Edge Computing QoE Maximization in EV Parking Scenario

Yu-Chieh Lee, Yao Chiang, Hung-Yu Wei
Department of Electrical Engineering, National Taiwan University, Taiwan

***Abstract*—Facing the emergence of 6G and the rapid increase in electric vehicles (EVs), smart parking lots providing real-time services like EV charging have become essential. Edge computing, due to its proximity to end devices, offers low latency and high bandwidth, but its limited resources necessitate efficient allocation. We present a parking lots scenario with edge computing system offering four key services: Supply Equipment Communication Controller, charging space detection, monitoring, and video streaming, along with their QoE models and corresponding estimation models. We predict system requests for the next time slot and employ the Maximum-chosen algorithm and Collaborative Optimal Decision Search method to optimize service deployment and assignment, maximizing QoE values and resource efficiency. Simulation results validate that we can obtain request status that is more similar to real requests by prediction and the Collaborative Optimal Decision Search method can generate optimal service assignment strategy within different methods.**

***Index Terms*—edge computing, resource allocation, QoE optimization, electric vehicle, smart parking lot**

I. Introduction

As the era of 6G [1] approaches, the proliferation of AI and IoT applications will lead to a surge in data generation, while sufficient bandwidth, data transmission speed, and quality of experiences (QoE) must be maintained. Concurrently, the high growth rate of EVs [2] necessitates efficient infrastructure, particularly for smart parking lots offering real-time services like EV charging.

To conquer the above issues brought by 6G and the need for parking lots for EVs, edge computing [3] can be a splendid solution due to its characteristics such as extreme user proximity, ultra-low latency, and high bandwidth, which can be taken advantage by real-time services for EV charging while maintaining end devices QoE. However, compared with cloud computing, resources equipped by edge servers are limited so efficient resource allocation and utilization have always posed challenges for edge computing.

So far, there is some research related to applications of edge computing for parking spaces. [4] proposed an infrastructure that utilized edge computing nodes to receive and process data of vehicles for accurate parking spot positioning and user status detection. [5] proposed federated learning-based parking space estimation with edge computing nodes and parked vehicles. Moreover, there is also research on electric vehicle charging and edge computing, for example, [6] proposed a model based on a stacked auto encoder neural network at edge platform to predict the load on EV charging stations. However, these studies often lack comprehensive integration of services within a smart EV parking lot.

Motivated by the above issues, we have done the following work in this paper:

1) Propose a scenario consisting of parking lots and an edge computing system architecture by following IEEE 1935 standard.
2) Implement four frequently used services and formulate their QoE models and estimation QoE models.
3) Predict requests from end devices in the next time slot to decide on service assignments and deploy them in advance using the proposed Maximum-chosen algorithm or Collaborative Optimal Decision Search algorithm for maximizing total QoE values and fast service provisions.

The remainder of this paper is organized as follows. We introduce our scenario and architecture in Section II. We define our problem formulation and sub-problems and propose our solutions in Section III. In Section IV, we show our simulation results. Moreover, we conclude the paper in Section V.

II. System Model

In this paper, we introduce a smart parking lot scenario utilizing the edge/fog framework proposed in [7]–[9], which is formed by a 3-level architecture, including computer-level, control-level, and orchestrator-level defined in IEEE 1935 standard. Furthermore, we also specify and implement four constantly used services in the scenario and define their QoE models for further QoE maximization strategies.

The proposed scenario shown in Fig. 1 contains a whole edge system architecture with three edge areas $E = \{X, Y, Z\}$ formed by three 3-story parking lots and managed by an edge/fog orchestrator (EFO). Each parking lot is composed of one control node, three compute nodes (one on each floor), and several end devices. The four major components in the system architecture will be introduced as follows:

a) End devices: Mainly include electric vehicle owners' mobile devices and all electric vehicle charging stations in the system, which are the components that request services. There are several and a fixed number of charging stations on each floor of a parking lot. The set of end devices in the whole system is denoted as $U = \{1, 2, ..., U\}$.

b) Compute node: The computer-level entity that handles practical computing tasks. In our scenario, it is responsible for executing and providing services, which will be mentioned in detail in the following paragraph, to end devices. The set

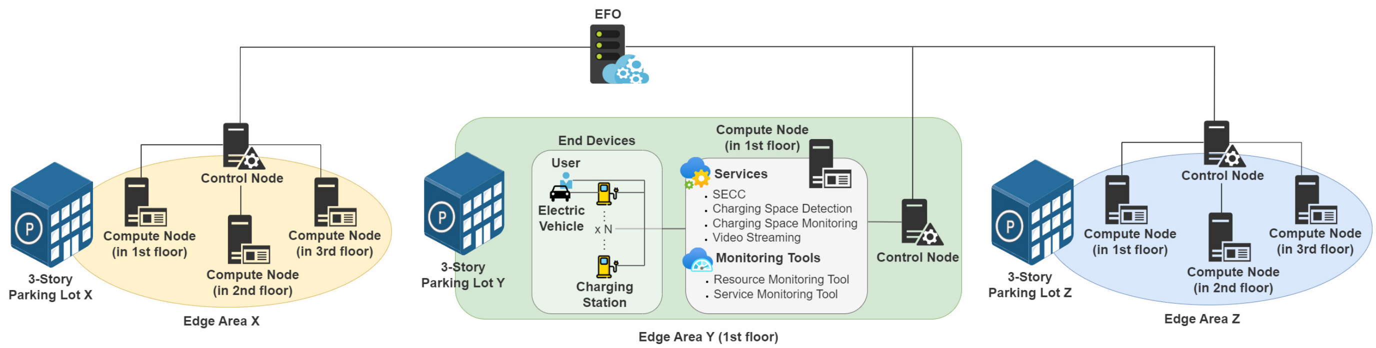

Fig. 1. The proposed system scenario and architecture.

of compute nodes in the whole system is denoted as $C = \{1, 2, ..., C\}$.

c) Control node: The control-level entity to manage and coordinate compute-level entities, which indicate compute nodes. It collects request and resource information from compute nodes in the same edge area and sends it to the EFO in our scenario. Furthermore, it also sends control messages from EFO to below compute nodes.

d) EFO: The orchestrator-level entity for the main management and orchestration of the whole edge system, which is composed of several management and orchestration components. In our scenario, the component collects a global view of the whole edge system from all control nodes and runs the proposed algorithm mentioned in Section III to predict future requests, make service deployment and assignment decisions, and send control messages to all edge areas.

In addition, we implement services and propose their QoE models for further optimization decisions as follows:

a) Supply equipment communication controller (SECC) service: Service defined in standard ISO 15118 and required by a charging station to keep communications with the electric vehicle communication controller (EVCC) on an EV before and within the charging process. The QoE model of service $SECC$ request by end device u and runs on compute node c is defined as below:

$$Q_{c,u}^{SECC} = \begin{cases} \frac{1}{\frac{1}{n}\sum_{i=1}^{n}(RTT_{c,u}+t_i^{SECC})} & \text{if no timeout} \\ -1 & \text{if timeout} \end{cases} \tag{1}$$

, in which $RTT_{c,u}$ represents the round-trip time between compute node c and end device u, n represents the total request number in the charging process, and t_i^{SECC} represents the processing time of the ith request. If the value of $RTT_{c,u}+t_i^{SECC}$ for any i exceeds the $V2G\_EVCC\_Msg\_Timeout$ value defined in ISO 15118 standard, we consider it a timeout event.

b) Charging space detection (CSD) service: Service implemented with MobilenetV2 model to detect if there is a car in the charging space. Its QoE model is defined as below:

$$Q_{c,u}^{CSD} = \beta \times \frac{1}{\frac{1}{n}\sum_{i=1}^{n}(RTT_{c,u} + t_i^{CSD})} + (1-\beta) \times 0.72 \tag{2}$$

, in which β and $1-\beta$ are the weight factors that represent the importance of the two metrics, and 0.72 is the accuracy value referenced from [10].

c) Charging space monitoring (CSM) service: Service implemented with OpenCV Haar cascade classifiers for face recognition to ensure electric vehicles' security during charging processes. Its QoE model is defined as below:

$$Q_{c,u}^{CSM} = \gamma \times \frac{1}{\frac{1}{n}\sum_{i=1}^{n}(RTT_{c,u} + t_i^{CSM})} + (1-\gamma) \times 0.9624 \tag{3}$$

, in which γ and $1-\gamma$ are the weight factors, and the accuracy value 0.9624 is referenced from [11].

d) Video streaming (VS) service: Service which can stream video to end devices while the car owner is waiting for the charging process to finish. Its QoE model is referenced and modified from [12] as below:

$$\begin{aligned} Q_{c,u}^{VS} &= \alpha X_u - (1-\alpha) I^a \\ &= \alpha X_u - (1-\alpha)(RTT_{c,u} + t^{VS}) \end{aligned} \tag{4}$$

, in which α and $1-\alpha$ are the weight factors, X_u represents the resolution of the video for end device u, and I^a represents the initial waiting time, which is equal to the round-trip time $RTT_{c,u}$ plus the initial processing time t^{VS}.

III. Proposed Method and Algorithm

In this section, we define value $v_{c,u}^s$ given by an end device u served by service s on compute node c at time slot t as below:

$$v_{c,u}^s(t) = prior^s \times R_u^s(t) \times Q_{c,u}^s \tag{5}$$

, where $prior^s$ indicates the priority and the importance factor of service type s in the four kinds of services $S = \{SECC, CSD, CSM, VS\}$, $R_u^s(t)$ is a boolean value indicating whether the end device u in time slot t has requested for service type s and $Q_{c,u}^s$ represents the corresponding QoE value given by end device u. Our main objective is to maximize the summation of v values in total q time slots. We split the problem into two sub-problems due to different time dimensions, where the first sub-problem predicts the requests from all end devices in the system in the next time slot based on historical data, and the second sub-problem does the service deployment and assignment decisions according to the request probabilities predicted by the first sub-problem.

A. Main Problem Formulation

Our objective is to achieve the most efficient utilization of system resources, which is to maximize the total value v in q system time slots under the constraints of CPU, memory, network resources of compute nodes, and uplink, and downlink capacities of end devices. Since $v^s_{c,u}$ is the comprehensive value that takes service priority, request probability, and QoE into consideration, the bigger the value $v^s_{c,u}$, the more significance it has for the system. Therefore, our objective function can be written as below:

$$\max_{D} \sum_{t=0}^{q} \sum_{u \in U} \sum_{c \in C} \sum_{s \in S} v^s_{c,u}(t) \times d^s_{c,u}(t) \tag{6}$$

s.t.

$$C1: \sum_{s \in S} \sum_{u \in U} \tau^s_{c,u}(t) < \tau_c, \forall c \in C$$
$$C2: \sum_{s \in S} \sum_{u \in U} \omega^s_{c,u}(t) < \omega_c, \forall c \in C$$
$$C3: \sum_{s \in S} \sum_{u \in U} dl^s_{c,u}(t) < ul_c, \forall c \in C$$
$$C4: \sum_{s \in S} \sum_{u \in U} ul^s_{c,u}(t) < dl_c, \forall c \in C$$
$$C5: \sum_{s \in S} \sum_{c \in C} ul^s_{c,u}(t) < ul_u, \forall u \in U$$
$$C6: \sum_{s \in S} \sum_{c \in C} dl^s_{c,u}(t) < dl_u, \forall u \in U$$
$$C7: \sum_{c \in C} d^s_{c,u}(t) <= 1$$

, where $d^s_{c,u}(t)$ is a boolean value which indicates whether we deploy service s on compute node c and assign it to end device u at time slot t, and D is the set consists of all $d^s_{c,u}$ values.

$C1$ and $C2$ limit the total usage of CPU and memory of all services on a compute node to be less than the total number of CPU and memory on the compute node, which is τ_c and ω_c. Moreover, $C3$ and $C4$ constrain the total data speed transmitted to and received from end devices to be less than the compute node's uplink capacity and downlink capacity. $C5$ and $C6$ constrain the total data speed transmitted to and received from services executing on compute nodes to be less than the end device's uplink capacity and downlink capacity. Besides, $C7$ ensures that end device u's request for service s will only be accepted by at most one compute node.

B. Sub-problem 1. Request Prediction

Assume that the request status in each parking lot has a certain level of periodicity. We can calculate request probability in time slot t through the historical request data and below equation:

$$P^s_u(t) = \frac{1}{w} \sum_{x=t-w}^{t-1} R^s_u(x) \tag{7}$$

, where $P^s_u(t)$ is the probability that end device u requests for service s, and w is the hyper-parameter indicating the time window size, which can be suitably modified based on the periodicity of the request status of an edge area.

C. Sub-problem 2. Service Assignment Decision

Our goal for the second sub-problem is to assign services to appropriate compute nodes to serve specific end devices in advance to maximize the sum of estimated v values at time slot t, which is defined as:

$$\overline{v^s_{c,u}(t)} = prior^s \times P^s_u(t) \times \overline{Q^s_{c,u}} \tag{8}$$

Compare with Eq. 5, actual request value R^s_u is replaced by request probability value P^s_u, and actual QoE value $Q^s_{c,u}$ is replaced by estimated QoE value $\overline{Q^s_{c,u}}$ so that our sub-objective function can be represented as follow and with the same resource constraints in formula 6:

$$\max_{D} \sum_{u \in U} \sum_{c \in C} \sum_{a \in S} \overline{v^s_{c,u}(t)} \times d^s_{c,u}(t) \tag{9}$$

To estimate the QoE value in the next time slot, we need to define estimated QoE models for the four services referencing Eq. 1∼4 since we are unable to know the actual service condition in the next time slot. Below are estimated QoE models of:

- SECC service:

$$\overline{Q^{SECC}_{c,u}} = (1 - prob_{timeout}) \times \frac{1}{\overline{RTT_{c,u}} + \overline{t^{SECC}}} + prob_{timeout} \times (-1) \tag{10}$$

  , where $\overline{prob_{timeout}}$ is the average timeout probability, $\overline{RTT_{c,u}}$ is the average round-trip time between end device u and compute node c, and $\overline{t^{SECC}}$ is the average process time of the SECC service.

- CSD service

$$\overline{Q^{CSD}_{c,u}} = \beta \times \frac{1}{\overline{RTT_{c,u}} + \overline{t^{CSD}}} + (1 - \beta) \times 0.72 \tag{11}$$

  , where $\overline{t^{CSD}}$ is the average process time of the charging space detection service.

- CSM service

$$\overline{Q^{CSM}_{c,u}} = \gamma \times \frac{1}{\overline{RTT_{c,u}} + \overline{t^{CSM}}} + (1 - \gamma) \times 0.9624 \tag{12}$$

  , where $\overline{t^{CSM}}$ is the average process time of the charging space monitoring service.

- VS service

$$\overline{Q^{VS}_{c,u}} = \alpha X'_u - (1 - \alpha)(\overline{RTT_{c,u}} + \overline{t^{VS}}) \tag{13}$$

  , where X'_u is the previous resolution value used by the video streaming service for the end device in the same charging space as end device u, and $\overline{t^{VS}}$ is the average process time of the video streaming service.

To achieve our sub-objective, we propose Maximum-chosen algorithm and Collaborative Optimal Decision Search (CODS) method.

a) Maximum-chosen Algorithm: Related pseudo code is shown in Alg. 1, in which we first calculate all possible v values, then every time pick the maximum value and check if resources are enough. If enough resources are available, we assign end device u to service s running on compute node c and accumulate the total value.

Algorithm 1 Maximum-chosen Algorithm

Input: $P_u^s,\ \forall s \in S, \forall u \in U$
Output: Total value $value$ and service deployment and assignment decisions D
Let $value \leftarrow 0$
Initialize $\overline{V} = \{\overline{v_{c,u}^s} = prior^s \times P_u^s \times \overline{Q_{c,u}^s} \mid \forall s \in S, \forall c \in C, \forall u \in U\}$
Initialize $D = \{d_{c,u}^s = 0 \mid \forall s \in S, \forall c \in C, \forall u \in U\}$
Initialize $RC = \{rc_c = \{\tau_c, \omega_c, dl_c, ul_c\} \mid \forall c \in C\}$
Initialize $RE = \{re_u = \{dl_u, ul_u\} \mid \forall u \in U\}$
while There exist non-zero value in set $\overline{V}$ **do**
  Find maximum value $\overline{v_{c',u'}^{s'}}$ in set $\overline{V}$
  if `CheckResourceEnough`(s', c', u') **then**
    $value \leftarrow value + \overline{v_{c',u'}^{s'}}$
    $d_{c',u'}^{s'} \leftarrow 1$
    for $c \in C$ **do**
      $\overline{v_{c,u'}^{s'}} \leftarrow 0$
    end for
  else
    $\overline{v_{c',u'}^{s'}} \leftarrow 0$
  end if
end while
function CheckResourceEnough(s, c, u)
  if all value in $(rc_c - \{\overline{\tau^s}, \overline{\omega^s}, \overline{ul^s}, \overline{dl^s}\}) > 0$ **and** all value in $(re_u - \{\overline{dl^s}, \overline{ul^s}\}) > 0$ **then**
    $rc_c \leftarrow rc_c - \{\overline{\tau^s}, \overline{\omega^s}, \overline{ul^s}, \overline{dl^s}\}$
    $re_u \leftarrow re_u - \{\overline{dl^s}, \overline{ul^s}\}$
    return True
  else
    return False
  end if
end function

b) Collaborative Optimal Decision Search (CODS) Method: Besides Maximum-chosen algorithm, we also proposed CODS method based on the integer linear programming (ILP) method. Same as Maximum-chosen algorithm, we first initialize set $\overline{V} = \{\overline{v_{c,u}^s} = prior^s \times P_u^s \times \overline{Q_{c,u}^s} \mid \forall s \in S, \forall c \in C, \forall u \in U\}$, and define a set of binary decision variables $D = \{d_{c,u}^s \mid \forall s \in S, \forall c \in C, \forall u \in U\}$, in which variable $d_{c,u}^s$=1 represents that the request from end device u for service s is assigned to compute node c and $d_{c,u}^s$=0 indicates the opposite. Subsequently, we can formulate our ILP problem with formula 9 as the objective function and $C1 \sim C7$ defined in formula 6 as constraints.

Given the potentially large number of variables in decision set D, which scales with the number of charging spaces and end devices, solving the ILP problem might incur significant computational time. To mitigate this, we impose a 5-minute time limit during simulations. Consequently, the obtained solution may be near-optimal rather than optimal.

IV. Simulation Results

In this section, we show our environmental parameter settings during simulations. Subsequently, we present simulation results using request datasets of the three edge areas within 120 time slots, including request status from a total 90 charging spaces evenly distributed on all floors of all parking lots. We compare the similarities between real requests and requests with and without predictions, and also compare the simulation results using Maximum-chosen algorithm, CODS Method, and DEUA-H algorithm [13].

A. Environmental Parameter settings

In Table. I, we present the environmental parameters and their corresponding values utilized during the simulations. The top four parameters denote service type priorities, with SECC service for charging assigned the highest priority. Charging station detection and monitoring services share the same priority level, followed by video streaming, which is deemed the least critical. Additionally, parameters associated with service resource usage represent average values for one end device, which are organized from the monitoring data, while we also record average processing time for all types of services.

TABLE I
Parameter Settings.

Parameter	Value	Parameter	Value
$prior^{SECC}$	4/9	$prior^{CSD}$	2/9
$prior^{CSM}$	2/9	$prior^{VS}$	1/9
$prob_{timeout}$	0.01	$\overline{t^{SECC}}$	9 (ms)
$\overline{\tau^{SECC}}$	0.79 (core/user)	$\overline{\omega^{SECC}}$	366 (MB/user)
$\overline{dl^{SECC}}$	86 (kbps/user)	$\overline{ul^{SECC}}$	21 (kbps/user)
β	0.9	$\overline{t^{CSD}}$	5244.8 (ms)
$\overline{\tau^{CSD}}$	0.23 (core/user)	$\overline{\omega^{CSD}}$	430 (MB/user)
$\overline{dl^{CSD}}$	2.1 (kbps/user)	$\overline{ul^{CSD}}$	327 (kbps/user)
γ	0.7	$\overline{t^{CSM}}$	768.2 (ms)
$\overline{\tau^{CSM}}$	0.17 (core/user)	$\overline{\omega^{CSM}}$	49.3 (MB/user)
$\overline{dl^{CSM}}$	2.9 (kbps/user)	$\overline{ul^{CSM}}$	1.09 (Mbps/user)
α	5/7	$\overline{t^{VS}}$	0 (ms)
$\overline{\tau^{VS}}$	0.0033 (core/user)	$\overline{\omega^{VS}}$	14.1 (MB/user)
$\overline{ul^{VS}}$	59 (kbps/user)	τ_c	4 (cores)
ω_c	32 (GB)	ul_c	1 (Gbps)
dl_c	1 (Gbps)	ul_u	100 (Mbps)
dl_u	100 (Mbps)	w	6

B. Effects of doing request prediction

To generate request status as the input of Maximum-chosen algorithm and CODS method, we can do request prediction using the equation mentioned in Section III.B, or we can choose to use the request from the previous time slot. We compare the request status of these two methods and calculate their Mean Square Error (MSE) values with the real requests, which are presented in Fig. 2. The MSE values between real requests and request probabilities with prediction are constantly smaller than the other, indicating that it is more suitable to be the input for further service assignment decisions.

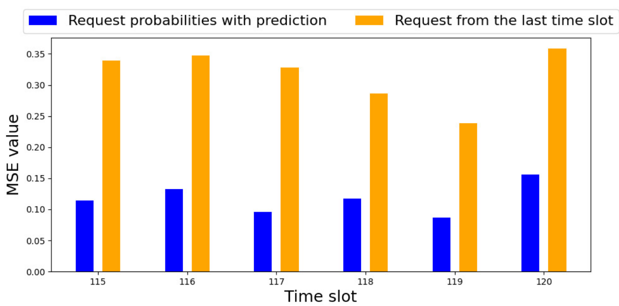

Fig. 2. MSE between real requests and request probabilities, requests from last time slot.

C. Comparison of results using different algorithm

In this section, we compare our approaches with DEUA-H [13]. We adapt DEUA-H by selecting service-end device pairs iteratively, following the order of $SECC, CSD, CSM, VS$ for services and selecting end device sequentially from edge area X, Y, and Z. Then, we assign each pair to the compute node with the highest $v^s_{c,u}$ value.

Using Maximum-chosen, CODS, and DEUA-H algorithms with request probabilities and real requests as inputs, as shown in Fig. 3 and 4, we find that CODS consistently performs better than others and achieves better resource efficiency by optimizing the total value v. Furthermore, DEUA-H which lacks priority consideration in assigning service-end device pairs to compute nodes, mostly performs worse compared to the Maximum-chosen algorithm.

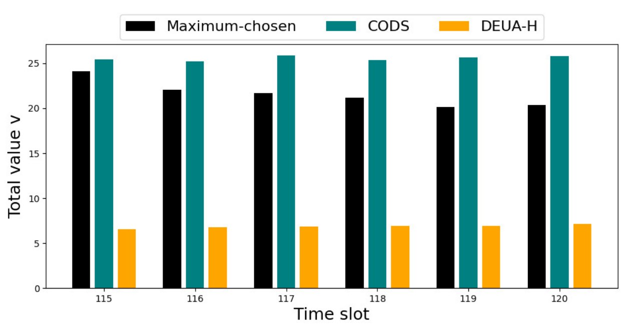

Fig. 3. Total value v by using Maximum-chosen/CODS/DEUA-H with request probabilities as input.

V. Conclusion

In this paper, we present a setup involving three smart parking facilities equipped with an edge system. We identify and evaluate four commonly utilized services within parking lots, establishing their QoE metrics and devising corresponding estimation models. By analyzing historical data, we determine request probabilities for upcoming time slots across all charging spaces. Subsequently, employing the Maximum-chosen algorithm and CODS method, we can optimize service deployment and assignment, thereby minimizing service wait times and maximizing resource efficiency. Our simulations demonstrate that with prediction, we can generate request status similar to the real requests and CODS is the best solution among different methods for service deployment and assignments.

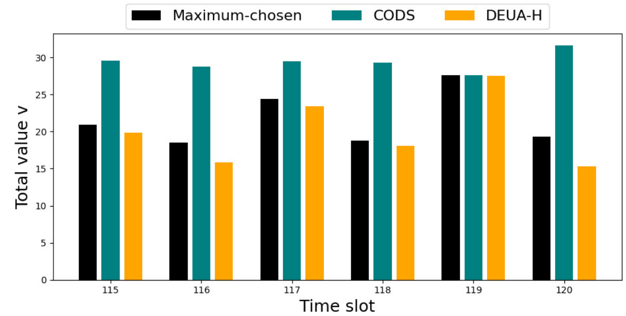

Fig. 4. Total value v by using Maximum-chosen/CODS/DEUA-H with real request as input.

VI. Acknowledgement

Hung-Yu Wei is grateful for the funding support by National Science and Technology Council (NSTC) of Taiwan under Grant 112-2622-8-002-021- and 112-2628-E-002-025-.

References

[1] J. R. Bhat and S. A. Alqahtani, "6g ecosystem: Current status and future perspective," *IEEE Access*, vol. 9, pp. 43 134–43 167, 2021.

[2] S. S. Ravi and M. Aziz, "Utilization of electric vehicles for vehicle-to-grid services: Progress and perspectives," *Energies*, vol. 15, no. 2, p. 589, 2022.

[3] Y. Chiang, Y. Zhang, H. Luo, T.-Y. Chen, G.-H. Chen, H.-T. Chen, Y.-J. Wang, H.-Y. Wei, and C.-T. Chou, "Management and orchestration of edge computing for iot: A comprehensive survey," *IEEE Internet of Things Journal*, 2023.

[4] C. Lee, S. Park, T. Yang, and S.-H. Lee, "Smart parking with fine-grained localization and user status sensing based on edge computing," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019, pp. 1–5.

[5] X. Huang, P. Li, R. Yu, Y. Wu, K. Xie, and S. Xie, "Fedparking: A federated learning based parking space estimation with parked vehicle assisted edge computing," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9355–9368, 2021.

[6] A. Luo, J. Yuan, F. Liang, Q. Yang, and D. Mu, "Load forecasting of electric vehicle charging station based on edge computing," in *2020 IEEE 3rd International Conference on Computer and Communication Engineering Technology (CCET)*, 2020, pp. 34–38.

[7] ETSI, "Mobile edge computing (mec); framework and reference architecture," *ETSI, DGS MEC*, vol. 3, pp. 1–18, 2016.

[8] *IEEE standard for edge/fog manageability and orchestration*, IEEE standard 1935-2023, 2023.

[9] T.-Y. Chen, Y. Chiang, J.-H. Wu, H.-T. Chen, C.-C. Chen, and H.-Y. Wei, "Ieee p1935 edge/fog manageability and orchestration: Standard and usage example," in *2023 IEEE International Conference on Edge Computing and Communications (EDGE)*, 2023, pp. 96–103.

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[11] A. B. Shetty, J. Rebeiro *et al.*, "Facial recognition using haar cascade and lbp classifiers," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 330–335, 2021.

[12] Y. Chiang, C.-H. Hsu, and H.-Y. Wei, "Collaborative social-aware and qoe-driven video caching and adaptation in edge network," *IEEE Transactions on Multimedia*, vol. 23, pp. 4311–4325, 2021.

[13] Z. Xu, G. Zou, X. Xia, Y. Liu, Y. Gan, B. Zhang, and Q. He, "Distance-aware edge user allocation with qoe optimization," in *2020 IEEE International Conference on Web Services (ICWS)*. IEEE, 2020, pp. 66–74.