{j,j+1}^{t}}{\hbar_{j,j+1}^{t}} + \xi_{\text{cloud}}^{t} \cdot w_{A_k|B_k|C_k}^{i}$$

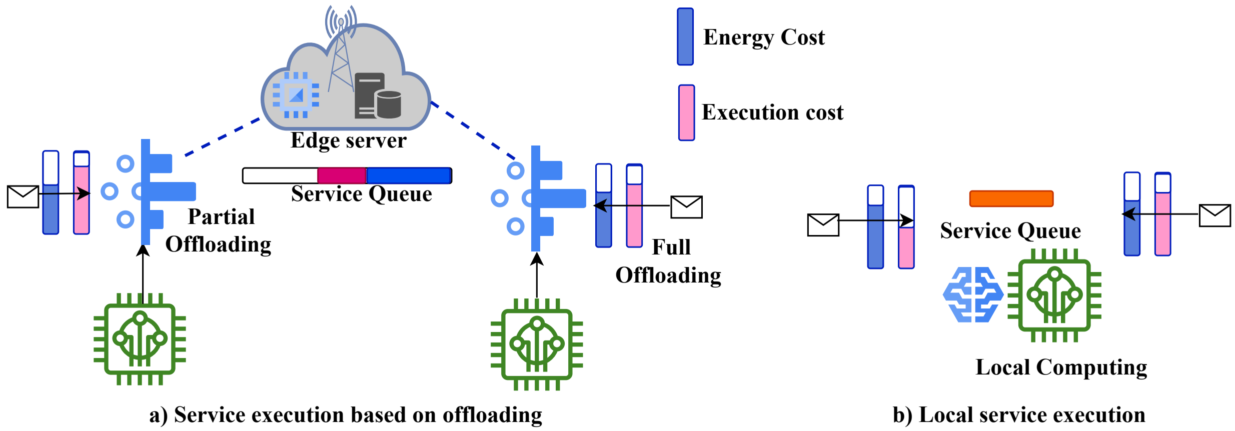
The energy usage of a service request is calculated using the following notation: ξ_i^{lc} represents the energy consumption (EC) for local computing, $v_{j,j+1|j,k}^{t}$ represents the amount of offloading data, $\xi_{j,j+1|j,k}^{fo}$ represents the energy consumption for offloading communication, ξ_{uc} represents the unit cost of EC, and ξ_{cloud}^{t} represents communication energy usage base-station or cloud.

$$\mathbb{C}_i^{j} = \sum_{j=1}^{J} \sum_{k=1}^{K} \left[a_i \cdot \xi_{j,j+1|j,k}^{t} + b_i \cdot w_{A_k|B_k|C_k}^{i} \cdot \phi_{po}^{v} \cdot L_k \right] \tag{4}$$

where $\rho_{j,j+1|z,j}$ represent the probability of service offloading with binary, a_i and b_i denote the energy weights during local execution and service offloading, respectively, taking into consideration the channel EC. Similarly, $w_{A_k|B_k|C_k}^{i}$ denotes the cost of each resource unit, and L_k represents the predicted server load.

B. Linear-programming based multi-tenancy model for utility optimization

In this subsection, the linear-programming based multi-tenancy model is formulated based on service and server utility, since this two factors play vital in mapping the offloading process. Basically, the sequential process in the traditional single-tenancy model fails to meet latency constraints, because of not balancing the trade-off between resource usage and low latency. As a solution, the employing a multi-tenancy model achieves an optimal resource distribution using a linear stochastic algorithm, resulting in a higher utility ratio. The server and service utility factors are derived as follows.



a) Service execution based on offloading

b) Local service execution

Fig. 4: Service execution constraints

1) Server Utility: In our mapping strategy, the service request cost and its utility is estimated prior allocation to enhance the offloading efficiency when the server is not overloaded. The classification of over-load and under-load servers are derived as follows.

$$\text{server} = \begin{cases} \text{is overloaded} & \text{for} \quad 0 \leq \widehat{\delta_k} < 0.5 \\ \text{is not overloaded} & \text{for} \quad 1 \geq \widehat{\delta_k} > 0.5 \end{cases}$$

In the iterative process, the probability of server utility is calculated as follows.

$$\widehat{\delta_k} = \frac{1}{\Gamma(i) + \left(p_{j,j+1|j,k} - r^i_{A_k|B_k|C_k}\right)} \tag{5}$$

Where $p_{j,j+1|j,k}$ refer execution-cost by vehicle for the offloaded service, $r^i_{A_k|B_k|C_k}$ refer server usage cost. When the service is offloaded, the estimated cost payable by the Vehicle for effective execution is as follows

$$p_{j,j+1|j,k} = 0.5^{\frac{1}{n_k}} \times \frac{v^t_{j,j+1|j,k}}{\hbar^t_{j,j+1|j,k}} \tag{6}$$

The estimated usage cost of the server includes receiving, executing, and reverting process towards a requested vehicle, including service migration when the vehicle moves out of the communication range. In this estimation, n_k represents the number of devices connected to the server, and $v^t_{j,j+1|j,k}$ represents the amount of offloading data.

$$r^i_{A_k|B_k|C_k} = \sum_{i=1}^{I} \psi(i) \times w^i_{A_k|B_k|C_k} + \beta_k \log(1 + n_k) \tag{7}$$

For further information, please refer to [30], [31], where β_k represents the weight factor, which calculates for each server based on its rank.

2) Service Utility: The service utility calculation is defined based on predicted-communication cost $\widehat{\eta}(i)$, predicted-computation cost $\widehat{\psi}(i)$, data similarity ratio $\wp(i)$ of i^{th} service as follows.

$$\widehat{\delta_i} = \frac{1}{\widehat{\eta}(i) + \widehat{\psi}(i) + \wp(i)} \tag{8}$$

The processing data similarity ratio calculation is defined based on number of input (ind) and distinct (did) data as follows

$$\wp(i) = \frac{1}{1 + e^{-\frac{\text{ind}}{\text{did}}}} \tag{9}$$

Service mapping among the servers is carried out based on utility ratio (i.e., the classified services list (please refer [30], [31] for classification details) and suitable servers are sorted based on the utilities to avoid service completion hiccups for accomplishing the targeted offload efficiency and completion ratio) to meet the application deadlines. Lets derive the offloading issues in context-perspective as follows

Theorem 2. *Assume a vehicle is prepared to offload the service, the predicted resource size required for efficient service execution is as follows.*

$$\widehat{v}^t_{j,j+1|j,k} = \frac{a_i \xi_{uc} \hbar \xi^t_{uc}}{\log_2\left(a_i \xi^t_{uc} + b_i w^i_{A_k|B_k|C_k}\right)} - \frac{1}{|J|} \cdot \frac{g^t_{j,j+1|j,k}}{(np)^2} \tag{10}$$

Proof. Assume, $\rho_{j,k} = 1$ and Eq. 5 is derived based on Eqs. 6,7,10 as follows where $y = a_i \xi^t_{uc} + b_i w^i_{A_k|B_k|C_k}$

$$\partial\left(\frac{\delta_k}{v^t_{j,j+1|j,k}}\right) = \frac{a_i \xi_{uc} \hbar \xi^t_{uc}}{\log_2\left(\frac{g^t_{j,j+1|j,k}}{(np)^2 + v^t_{j,j+1|j,k}}\right) + y} \tag{11}$$

To assess the greater than zero value $\partial\left(\frac{\delta_k}{v^t_{j,j+1|j,k}}\right) \geqslant 0$, let again apply partial derivation

$$\partial^2\left(\frac{\delta_k}{v^t_{j,j+1|j,k}}\right) = \frac{a_i \xi_{uc} \hbar \xi^t_{uc}}{\log_2\left(\frac{g^t_{j,j+1|j,k}}{(np)^2} + v^t_{j,j+1|j,k}\right)^2} \tag{12}$$

Here, the value of Eq. 12 is greater than zero. Our goal is to minimize the cost of the service offloading request, while keeping the server resource unit cost constant. To achieve a finite resource size for the service offloading request, and to ensure that the server capacity is high, we set the value of Eq.

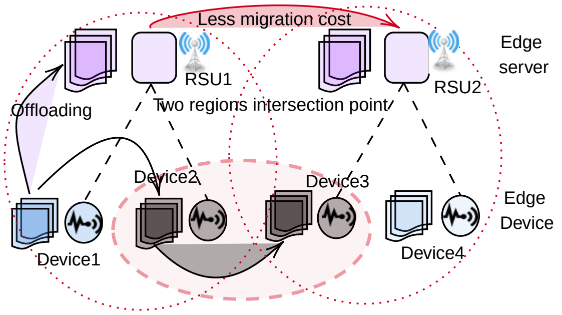


Fig. 5: Service migration formulation

11 to zero. So, $\partial\left(\frac{\delta_k}{v^t_{j,j+1|j,k}}\right) = 0$, and the service offloading request policy is derived as follows

$$\begin{aligned}
&= \frac{a_i \xi_{uc} \hbar \xi^t_{uc}}{\log_2\left(\frac{g^t_{j,j+1|j,k}}{(np)^2} + v^t_{j,j+1|j,k}\right)} + a_i \xi^t_{uc} + b_i w^i_{A_k|B_k|C_k} \\
&= \frac{a_i \xi_{uc} \hbar \xi^t_{uc}}{\log_2\left(a_i \xi^t_{uc} + b_i w^i_{A_k|B_k|C_k}\right)} - \frac{1}{|J|} \cdot \frac{g^t_{j,j+1|j,k}}{(np)^2} \\
&= \widehat{v}^t_{j,j+1|j,k}
\end{aligned} \tag{13}$$

Therefore, R.H.S= L.H.S, where the derivation confines that, the requesting service cost is low with maximum server utility. □

Corollary 1. *Optimizing the cost of service migration is a crucial aspect, especially when a device moves out of communication range, as this helps to improve server utility. The service allocations are stored in a matrix format, which can be represented as follows:*

$$\rho^t_{j,j+1|j,k} = \begin{bmatrix} \rho^1_1 & \dots & \rho^K_1 \\ \dots & \dots & \dots \\ \rho^1_J & \dots & \rho^k_j \end{bmatrix} \tag{14}$$

Fig. 5 depicts the significance of service migration when the vehicle requesting the service moves beyond the communication range of the responding server. In such scenarios, optimizing communication cost becomes crucial, which involves considering factors such as distance, the resource potential of the targeted server, utility rate, and trade-offs between vehicles. This approach helps to achieve high performance and service reliability. For instance, if $device_2$ requests a service from $server_1$ but moves out of the region, $server_2$ is selected based on the above constraints.

$$\begin{aligned}
\sigma\left(\mathbb{C}^{i,t}_{k,k+1}\right) &= \frac{e^{\mathbb{C}^{i,t}_{k,k+1}}}{\sum\limits_{k\in K} e^{\mathbb{C}^{i,t}_{k,k+1}}} \\
\mathbb{C}^{i,t}_{k,k+1} &= \sum_{k\in K} \rho^{i,t}_{j,k} \times \mathbb{C}^{i,t}_{Trans} \times \delta_k
\end{aligned} \tag{15}$$

$$\mathbb{C}^{i,t}_{Trans} = \begin{cases} \phi^v_{po} \xi^t_{uc} \times d_{k,k+1}\left(\varepsilon D_i\right) \therefore k \neq k+1 \\ 0; \text{otherwise} \end{cases} \tag{16}$$

Migration occurs when the value of $\sigma\left(\mathbb{C}^{i,t}_{k,k+1}\right)$ for the server exceeds 0.5; otherwise, a nearby node is recursively searched. The Lyapunov optimization approach maintains a balance between server usage ratio and service queue consistency. The Lyapunov utility method can be defined as follows:

$$= q^t_{j,k}\mathbb{C}^{t,i}_{j,k} + \chi^o_{j,k} + L_k \tag{17}$$

Theorem 3. *The nonsuccessive service ratio ϖ^t_k is a other crucial factor in determining server performance, as stated in Corollary-1, which outlines the following derivation for server performance.*

$$\begin{aligned}
&q^t_{j,k}\mathbb{C}^{t,i}_{j,k} + \chi^o_{j,k} + L_k \leqslant \\
&\sum_{k=1}^{K}\left(q^t_{j,k}\delta^t_k + B_k - \varpi^t_k\right) + \frac{1}{|I|}\log\left(v^t_{j,j+1|j,k}\right)
\end{aligned} \tag{18}$$

Proof. To maximize utility and minimize costs, it is essential to ensure that the server storage capacity, communication costs, and satisfaction of Eq. 1 are all met, along with the nonsuccessive service ratio. It is worth noting that optimizing the utility and cost of communication can be achieved by maintaining a server storage capacity (B_k) that is nearly equal to the nonsuccessive service queue.

$$q^t_{j,k}\mathbb{C}^{t,i}_{j,k} + \chi^o_{j,k} + L_k \leqslant \sum_{k=1}^{K}\left(q^t_{j,k}\delta^t_k + B_k - \varpi^t_k\right) \tag{19}$$

The server has to fulfil the below condition to maintain the consistent sustainability of server usage.

$$\begin{aligned}
&\sum_{t\in T}\sum_{k\in K} q^t_{j,k}\mathbb{C}^{t,i}_{j,k} + \chi^o_{j,k} + L_k \leqslant \\
&\sum_{k=1}^{K}\left(\delta^t_k + B_k\right) + \frac{1}{|I|}\log\left(v^t_{j,j+1|j,k}\right) \\
&+\left(\sum_{k=1}^{K}\left(\frac{\chi^o_{j,k}\left(\delta^t_k - \delta^{t-1}_k\right)}{B_k}\right) - \varpi^t_k\right)
\end{aligned} \tag{20}$$

When there is no service migration (responsive server itself has completed the service execution), and the system utility is

$$\begin{aligned}
&\sum_{t\in T}\sum_{k\in K} q^t_{j,k}\mathbb{C}^{t,i}_{j,k} + \chi^o_{j,k} + L_k \leqslant \\
&\sum_{k=1}^{K}\left(\delta^t_k + B_k\right) + \frac{1}{|I|}\log\left(v^t_{j,j+1|j,k}\right) \\
&+\sum_{k=1}^{K}\left(\frac{\chi^o_{j,k}\delta^t_k}{B_k} - \varpi^t_k\right)
\end{aligned} \tag{21}$$

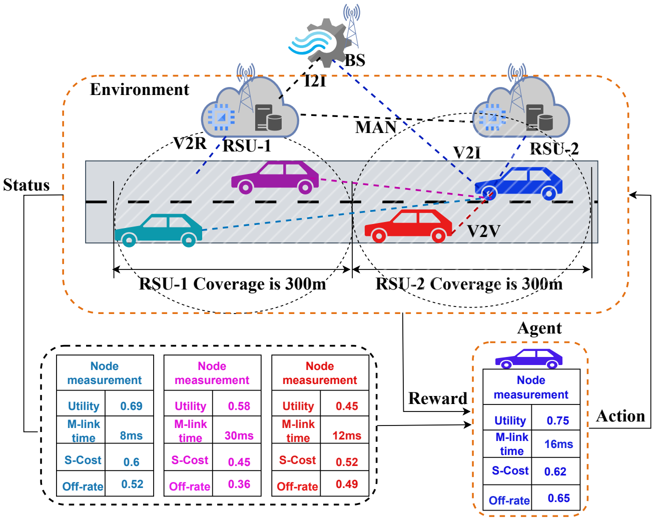
When nonsuccessive service ratio ϖ^t_k is equal to *zero*, then the performance is

$$\begin{aligned}
&\sum_{t\in T}\sum_{k\in K} q^t_{j,k}\mathbb{C}^{t,i}_{j,k} + \chi^o_{j,k} + L_k \leqslant \\
&\sum_{k=1}^{K}\left(\delta^t_k + B_k\right) + \frac{1}{|I|}\log\left(v^t_{j,j+1|j,k}\right) \\
&+\sum_{k=1}^{K}\left(\frac{\chi^o_{j,k}\delta^t_k}{B_k}\right)
\end{aligned} \tag{22}$$

□

```
Algorithm 1: Node centric-heuristic Approach
input : server set $\{k = 1, \cdots, K\}$, vehicle device set
        $\{j = 1, \cdots, J\}$, services set$\{i = 1, \cdots, I\}$
output: Cost effective service offloading
1  Let $\phi^{v}_{po} \neq 0$, $g^{t}_{r,v} \neq 0$, $\xi^{LC}_{i} \neq 0$, $\xi^{t}_{uc}$, $\rho_{z,j} \epsilon \{0, 1\}$,
     $\gamma^{\tau}_{j,j+1|j,k} \neq 0$, $\delta^{i}_{j,k} \neq 0$;
2  for each $j \in \boldsymbol{J}$ do
3      Estimate energy consumption
         $\xi^{t}_{j,j+1|j,k} = \rho_{j,k} \cdot \left(\xi^{\mathrm{lc}}_{i} + \xi^{\mathrm{o}}_{j,k}\right) + (1 - \rho_{j,k})\, \xi^{\mathrm{fo}}_{j,k}$;
4      Estimate final energy cost of each service
5      $\mathbb{C}^{j}_{i} =$
       $\sum_{j=1}^{J} \sum_{k=1}^{K} \left[a_i \cdot \xi^{t}_{j,j+1|j,k} + b_i \cdot w^{i}_{A_k|B_k|C_k} \cdot \phi^{v}_{po} \cdot L_k\right]$;
6      Estimate server usage cost to execute the service as
         follows
7      $r^{i}_{A_k|B_k|C_k} =$
       $\sum_{i=1}^{I} \psi(i) \times w^{i}_{A_k|B_k|C_k} + \beta_k \log(1 + n_k)$;
8      # Estimate the final cost $\Upsilon(i)$;
9      while $\Upsilon^{k}_{j}(i) \leqslant \Upsilon^{j,k}_{th}(i)$ do
10         Estimate server utility
11         $\widehat{\delta_k} = \frac{1}{\Gamma(i) + \left(p_{j,j+1|j,k} - r^{i}_{A_k|B_k|C_k}\right)}$;
12         Estimate link quality $\gamma^{\tau}_{j,j+1|j,k}$ using Eq.23;
13         if $\delta^{i}_{j,k} \leqslant \delta^{i,j,k}_{th} || \gamma^{\tau}_{j,j+1|j,k} \leqslant \gamma^{i,j,k,\tau}_{th}$ then
14             Offload the service to nearby server;
15             Update $\hat{\rho}^{t}_{j,j+1|j,k} = \begin{bmatrix} \rho^{1}_{1} & \dots & \rho^{K}_{1} \\ \dots & \dots & \dots \\ \rho^{1}_{J} & \dots & \rho^{k}_{j} \end{bmatrix}$
16         end
17         else
18             Go to step-3 and iterates till the selection
                 of suitable server with targeted value.
19         end
20     end
21 end
```

Therefore, always the server performance is dependent on amount of services arrived and nonsuccessive ratio. Algorithm 1 outlines a method for improving system performance and service reliability by using node-centric measurements to determine service offloading decisions. The first line defines communication and energy consumption factors, while the second line checks the status of all devices based on demand. The third line defines the energy consumption factor, and line 5 predicts the cost of each service, along with the server usage cost, which is evaluated using line 8. If the threshold value is met, server utility is estimated using line 11, and all servers within vehicle communication range are checked using line 12. If more than two servers can accommodate the service, the residual resource capacity is examined to make a decision, taking into account vehicle directions in the simulation. Line 14 updates the matrix for the next service execution to evaluate global decisions. If a server is not finalized, the process continues to iterate from line 3.



Node measurement	
Utility	0.69
M-link time	8ms
S-Cost	0.6
Off-rate	0.52

Node measurement	
Utility	0.58
M-link time	30ms
S-Cost	0.45
Off-rate	0.36

Node measurement	
Utility	0.45
M-link time	12ms
S-Cost	0.52
Off-rate	0.49

Node measurement	
Utility	0.75
M-link time	16ms
S-Cost	0.62
Off-rate	0.65

Fig. 6: Service cost-effective device selection for service offloading learning

C. Mobility impact on offload-service execution

In this subsection, vehicle mobility impact is formulated based on vehicle density, speed, direction, communication range, and service cost to optimise the average offloading efficiency. The primary reason for the lack of success in service offloading objectives is the high mobility of vehicles, leading to data transmission failures or disturbances when the server's execution time is prolonged [32]. Therefore, ensuring a short transmission time for the link has become crucial in designing a cost-effective service offloading strategy, which is one of the main goals addressed in this article. To achieve this, the article explores the application of frequency-modulated continuous (FMCW) and multiple frequency-shift keying (MFSK) waveform techniques, taking into account the vehicle speed and the average time required to move out of the server's range. The mathematical derivation of the short transmission link time is presented as follows.

$$\gamma^{\tau}_{j,j+1|j,k} = \begin{cases} \frac{\Re + \varpi_{j,j+1|k}}{\left|\varkappa_j - \varkappa_{j+1|k}\right|} & \text{for} \quad \varkappa_j > \varkappa_{j+1|k} \\ \frac{\Re + \varpi_{j,j+1|k}}{\left|\varkappa_j - \varkappa_{j+1|k}\right|} & \text{for} \quad \varkappa_j \leq \varkappa_{j+1|k} \end{cases} \tag{23}$$

When the direction of vehicles are not same, the transmission channel link quality

$$\gamma^{\tau}_{j,j+1|j,k} = \frac{\sqrt{\Re^2 - \left(y_j - y_{j+1|k}\right)} + \varpi_{j,j+1|k}}{\left|\varkappa_j + \varkappa_{j+1|k}\right|} \tag{24}$$

Where $\gamma^{\tau}_{j,j+1|j,k}$ is transmission link between j^{th} vehicle, $j+1^{th}$ vehicle or between j^{th} vehicle, k^{th} server over the network. Here, $\varkappa_j$ is speed of j^{th} vehicle, $\Re$ indicates coverage range, ϖ_j, $\varpi_{j+1|k}$ is the distance between vehicles and server, respectively; the Euclidean distance is used to calculate distances among vehicles and servers as well. In this scenario, we select the offloading device based on a specific condition: the probability of service execution elapsed time being less than the probability of transmission link time.

D. Functionality

A systematic approach is followed to design the decisions about computation offloading. This process involves allocating a specific channel time for two groups divided by their queue length and required resources based on service and server utility ratios for effective service execution. Computation-intensive services are permitted to offload, while the remaining services are executed locally due to limited computation resources on the edge server, as can observe in Fig.6. Once an edge device is authorized to offload, a joint resource provision method is initiated to evaluate the server utility and anticipated resource rate for the service execution. The queue waiting for service length and antenna usage components are used to leverage each device's performance in our simulation. The utility weight factor is used to prioritize the energy preservation and execution delay of the channel.
Subsequently, our model has developed in which the base station acts as the agent and the vehicular network as the environment. The agent received input on the RSU profile, including the remaining battery backup, vehicle distance from the RSU, connected devices, and service waiting time. If the RSU offloaded the service to a vehicle, the reward estimated the transmitted service size in bits, and the energy required to receive the service was also evaluated. Here, the RSU is reliable for satisfying requests when the requested vehicle is left the RSU range or if the concerned RSU is not completed the download request.

E. Complexity analysis

In this subsection, our algorithm complexity is derived as follows. The algorithm 1 is with three primary objectives, each have its own complexity. For instance, selecting a potential server based on delay-sensitive service complexity is $O\left(n\right)$, while sorting the services according to latency demand, complexity is $O\left(n\log_2 n\right)$. Therefore, the overall complexity is like $O\left(n\log_2 n\right) + O\left(n\right)$ Let's consider that each server, denoted as k, can provide services from multiple devices, represented by j. The estimation of each server's capability, denoted as β, is performed multiple times. The count of communication request and response events for each service execution determines the communication overhead of the server, which is represented by ϱ_k based on latency demand. The approximate space complexity can be calculated as $= k\times(\varrho_k + 2\varrho_k)\times\beta = 3k\cdot\varrho_k\cdot\beta$ The complexity of STOA models are defined like $O\left(n^3\right)+O\left(n\log_2 n\right)$ for Static scheme [14] , $O\left(n^2+n\right)$ for Game-based [12], $O\left(n^3\right)+O\left(n\right)$ for GORA [19], $O\left(n^2\right)+O\left(n\right)$ for EMSRA [25], respectively.

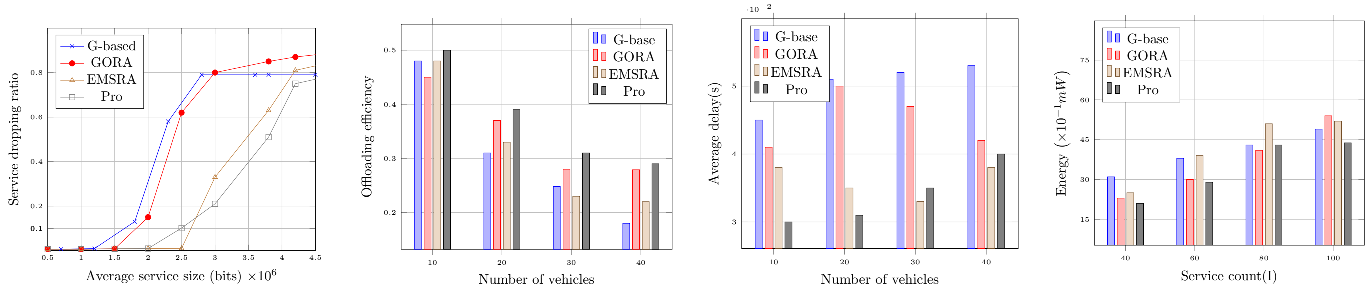
V. Simulation Results

In our simulation, the environment is enabled 5 servers, 10-25 devices. Each server has a radius of 250 m. Each device is randomly dispersed within a communication range of approximately 3 servers, covering an area of 1000 × 1000m. The wireless channel mode is employed for efficient communication in this setup, and a bunch simulation parameters are listed in Table I.

Fig. 7(a) illustrates the relationship between service dropping ratio and average service size. We can see our model achieves the lowest dropping ratio compare to SOTA models. Our model dropping ratio is increased from 2×10^6 service size though it is less than 0.8% as continues increase in size. The relationship between number of vehicles and offloading efficiency is shows in Fig. 7(b), indicating that our approach achieves better offloading efficiency compared to SOTA approaches. We noticed that, offloading efficiency is increased better as the increase of vehicle density due to regulating the server computing capacities to optimize the workload. However, existing approaches prioritize energy optimization in respective resource usage rates rather than conceding latency in this scenario. Fig. 7(c) shows the latency analysis report while the offloading request increase eventually. The vehicle density [10-40] is considered in initial simulation where latency rate is significantly efficient, but the latency increases as the device count rises for G-based model, becasue the server assets are not fully utilized under high offloading ratio, even when considering high density rate and service arrival rates. Fig. 7(d) shows the relation between energy usage and service count based on active devices, revealing a significant drop in energy usage when the server count increases. The energy usage is drastically reduced even as the service arrival rate and offloading rate rise because the Lyapunov method effectively controls the exploration and exploitation ratio for optimizing resource utility. Furthermore, the device is constrained by connecting less than three servers ($\leqslant$3) for avoiding resource starvation. In many cases, services that require limited computing capacity have been offloaded to the server for efficient execution in SOTA models. However, it should be noted that the communication latency increases due to the instability of the device communication range during the response catch from the server towards sub-services. Fig.

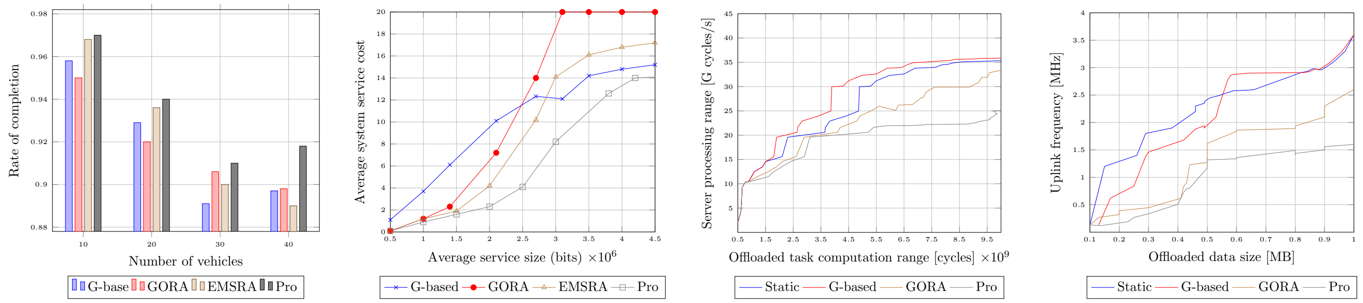
TABLE I: Simulation parameters and their values

Parameter	Value	Parameter	Value
server configuration		**vehicle configuration**	
transmit power	30 dBm	**transmit power**	23 dBm
CPU frequency	50 GHz	**CPU frequency**	3 GHz
antenna gain	8dBi	**antenna gain**	3dBi
receiver noise	9dB	**receiver noise**	5dB
Parameter		**Value**	
delay weights u$_i$		0.6+$\frac{0.01}{t_n^{\text{deadline}}}$	
Packet size		[100,300]kb	
thermal noise power		-114dBm	
energy coefficient		10^{-25}	
energy usage weights		1-u$_i$	
server cache capacity		5Gb	
service-size		15-20 Mb	
time variance		4.5×10^{-4}s	
total bandwidth		20MHz	
vehicle mobility speed		[60,80]km/h	
path-loss model, d in km		128.1+37.6$\log_{10}^{d}$	
radius		250m	
CPU cycles of each service		[10,30]Megacycles	



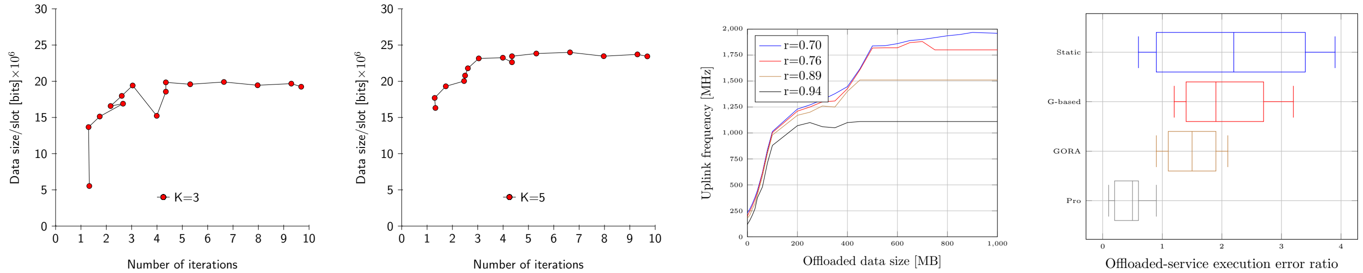
(a) Service dropping analysis (b) Offloading efficiency (c) delay analysis (d) Energy usage analysis

Fig. 7: Performance analysis in terms of service dropping ratio, offloading efficiency, delay



(a) Rate of completion (b) Cost analysis (c) Server capability analysis (d) Uplink frequency analysis

Fig. 8: Performance analysis of SOTA models based on rate of completion, cost, and frequency analysis



(a) Data collection when $K = 3$ (b) Data collection when $K = 5$ (c) Convergence analysis (d) Error analysis

Fig. 9: Performance analysis with partial connectivity and offloading strategy error analysis with max positive mobility +6dBm

8 presents the performance analysis of state-of-the-art (SOTA) models across four different measurements. Fig. 8(a) shows the relation between the rate of service completion and vehicle density. Our model achieves better rate of completion performance than other models as the vehicle density continues to increase. However, the EMSRA achieves better results than our model when the vehicle density ranges from 10 to 20. Additionally, the GORA model achieves better results with a density range of 30 to 40 because they have used graph theory to optimize latency and the offloading design strategy. Similarly, Fig. 8(b) shows the relation between service cost and size. In this case, the GORA model achieves a constant cost from a size of $[3-4.5]\times10^6$. However, our model achieves better cost performance than the SOTA model for sizes ranging from $[0.5-4.5]\times10^6$, and the G-based model performs even better than the other two models as the service size increases.

Fig. 8(c) showcases the analysis of server processing capability. The proposed method demonstrated a lower usage of computing resources due to an effective resource requirement prediction mechanism based on the node-centric Lyapunov method using a Markov decision approach. However, the demand for G cycles/s for the proposed method, GORA, G-based, and static methods was 21%, 20%, 23%, and 30%, respectively, when the task size is 4×10^9. As the task size increased for SOTA models, the resource demand scale changed significantly. Moreover, in Fig. 8(d), the analysis of uplink frequency usage using packet size is depicted. The proposed method achieved a low average frequency demand for a service offload size of 1MB, specifically for small-sized data packets. Conversely, the uplink frequency demand for SOTA models increased as expected. For instance, for a 1MB data size, the demanded frequencies for the proposed method, GORA, G-based, and static methods were 1.52MHz, 2.56MHz, 3.5MHz, and 3.6MHz, respectively. Fig. 9(a) presents the analysis of data computation with respect to the number of iterations. The sub-figures illustrate the data computation time in different scenarios, where the k value scales from 3 to 5. Fig 9(b) showcases the average data size (in bits) to be shared over the network. It is evident that the data size significantly increases from the first iteration, and from the sixth iteration onwards, there is consistency in size. The proposed algorithm aims to reduce the cost by leveraging the device/RSU measurements before selecting potential RSUs for consistent frequency, and minimizing computation-communication overhead. The use of the node-centric Lyapunov method has had a significant impact on performance due to its ability to predict service costs during service offloading. Additionally, the distributed Markov model is effectively generalized the correlation between devices and servers and construct the coefficient matrix to facilitate effective offloading. The non-linear programming multi-tenancy heuristic method is played a crucial role in optimizing the resource utilization rate based on the service preference prediction method and the results are represented in Table.II and Table.III.

TABLE II: Computation-Communication tuning parameters analysis of our model

Communication-based			**Computation-based**	
Transmitted data (kb)	**Average Offloading Efficiency**	**Rate of Completion**	**CPU Cycles (Megacycles)**	**Average Offloading Efficiency**
100	0.4	0.94	**10**	0.17
150	0.36	0.93	**15**	0.26
200	0.31	0.92	**20**	**0.32**
250	0.27	0.91	**25**	**0.29**
300	0.22	0.90	**30**	0.27
350	0.18	0.84	**35**	0.16
400	0.13	0.82	**40**	0.058

TABLE III: Overall performance of SOTA models when k=4, [20-25] devices, transmitted data 560kb.

Model	**Latency [ms]**	**Service dropped ratio**	**Average offloading efficiency**	**Rate of completion**
G-based[12]	896	0.79	0.45	0.21
GORA[19]	931	0.86	0.56	0.39
CESL[26]	823	0.81	0.72	0.56
EMSRA[25]	889	0.83	0.36	0.45
CACTE[27]	790	0.78	0.18	0.67
Pro	329	0.76	0.12	0.81

A. Convergence Analysis

In this subsection,the convergence of the proposed method is analyzed in the context of the proposed environment, and its characteristics are discussed earlier in this section. In order to facilitate effective communication among connected devices, the transmission links are randomly generated with a probability range of *[0.7, 1]*. The DMDP-inspired algorithm aims to achieve an efficient convergence ratio by gathering a set of actions and responses from the environment, as depicted in the figure below. Fig. 9 depicts the performance analysis of the proposed environment with 1000 iterations considering partial coverage connectivity range [0.70,0.95] and offloading strategy. The other side shows an average error analysis based on vehicle mobility for offloading strategy analysis. The proposed method achieved the desired convergence ratio within 400 iterations with a coverage of 0.95%. It exhibited a low error failure rate, meeting the application latency requirements compared to state-of-the-art (SOTA) methods. In Fig. 9(c), we observe the convergence analysis of the proposed environment under different coverage ranges. As the number of interactions increases, the rewards count also increases. However, beyond 400 iterations, the reward count stabilizes when r is set to 0.89 and 0.94 (high coverage range). Conversely, the rewards continue to increase with an increase in the iterations count.

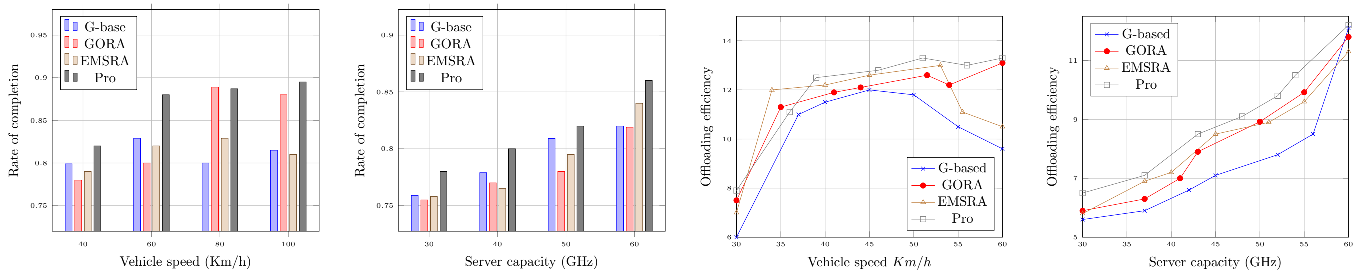
Fig. 9(d) presents the error analysis with maximum positive mobility (+6dBm) for both the proposed method and SOTA methods. The proposed method achieved a low failure rate in meeting the latency deadline for the application service. This accomplishment is attributed to leveraging node-centric measurements, such as service execution cost and execution time, based on the vehicle's direction, range, and speed. The average mean error rates for the proposed method, GORA, G-based, and static methods were 0.55%, 1.5%, 1.9%, and 2.2%, respectively.

B. Tuning parameters analysis

This subsection describes the proposed model performance based on server residual computing capacities, impact of speed on data transfer and their required CPU cycles to execute the offload services.

1) Vehicle mobility impact: Let's assume that the device density is defined based on speed [33], and more than 50% of vehicles require RSU assistance to execute the services. In other words, as the vehicle density increases, the offloading requests also increase and decrease, as observed in 10(c). The rationale behind this change is that when the speed is low and the density is high, it leads to exceeding network load. Subsequently, when the speed increases, the density reduces, resulting in low resource usage due to fewer offloading requests. Additionally, the vehicle speed range [40-100 Km/h] is amended to illustrate and map real-time scenarios. Compared to state-of-the-art (SOTA) models, our model exhibits better offloading efficiency and service completion ratio while increasing vehicle speed and increasing server capacity (Fig.10(a)), since the density and environment load depend on each other (as density increases, the environment load increases).

2) Computing capacity impact: Continuing from the above section, we are currently analyzing the offloading efficiency and service completion ratio based on server computing capacity. We observe that the offloading efficiency exceeds as the capacity increases because the time required to complete the service is reduced due to the increase in server computing power. However, the EMSRA completion ratio and offloading ratios are better than other models but still do not provide satisfactory performance compared to our model. Additionally, our model has a service completion ratio of 0.859%, which is still not satisfactory for video sharing over the network in smart transportation environments (see references 10(b)-10(d)).



(a) Service completion with speed (b) completion rate with 40Km/h (c) Offloading efficiency with speed (d) Offloading with 40Km/h

Fig. 10: Mobility impact analysis based on service completion ratio, offloading efficiency with speed 40 Km/h

VI. Conclusion

In this article, we examined the service computation and offloading strategy problem by the proposed ASXC^2 approach for fog environment. At first, the objective function is assessed to measure the node-centric functionality to enhance the network performance with cost-effectiveness, high utility, and service reliability. The node/device centric service cost prediction facilitates the efficient mapping of services to suitable servers during offloading. The service cost estimation is played a crucial role in enhancing the system performance by 0.13% service offloading efficiency, 0.82% rate of service completion when transmit data size is 400 kb, and 0.058% average service offloading efficiency with 40 CPU Megacycles when the vehicle moves 60 Km/h speed in around the server communication range. The non-linear programming multi-tenancy heuristic method is played in enhancing the server utility efficiency compared to state-of-the-art approaches by scheduling services and managing resources to prevent service execution issues. The deployment of a distributed Markov model is influenced the policy convergence ratio of service offloading strategy by assessing the required resources to execute the arrived service requests within the latency constraints. The status of the server is continuously estimated, and the correlation matrix coefficients are continues update to assess server feasibility. The deployment of Lyapunov method is effectively balanced server usage and service queue stability to optimize service offloading to meet the application deadline.

Acknowledgments

This work was supported in part of Basic Science Research Programs of the Ministry of Education (NRF-2018R1A2B6005105) and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2019R1A5A8080290).

References

[1] R. I. Meneguette, A. Boukerche, and A. H. Pimenta, “Avarac: An availability-based resource allocation scheme for vehicular cloud,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3688–3699, 2018.

[2] J. Arbib and T. Seba, “Rethinking transportation 2020–2030(rethinkx sector disruption),” p. [Online]. Available: https://books.google.com.tr/books?id=lLMtswEACAAJ, 2017.

[3] M. S. Mekala and Jolfaei, “Resource offload consolidation based on deep-reinforcement learning approach in cyber-physical systems,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–10, 2020.

[4] M. Mekala and G. Dhiman, “Deep learning-influenced joint vehicle-to-infrastructure and vehicle-to-vehicle communication approach for internet of vehicles,” *Expert Systems*, p. e12815, 2021.

[5] R. Morabito and Farris, “Evaluating performance of containerized iot services for clustered devices at the network edge,” *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 1019–1030, 2017.

[6] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, “Cost efficient resource management in fog computing supported medical cyber physical system,” *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 1, pp. 108–119, 2015.

[7] Z. Ning, “When deep reinforcement learning meets 5g-enabled vehicular networks: A distributed offloading framework for traffic big data,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1352–1361, 2019.

[8] Q. Qi, J. Wang, Z. Ma, H. Sun, Y. Cao, L. Zhang, and J. Liao, “Knowledge-driven service offloading decision for vehicular edge computing: A deep reinforcement learning approach,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4192–4203, 2019.

[9] M. Li, J. Gao, N. Zhang, L. Zhao, and X. Shen, “Collaborative computing in vehicular networks: a deep reinforcement learning approach,” in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[10] Z. Ning and Zhang, “Joint computing and caching in 5g-envisioned internet of vehicles: A deep reinforcement learning-based traffic control system,” *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[11] H. Li, K. Ota, and M. Dong, “Learning iot in edge: Deep learning for the internet of things with edge computing,” *IEEE network*, vol. 32, no. 1, pp. 96–101, 2018.

[12] S. E. Mahmoodi, R. Uma, and K. Subbalakshmi, “Optimal joint scheduling and cloud offloading for mobile applications,” *IEEE Transactions on Cloud Computing*, 2016.

[13] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, “Delay-optimal computation task scheduling for mobile-edge computing systems,” in *2016 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2016, pp. 1451–1455.

[14] Y. Mao, J. Zhang, and K. B. Letaief, “Dynamic computation offloading for mobile-edge computing with energy harvesting devices,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[15] R. Deng, “Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption,” *IEEE internet of things journal*, vol. 3, no. 6, pp. 1171–1181, 2016.

[16] S. Ulukus, “Energy harvesting wireless communications: A review of recent advances,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 3, pp. 360–381, 2015.

[17] W. Zhang and Wen, “Energy-optimal mobile cloud computing under stochastic wireless channel,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, 2013.

[18] K. Liang, “An integrated architecture for software defined and virtualized radio access networks with fog computing,” *IEEE Network*, vol. 31, no. 1, pp. 80–87, 2017.

[19] Y. Li, S. Xu, and D. Li, “Deep reinforcement learning for collaborative computation offloading on internet of vehicles,” *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–13, 2021.

[20] A. Kiani and N. Ansari, "Toward hierarchical mobile edge computing: An auction-based profit maximization approach," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2082–2091, 2017.

[21] X. Xu, Q. Huang, H. Zhu, S. Sharma, X. Zhang, L. Qi, and M. Z. A. Bhuiyan, "Secure service offloading for internet of vehicles in sdn-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720–3729, 2021.

[22] X. Chen, L. Jiao, and Li, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2015.

[23] R. Atallah, "Deep reinforcement learning-based scheduling for roadside communication networks," in *2017 15th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE, 2017, pp. 1–8.

[24] R. F. Atallah, C. M. Assi, and M. J. Khabbaz, "Scheduling the operation of a connected vehicular network using deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1669–1682, 2018.

[25] T. Zhang, K. Zhu, and J. Wang, "Energy-efficient mode selection and resource allocation for d2d-enabled heterogeneous networks: A deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1175–1187, 2020.

[26] G. Sun, V. Chang, G. Yang, and D. Liao, "The cost-efficient deployment of replica servers in virtual content distribution networks for data fusion," *Information Sciences*, vol. 432, pp. 495–515, 2018.

[27] B. Huang, X. Liu, S. Wang, L. Pan, and V. Chang, "Multi-agent reinforcement learning for cost-aware collaborative task execution in energy-harvesting d2d networks," *Computer Networks*, vol. 195, p. 108176, 2021.

[28] J. D. Little and S. C. Graves, "Little's law," *Building intuition: insights from basic operations management models and principles*, pp. 81–100, 2008.

[29] M. S. Mekala, P. Rizwan, and M. S. Khan, "Computational intelligent sensor-rank consolidation approach for industrial internet of things (iiot)," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2121–2130, 2023.

[30] M. S. Mekala, G. Dhiman, G. Srivastava, Z. Nain, H. Zhang, W. Viriyasitavat, and G. P. S. Varma, "A drl-based service offloading approach using dag for edge computational orchestration," *IEEE Transactions on Computational Social Systems*, pp. 1–9, 2022.

[31] P. Mekala, Viswanathan, "Energy-efficient virtual machine selection based on resource ranking and utilization factor approach in cloud computing for iot," *Computers & Electrical Engineering*, vol. 73, pp. 227–244, 2019.

[32] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading," *IEEE vehicular technology magazine*, vol. 12, no. 2, pp. 36–44, 2017.

[33] T. Generation *et al.*, "Technical specification group radio access network; study on channel model for frequencies from 0.5 to 100 ghz," *3GPP TR 38.901 v16. 1.0*, 2019.

M. S. Mekala (Senior Member, IEEE, AFHE) received the Ph.D. degree from VIT University. He is currently working as an Assistant Professor with the School of Computing, Robert Gordon University, Aberdeen, U.K. He is a Former Post-Doctoral Researcher at RLRC LAB, Yeungnam University, gyeonnsan, Korea, and a member of Vice-Chancellor Peshi, Research Coordinator and Member of FEB Laboratory at KL University. He has published more than 20 peer-review research papers (indexed in SCI-SCIE). His research interests include service computing, intelligent machine vision, data communication, decision making system design, edge computing, CPS, IoT communication, and Reliability Analysis. He is a recipient of Best Research award for two consecutive years in 2018 and 2019. He received the research scientist award in 2020. He is a Former-GE of IEEE TRANSACTION ON SERVICE COMPUTING, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTION ON ROBOTICS, computer networks, neural computing and applications, Applied soft-computing Methods.

Gaurav Dhiman (Senior Member, IEEE) received the master's degree in computer applications and the Ph.D. degree in computer engineering from the Thapar Institute of Engineering and Technology, Patiala, India. He completed his Post-Doctoral research work from Federal Institute of Science, Education and Technology of Ceará, Fortaleza-CE, Brasil. He is currently working as an Assistant Professor in School of Sciences and Emerging Technologies, Jagat Guru Nanak Dev Punjab State Open University, Patiala. He is also associated with Chandigarh University and Graphic Era Deemed to be University. He was selected as an outstanding reviewer from Knowledge-Based Systems (Elsevier). He has published more than 300 peer reviewed research papers (indexed in SCI-SCIE) and ten international books. He is a senior member of IEEE. His research interest includes single, multi, many-objective optimization (bio-inspired, evolutionary and quantum), soft computing (type-1 and type-2 fuzzy sets), artificial intelligence, machine-learning, deep-learning, software engineering, and healthcare analytics. His research articles can be found in various reputed IEEE Transactions, Elsevier, Springer, and Wiley journals.

Ju H. Park (Senior Member, IEEE) received the Ph.D. degree in electronics and electrical engineering from Pohang University of Science and Technology (POSTECH), Pohang, Republic of Korea, in 1997. From May 1997 to February 2000, he was a Research Associate in Engineering Research Center-Automation Research Center, POSTECH. He joined Yeungnam University, Kyongsan, Republic of Korea, in March 2000, where he is currently the Chuma Chair Professor. He has published a number of articles in these areas. His research interests include robust control and filtering, neural/complex networks, fuzzy systems, multiagent systems, and chaotic systems. Since 2015, he has been a recipient of the Highly Cited Researchers Award by Clarivate Analytics (formerly, Thomson Reuters) and listed in three fields, Engineering, Computer Sciences, and Mathematics, in 2019 to 2022. He is a Subject Editor, Advisory Editor, Associate Editor and Editorial Board Member of several international journals, including IET Control Theory and Applications, Applied Mathematics and Computation, Journal of The Franklin Institute, Nonlinear Dynamics, Engineering Reports, Cogent Engineering, the IEEE TRANSACTION ON FUZZY SYSTEMS, the IEEE TRANSACTION ON NEURAL NETWORKS AND LEARNING SYSTEMS, and the IEEE TRANSACTION ON CYBERNETICS. He is a fellow of the Korean Academy of Science and Technology (KAST).

Ho-Youl Jung received the Ph.D. degree in electronics engineering from the INSA de Lyon (Institute National des Sciences Appliquées de Lyon), France, in 1998. He is currently a Professor with the Department of Information and Communication Engineering, Yeungnam University, Korea. His Teaching and research interests include digital signal processing, computer vision, deep-learning, autonomous vehicle, computer graphics, control signal processing, and IoT.

Wattana Viriyasitavat (Senior Member, IEEE) is an Associate Professor of Information Technology at Chulalongkorn University, Bangkok, Thailand. He received the D.Phil. degree (Ph.D.) in computer science from the University of Oxford, Oxford, U.K., in 2013. He is an IEEE Senior member (2019). Presently, he is a full-time lecturer and researcher with the Business Information Technology Division, Department of Statistics, Faculty of Commerce and Accountancy, Chulalongkorn University. He is a co-Founder and Chief Technology officer of PrimeCapture, a startup company and has served as a consultant in a few leading organizations namely Tourism Authority of Thailand, Bank for Agriculture and Agricultural Cooperatives, and Metropolitan Electricity Authority. He is also certified by ISO 15540 as a professional assessor. Dr. Viriyasitavat's teaching and research activities emphasize on the area of trust and security, service selection and composition, service workflows, business process management, and recently Internet of Things, Cyber-Physical-System, and Blockchain technology. He has published several articles in both academic and practitioner journals indexed by SCI, including IEEE Transactions on Industrial Informatics, IEEE Communications Surveys & Tutorials, Journal of Intelligent Manufacturing, Enterprise Information System, and Journal of Industrial Information Integration. He has served as an editorial advisory member for Journal of Industrial Information Integration. His received annual research awards from National Research Council of Thailand (2007), and the 2016 Thailand Frontier Researcher Awards (Thomson Reuters).