IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 20 September 2023, accepted 4 October 2023, date of publication 11 October 2023, date of current version 18 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3323861

RESEARCH ARTICLE

Smart Parking Services: An Overall Solution Based on Internet of Things and Fog Computing

THANH-NAM PHAM[1], **DUC-THO MAI**[2], **AND XUAN-KIEN PHAM**[1]
[1]Thai Nguyen University of Information and Communication Technology, Thainguyen 24000, Vietnam
[2]Academy of Cryptography Techniques, VGISC, Hanoi 12511, Vietnam

Corresponding author: Thanh-Nam Pham (ptnam@ictu.edu.vn)

This work was supported by the Ministry of Education and Training, Vietnam (Grant number: B2020-TNA-18).

ABSTRACT This paper proposes constructing a smart parking service model based on the Internet of Things and Fog computing architecture. The weaknesses of previous parking service delivery systems are that they did not offer a complete set of end-to-end parking services, such as inaccurate information about parking in real-time, poor system storage and expansion capacity, and knowledge to assist the driver in successfully reaching the parking space at a minimum cost. Therefore, our system offers a complete set of parking assistance services, from when the driver searches and makes a reservation to navigation services that help the driver successfully park at the desired parking spot. To address the problems of storage and system expansion, we also implemented services based on the Internet of Things and Fog computing platforms. The simulation and experimental implementation results show that the services we designed significantly improve the performance of the existing parking system compared to the other systems. Real-time vehicles are handled quickly, significantly reducing driver costs and increasing profits for managers.

INDEX TERMS Fog computing, Internet of Things, indoor navigation, parking cost, parking services.

I. INTRODUCTION

In recent years, the Internet of Things (IoT) has gradually promoted integration, interaction, and communication processes with digital electronic devices, sensors, and actuators, supplying the essential services to achieve specific goals more efficiently [1], [2]. Due to the exponential expansion of IoT and cloud-based smart systems, the concept of developing smart cities has reached a new level of vision. Smart cities help reduce operational costs, improve city management, and enhance effectiveness as well as productivity [3]. The smart city concept comprises systematic monitoring and control of infrastructure, skyscrapers, intelligent transportation systems [4], healthcare [5], education, energy consumption, and public security [6].

In traffic management strategies of smart cities, Smart Parking (SP) technologies provide an efficient solution to decrease parking arrival time and traffic congestion. Researchers in both academic and industrial fields have paid attention to SP systems due to their economic, environmental, and aesthetic benefits. IoT, with the advantage of features for smart object communication [7], is a crucial infrastructure for SP architectures, which use ultrasonic, magnetometer, visual, ZigBee, Wi-Fi, and next-generation networking technologies. Collecting real-time sensor data and merging it with machine learning and fog computing enables real-time monitoring and managing [8]. Commercial typical parking systems, including SmartParking [9], PlacePod [10], and Sitraffic Scala [11], have been offered and conceived. These systems have remote parking spot reservations, fee payment, interactive parking maps, and more; however, they are costly and complicated for public developers to produce due to their non-open source nature, so current parking systems have not entirely passed smart city evolution requirements yet. Many academic studies on SP systems have also been offered, as in [12], [13], and [14], but most only solve parking problems for local

The associate editor coordinating the review of this manuscript and approving it for publication was Taehong Kim.

spaces without considering the overall situation. Parking is a complex problem; hence, a proposal that allocates traffic as well as optimizes the parking schedule of all car parks needs to be figured out and evaluated. This plan must be tailored to the driver's essential services and solve the distribution of parking requirements. Some systems offered valet parking [15], but these suggestions only addressed individual options rather than a complete set of services for drivers, such as reserve, guidance, and payment services.

Fog computing has already become a prominent and attractive interest to develop its foundations on emergent infrastructure for storage, control, and security, providing services much closer to its end-users [16]. Inspired by cloud computing, this method offers a range of beneficial features, such as resource pooling, real-time processing, and less expensive scaling ability [17]. The fog computing model supplies all the resources at the edge of the network. In the case of parking premises, various available fog nodes will be located to collect data, offer information related to the availability of parking areas for vehicles, and help users make parking decisions. In addition, an online system has also been designed for reserving parking spaces in advance so that users can easily immediately book before approaching any parking lots after registering to access them. Users can also cancel the reserved parking area; in that case, another vehicle from the wait list will be allocated under the system's priority. Parking costs will be lower than online reservation parking since waiting times are shorter. The information concerning online reservation parking gets updated using fog nodes, and the status of available parking spaces is checked by tracking the information about online and on-site parking reservations.

From these outstanding features of fog computing, in this paper, we propose to build an SP system based on IoTs and Fog computing technologies. Our suggested system will provide a closed end-to-end service cycle from searching and booking the desired parking space to successfully parking. This study has the advantages of supplying the services required for smart parking lots, enhancing navigation services inside parking lots, and evaluating the deployment of a parking system using fog computing technology, which is suitable for real-time parking applications with minimal service delays and makes it simple to scale the system to satisfy a large number of parking demands.

A. RELATED WORKS

A lot of cutting-edge studies on how to design an SP system currently exist. In order to deploy SP services to many users with real-time parking requests, recently recommended procedures apply based on the IoTs and cloud computing technologies.

The authors in [18] proposed to build an SP system based on Internet of Things technology and cloud computing. Their car park offers a parking reservation service running on the user's smartphone. This booking service is built quite simply and does not provide options for users to find the desired parking space. Rajvanhi et al. [19] deployed an SP system based on sensors and Cloud computing technology. They used ultrasonic and IR sensors in combination with LCD screens to help users quickly find available parking spaces. This system was only suitable for small-sized parking lots, and system performance would be significantly degraded with large-scale parking lots. The reason was that using only sensor equipment leads when the number of vehicles is high, the implementation of collecting information about parking positions will be complicated, and the system cost is not optimal. Saharan et al. [20] presented a strategy for dynamic pricing and distribution of parking spots in on-street parking scenarios based on machine learning and game theory. Two categories of parking users (PUs), namely paid parking users (PPUs) and restricted parking users (RPUs), are considered. RPUs have access to FoC parking places once each day. PPUs compete to reduce costs, whereas RPUs strive to extend the FoC-granted time. Parking controllers (PCs) seek to increase PPU income while lowering the overall FoC parking period provided to RPUs. The random forest model is utilized to forecast occupancy and calculate parking fees. The Seattle city parking and price datasets are used to anticipate occupancy and collect payments. The recommended DyPARK Pricing and Allocation Scheme (PAS) is compared with its four versions to assess the communication system's performance, and simulated results prove their scheme's superiority over existing state-of-the-art schemes.

Previous SP systems have a weakness in that when the number of parking requests to the system is large simultaneously, the system can not handle all these requests, leading to congestion and slow service response times. In recent years, many SP systems have applied Fog Computing [21] and Edge computing [22] technologies to speed up the real-time calculation and processing of parking requests in the parking lot. Since fog computing was coined by Cisco in 2012, a few efforts were proposed to adopt it on the SP system implementations. In several recent studies [23], [24], [25], [26], the authors suggested building an SP system based on the Internet of Things and Fog Computing technologies. The fog computing technology reduces service latency and performs well when parking requests are significant. However, these experiments only stop at a shallow level, such as offering a basic deployment model of parking services based on fog computing technology [23] and proposing a secure and Reliable Smart Parking Scheme (RSPS) for parking services [24]. Tang et al. [27] introduced a fog computing-based SP architecture and recommended various algorithms to optimize parking request allocation so as to decrease parking charges, fuel consumption as well as gas emissions. Fog gateways are located throughout different parking lots and transmit parking suggestions to the means in the list. These recommendations are created from the combinations of multiple thresholds, such as the cost of waiting time, walking, or driving to seek an available slot.

Another fog computing-based SP model is illustrated in [28]. In this model, a simulation deployment could depict how fog computing reduces lag and network utilization when compared to traditional cloud-based metrics. Lee et al. [29] presented in detail a system that adopts a relatively powerful computer to replace fog gateways. The suggested computer supplies advanced edge computing services (machine learning algorithm-based processing), demonstrating a structure as same as the ones used with cloudlets [30]. In the noted work, the authors concentrated on enhancing the accuracy of vehicle position in the parking area and announced 99.1% in the selected experimental scenario by handling the Received Signal Strength Indicator (RSSI) from Bluetooth Low Energy (BLE) beacons collected from a parking lot. In the works of [31] and [32], the authors described sequentially a theoretical design for an IoT-enabled fog computing-based SP and a low-cost SP system based on Arduino nodes, respectively, and these researchers were planned to perform for the Nigerian market in the schedule.

This article proposes constructing a prevailing set of reliable parking assistance services for the SP system. The model is deployed based on Fog computing technology to meet users' service requirements in real-time and allows users to pay the minimum cost for parking services.

B. CONTRIBUTIONS

The contributions of this work are summarized as follows:

1) We designed a whole set of smart services to build a parking system based on Fog computing technology;
2) Considering the parking traffic load balancing factor, a method is offered to find the appropriate cost to be paid by the user in the parking system;
3) The recommendation system has been successfully simulated and built and given several evaluations.

C. ORGANIZATION

The remainder of this paper has the following organization: Section II describes the proposed system architecture. Section III presents the offered smart services. Section IV is the implementation and evaluation of our system. We have a conclusion in Section V.

II. PROPOSED SYSTEM ARCHITECTURE

A. PARKING SERVICES BASED ON FOG COMPUTING TECHNOLOGY

Our proposed SP system architecture based on IoT and fog computing technology is depicted in Fig.1.

These advanced services are deployed based on the Internet of Things and cloud data storage technologies. When applied in practice, we propose implementing the Fog computing model to speed up computation and process real-time data. Detailed descriptions of the implementation of these services are described in the following sections. The proposed system is separated into the Service, Core, and Management layers.

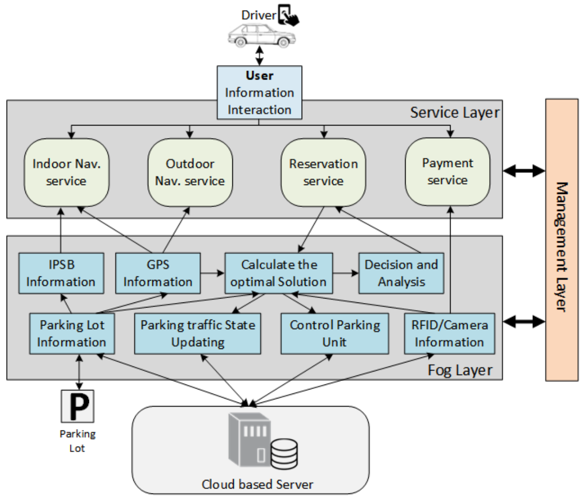

FIGURE 1. Proposed architecture for SP services.

- Service layer: This directly interacts with the user as a class. User access to the functionality of our system is through this class. Here, we provide reservation, outside guidance, indoor guidance, and payment services. These essential services offer complete support, enabling the driver to reach the desired parking spot quickly. Users who want to use the system's services must first log in via a software application running on a smartphone. Data is exchanged with users through 3G/4G/Wi-Fi connections.
- Fog layer: This layer contains the system's core components. It executes the commands and service requirements of the user. The main features include the Global Positioning System (GPS) information center, the Indoor Positioning System based on Bluetooth Low Energy (IPSB) center, the Calculation center, and the Decision and Analysis center. These centers get the database from the Parking Information block, Parking Traffic State updating block, Control Parking Arduino Unit, and the RFID Reader Card/Surveillance Camera Information. Core components retrieve data from the database, then calculate, analyze, and return the results to the requested services. Of these components, the parking traffic state updating center provides real-time traffic status information for vehicles arriving at the parking lots. The GPS information center provides information on the user's current location based on the GPS address. The Calculation center calculates the cost to the user based on data from the GPS information center and the location of the nearest car parking place for the user. The Decision and Analysis Center analyzes and selects the optimal result based on the user's request. This result will be used in the parking reservation service. The IPSB center will provide data to help deploy indoor guidance service for basement parking where GPS signals are no longer active.

- Management Layer: This layer will manage the entire operation of the parking network. Monitor data and detect irregularities in parking lots.

III. PARKING SERVICE IMPLEMENTATIONS

This section proposes a complete set of parking services based on IoT and Fog computing technologies for our SP system.

A. RESERVATION SERVICE

An algorithm is adopted to optimize the user's cost based on the desired destination to implement the reservation system. This algorithm helps to find a cost function based on factors related to the user's parking decision, such as the distance to the parking lot, the number of available parking spaces, and the parking cost. Users can choose the available parking slot based on the recommendations of the reservation service, including:

- shortest distance
- parking fee per hour
- availability of the parking lot (or the number of free spaces)
- optimal solution (based on the cost function)

To find the optimal solution for the user's cost of parking, we use a cost function as shown in Eq. (1). This function will calculate the cost of the user to get the destination car park CP_j:

$$F_j = F\left(m_{ij}\right) = \sum_{i=1}^{N} a_i \times Cost\left(m_i\right) \tag{1}$$

where a_i is the weight of the influence of factors m_i on the user's decision to choose a parking location, these factors can be the travel distance between the user and the j^{th} parking lot, the number of available parking spaces in the j^{th} parking lot, the parking fee in the j^{th} parking lot, or some other parameters. If we are only interested mainly in the influence of the first three parameters on the user's decision, then we have the following cost function in Eq. (2) to park the car at the j^{th} parking lot:

$$\begin{aligned} F_{cost(j)} &= F_{cost(j)}\left(\alpha, \beta, \gamma\right) \\ &= Cost_j\left(\alpha\right) + Cost_j\left(\beta\right) + Cost_j\left(\gamma\right) \end{aligned} \tag{2}$$

where $Costj(\alpha)$, $Costj(\beta)$ and $cosj(\gamma)$ are the cost function of travel distance (α), availability of car park (β) and the parking fee (γ), respectively. These cost functions were calculated through the previous study [15]. We assume that we mapped the value of α, β, and γ to the user's fee as in Table. 1. For example, if the smaller value of α is, the smaller user's fee is and vice versa. Assuming the user wants to get to the car park with the shortest distance, then the probability is that he will encounter a chargeable car park with a high fee. β is similar to α, but its value is smaller because more parking spaces will have a smaller charge rate. The value of γ is the smallest value and reverse with α and β because if users select a larger probability, γ means that users are more interested in charging rates in the car park.

TABLE 1. The table of cost function.

Range of Values	Cost function		
	Cost(α)	Cost(β)	Cost(γ)
0.0 - 0.1	10	5	10
0.1 - 0.2	20	10	9
0.2 - 0.3	30	15	8
0.3 - 0.4	40	20	7
0.4 - 0.5	50	25	6
0.5 - 0.6	60	30	5
0.6 - 0.7	70	35	4
0.7 - 0.8	80	40	3
0.6 - 0.7	90	45	2
0.7 - 0.8	100	50	1

Table. 1 illustrates a cost function that considers the factors influencing the user's decision to choose a parking space. Based on the above analysis, the system will give the user the following options:

- Shortest distance solution: Users want to use the nearest car park. The distance between the user and suitable car parks in the area is calculated based on the GPS address of the user and the GPS address of car parks in the area. We assumed that the GPS coordinates of the user U_i are $u_i(x_i, y_i)$ and of the car park P_j are $p_j(x_j, y_j)$. Where x is the latitude, and y is the longitude. We calculate the distance between the user and the car park is d_{ij}. The user will choose the car park with the smallest value of d_{ij} corresponding to the smallest value of the search function.
- Least cost solution: Users want the car park at their destination with the lowest hourly cost. After searching, the system calculates and returns the appropriate result using the database. That means we can get a result from Eq. (2) if we set $\alpha = 0$ and $\beta = 0$. The result corresponds to the smallest value of the search function.
- Best space solution: Users want an optimal car park at an appropriate distance and price with a high probability of many free parking spaces. Here, users do not have to waste time waiting to park. This solution helps the system distribute the parking-load requirement to all the car parks in the network, avoiding the time-wasting parking bottleneck phenomenon where many users go to the same car park. This also generates higher profits through the effective utilization of available parking spaces.
- Optimal solution (based on the cost function).

To find the optimal results for the user, we picked an appropriate set of values (α, β, γ) based on the simulation and implemented results. The result corresponds to the smallest value of the search function in Eq. (2). Users who want to reserve a parking space will be able to use our software client on a smartphone. At the interface of the software system, we offer three options for users: selection based on the shortest distance to a car park, selection based on the parking

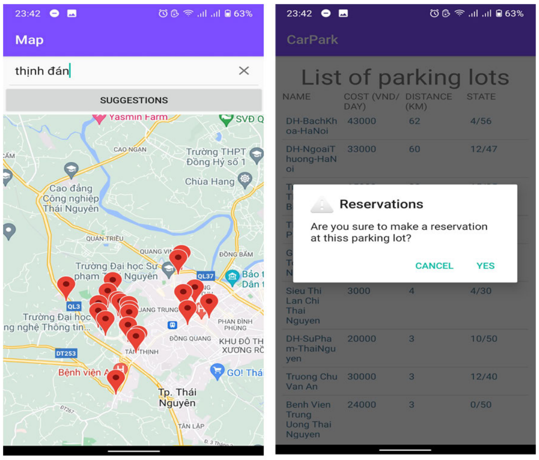

FIGURE 2. Illustration of parking reservation service design.

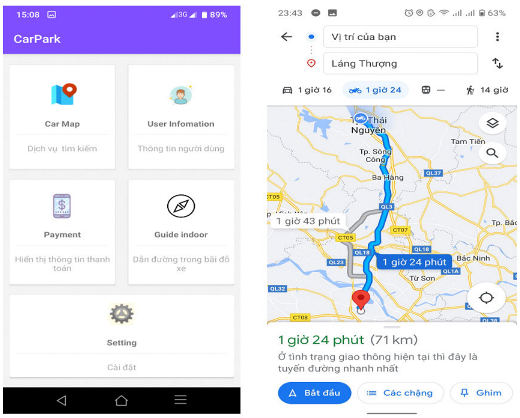

FIGURE 3. Illustration of the deployment of navigation service outside the parking lot.

fee, and selection based on the optimization of three factors: the shortest distance, the parking fee, and the total number of free parking spaces. The design of the parking reservation service and the parking network on the application software are depicted as shown in Fig. 2.

The comparison analysis between these methods will be described in the next section.

B. GUIDANCE SERVICES

To help users reach the reserved parking spot conveniently and quickly, we have implemented two guidance methods, including outside and inside parking lot navigation. After successfully booking a slot, the outside phase guides users to their parking places. For this, we designed a navigation system by specifying detailed directions on a map. This map is based on Google Maps but was supplemented with detailed information about the car parks nearest to the user's GPS location, enabling users to find suitable parking places and save time.

The indoor guidance phase helps drivers get to their desired parking space. The user can use the mini-map with

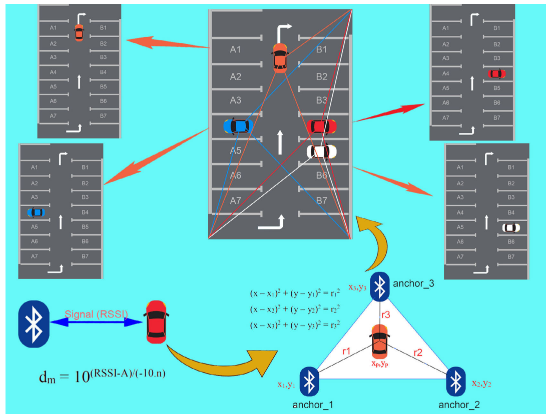

FIGURE 4. Illustration of the deployment of indoor guidance service.

the instructions in the car park to reach the parking space. For outdoor parking lots, we can use GPS technology for navigation, as shown in Fig. 3. However, many parking systems are obscured; in basements of shopping malls or large buildings, the GPS signal no longer works, and we must build a navigation system inside the parking lot. Here, we make an indoor navigation system using Bluetooth Low Energy devices (IPSB); the deployment diagram is described in Fig. 4.

Our indoor navigation systems include hardware and software implementations. The hardware is Bluetooth low-energy devices used as anchors to transmit the Bluetooth signal to the receiver, which we call the TAG. This receiver is typically mounted on the vehicle or the user's smart mobile device. We arrange these anchors at fixed positions in the parking lot; the positions of these anchors are known in advance. Usually, these anchors will be placed at corner positions in the parking lot. These anchors must be arranged so that at each time, the TAG receiver always collects signals from at least three anchors located at different locations.

Based on the Received Signal Strength Indicator (RSSI) value at the tag, the distance value between the transmitter (anchor) and receiver (tag) is estimated. The RSSI value is calculated according to [33] as follows:

$$RSSI_{dBm} = -10 \times n \times log_{10}(d) + A_0 \quad (3)$$

where $RSSI$ is the received signal strength at a tag, n is the path-loss parameter of the wireless transmission environment, d is the estimated distance between the anchor and the tag, and A_0 is the RSSI value at a distance of one meter. From Eq. (3), we can infer the relationship between the estimated distance and the received RSSI value, as depicted in Eq. (4):

$$d = 10^{(RSSI-A_0)/(-10\times n)} \quad (4)$$

To enhance accuracy and reduce errors, we will take the estimated distance value as the average of several data points measured in each step. The number of data points for each step is 20. The resulting value will then be the average of them. After estimating the distance from at least three anchors to the TAG receiver, the system will use the Triangular algorithm to determine the vehicle's location in the parking lot. An illustration of the implementation algorithm is described in Alg. 1.

Algorithm 1 The Indoor Vehicle Position Calculation

```
1:  Input: anchorX_distance
2:  Output: Vehicle location
3:  Count the distance_temp each time
4:  distance_time ← []
5:  while count < 20 do
6:      if count = 20 then
            Break
7:      end if
8:  end while
9:  Calculate the distance
    anchorX_distance_final = ∑anchorX_distance/count;
10: Compute vehicle location
    vehicle_location = Algorithm(∑anchorX_distance_final);
    =0
return
```

Receiving signals from 3 different anchors supports it to determine the current position of the TAG receiver (position of the vehicle in the parking lot), as described in the following section. We developed software that calculates the current position of the vehicle in the parking lot and gives the driver directions to the desired parking spot. The software runs on the user's smartphone. The calculation of the vehicle positions based on the Triangular algorithm is depicted in Fig. 5. It can be assumed that the anchors are arranged on the same plane. The calculation of positions in 3D coordinates is converted to the calculation of positions in 2D coordinates using a compression algorithm. Therefore, we consider the position result obtained in 2D space approximately equal to the position value in 3D space. Suppose the position of the anchors is $P(x_i, y_i)$, and the vehicle's position is $V(x_p, y_p)$. The RSSI signal value received from each anchor device will correspond to a distance r_i, i = 1, 2, 3. From Eq. (5), we can calculate the vehicle's position in the parking lot.

$$\begin{aligned}(x_p - x_1)^2 + (y_p - y_1)^2 &= {r_1}^2\\(x_p - x_2)^2 + (y_p - y_2)^2 &= {r_2}^2\\(x_p - x_3)^2 + (y_p - y_3)^2 &= {r_3}^2\end{aligned} \tag{5}$$

In actual implementation, due to an unstable Bluetooth signal, we propose attaching multiple anchor transmitters at the exact location to improve accuracy, thus increasing the probability of receiving a stable signal. The RSSI value will be the average value of the anchors located at this same location. Suppose M BLE devices are located at the corners of the parking lot, denoted by the navigation devices A_i, $i = 1, 2, \ldots M$. Based on the RSSI values obtained from these

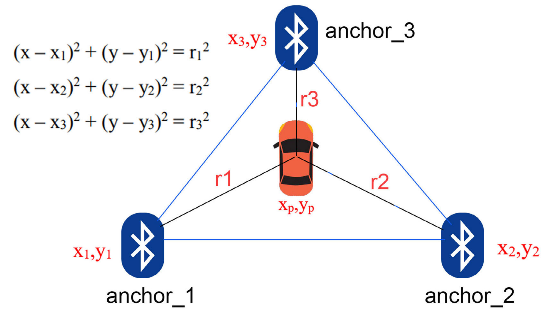

FIGURE 5. Calculate the vehicle's position based on BLE signals.

devices, we will calculate the position of the driver. The RSSI value received at each navigation device shall be the average of the 20 times obtained. For data collected with less than 20 values, we will use the existing data to calculate the average RSSI value. If the number of received data values is more significant than 20 values, then we calculate the average RSSI value by taking the 20 earliest values obtained to calculate the average RSSI value. The number of data values can be changed according to different circumstances. As the vehicle moves into the parking lot, data values will decrease. A newly received data value is compared with the existing mean. If the deviation does not exceed a given threshold, this data is saved and used to calculate the new mean RSSI value. If the threshold is exceeded, this RSSI value is discarded. The average values of M navigation devices are named r_i, $i = 1, 2 \ldots M$. After stacking the navigation devices, we will have N new navigation devices named B_j., $i = 1, 2, 3, \ldots N$ ($M = N * k$, where k is the superposition factor). R_i is the RSSI value of the navigation device B_j. We will rely on the RSSI value obtained from the navigation devices to determine the vehicle's current position in the parking lot. The proposed algorithm for calculating the value of R_i and filtering out the RSSI deviation is in Alg.2 as follows:

Algorithm 2 Filter RSSI Deviation Error

```
1: for each r_i ∈ [1, N]
2: end for
3: if r_i − r_{i+N} ≤ Deviation threshold value
4:    R_i = Average(r_i, r_{i+N}, time)
5: end if
   =0
```

where $average(r_i, r_{i+N}, time)$ is the expression to calculate the value of R_i from r_i and r_{i+N}, $time$ is the average number of executions. In this case, r_i and r_{i+N} are stacked in the same corner. The Eq.(6) to calculate the mean is:

$$Average\,(r_i, r_{i+N}, time) = \frac{\sum\left((r_i + r_{i+N})/2\right)}{time} \tag{6}$$

C. PAYMENT SERVICE

A payment service system is implemented on the user's application software. Each user has an account to top up and pay parking fees. Based on the parking location and parking time in the parking lot (collected from the vehicle's parking entry/exit data), determine the cost of money that the user has to pay. Fig. 3 describes the payment service implementation interface.

IV. SYSTEM ANALYSIS AND EVALUATIONS

A. SYSTEM PERFORMANCE METRICS

1) TIME COMPLEXITY AND EXECUTION TIME

Time complexity is the time it takes for the proposed algorithms in our system to complete. It is a function that depends on the input factors of the system, such as the number of sensor nodes and the number of fog nodes in the SP system. It represents the complexity of algorithms in the system according to these input parameters. In our algorithm, the time complexity is $O(n * K)$, where n is the number of sensor nodes, and K is the number of Fog nodes in our system. Execution time is the total running time to complete the proposed system's tasks. In the Fog-based system, data from parking lots is processed and calculated through Fog nodes and the results are sent to the Cloud server, thus reducing the traffic load in the network compared to directly calculating data on the Cloud server. This leads to reduced processing costs and system execution time.

2) SERVICE LATENCY

Assuming that we have a total of N smart parking lots stored in set C ($C = c_1 + c_2 + \ldots + c_N$) and the total volume of parking data to be processed at time t is V_t, V_t will be the total volume of data collected from sensors and cameras at individual smart parking lots ($V_C^t = v_{c_1}^t + v_{c_2}^t + \ldots + v_{c_N}^t$). Denote that the total time consumed to process parking lot data at any time t is T_S^t. According to [34], we can determine this total time as Eq.(7) follows:

$$T_S^t = T_c^t \times \sum_{i=1}^{C} v_{c_i}^t \tag{7}$$

where, T_c^t is the data processing time at each parking lot. We call that T_F the total time spent collecting parking lot data, calculating and returning service results to the system's end user based on the Fog computing platform. T_F can be calculated through the following Eq.(8):

$$T_F = T_S^t + T_{cf}^t + T_{fa}^t \tag{8}$$

where T_{cf}^t is the data transmission time from the parking lot to the fog node, and T_{fa}^t is the data transmission time from the fog node to the user's application software. For a system only based on the Cloud computing platform, the total delay for processing data T_C is determined according to Eq.(9):

$$T_C = T_S^t + T_{cc}^t + T_{ca}^t \tag{9}$$

where T_{cc}^t is the data transmission time from the parking lot to the Cloud server, and T_{ca}^t is the data transmission time from the Cloud server to the user's application software.

3) NETWORK USAGE

The total network usage of the Cloud-based system and the Fog-based system is calculated in Eq.(10) and Eq.(11) as follows:

$$NetU_C = \frac{T_C \times S_{tuple}}{T_{MRT}} \tag{10}$$

$$NetU_F = \frac{T_F \times S_{tuple}}{T_{MRT}} \tag{11}$$

where S_{tuple} represents the total size of tuple, and T_{MRT} represents the maximum running time of the system.

4) ENERGY CONSUMPTION

The energy consumption of the SP system can be calculated according to research [35], Eq.(12) defines the energy consumption as follows:

$$E_c = E_{ct} \dotplus (T_n - T_{LUUT}) \times L_{HLU} \tag{12}$$

where E_{ct} is the current energy consumption, T_n is the now time, T_{LUUT} is the last utilization update time and L_{HLU} is the last utilization of the host.

B. EVALUATION CASE STUDY

Assume that smart parking service models deployed based on Fog computing technology include 04 parking lots, as shown in Fig. 6. Parking lots have cameras to recognize available parking positions and RFID cards to authenticate in/out of the parking lot. The number of cameras and RFID tags will be changed to evaluate the performance of the parking system based on Fog technology. The data of these parking lots are collected and immediately calculated by each Fog node; the data after the calculation is completed will be sent to store at the Cloud computing server.

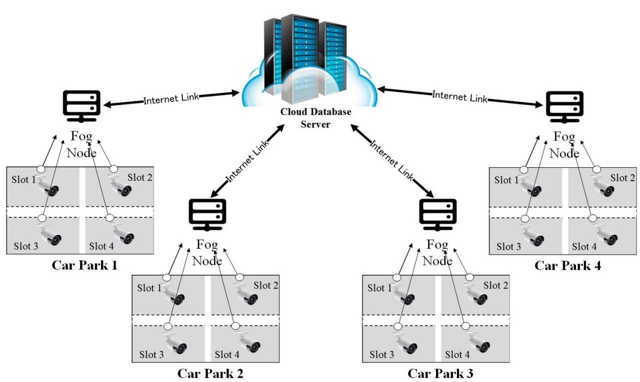

FIGURE 6. A 4 nodes - smart parking services model based on Fog computing.

We compare and evaluate the implementation of smart parking services based on Fog computing technology and traditional cloud computing technology when the number of sensors, RFID tags, and cameras changes. The comparison is based on the iFogSim simulation tool described in [28]. The comparisons taken into account are the network usage of the SP system, the latency of the service request being processed, and the execution time. The simulation settings for evaluating the performance of the SP system based on fog computing technology are shown in Table. 2.

TABLE 2. Simulation parameters.

Parameter	Cloud	Proxy Server	Fog	Sensor/Camera
CPU (MIPS)	44800	2800	2800	500
RAM (MB)	40000	8000	8000	1000
Busy Power	16*103	107.339	107.339	87.53
Idle Power	16*83.25	83.4333	83.4333	82.44
Uplink Bandwidth (MB)	100	10000	1000	10000
Downlink Bandwidth (MB)	10000	10000	10000	10000

In a Cloud-based system, all real-time parking data is processed at the Cloud server. Therefore, when the number of parking requests entering the system is large, the traffic load in the system network will increase. Deploying a system based on Fog computing technology helps data calculations to be immediately processed at Fog nodes before uploading to the Cloud server. Therefore, it significantly reduces the traffic load in the system network, which also means reducing the network usage of the Fog-based system. In Fig. 7, we compare network usage as the number of cameras and RFID tags in the parking lot changes. The results show that the deployment system based on Fog computing technology has much lower network usage than Cloud Computing.

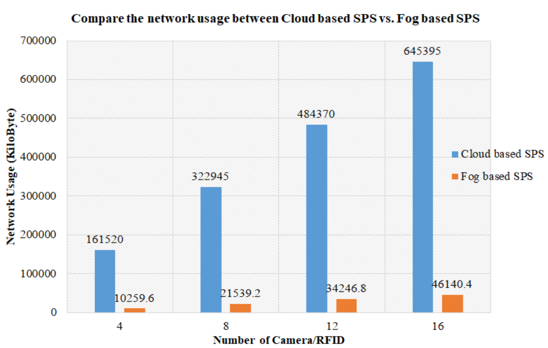

FIGURE 7. Compare the Network usage of the SP system based on cloud vs. Fog computing.

In a Fog-based system, most data processing and calculation operations are performed on Fog nodes located close to individual parking lots. These individual parking lots do not need to cooperate directly with the Cloud server. This reduces the overall latency of the SP system. Compare service latency of the SP system based on Cloud and Fog computing when the number of Fog nodes has changed, as shown in Fig. 8. We also find that the SP system based on Fog technology has a minimal latency. When the number of Fog nodes increases in the range from 4, 8, 16, 32 nodes, the latency in the Cloud system is very large, while the latency in the Fog system is negligible. This is suitable for deploying real-time applications.

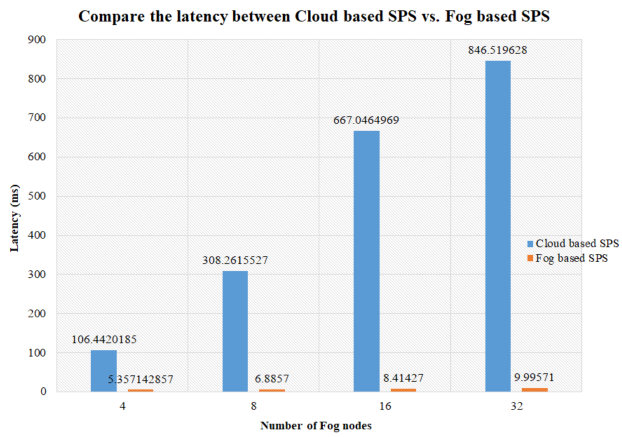

FIGURE 8. Cloud vs. Fog latency comparison.

We evaluate the energy consumption of the parking system based on Cloud technology and based on Fog technology, as shown in Fig. 9. We can see that as the size of the parking system increases, the energy consumption also increases, and the system based on Fog technology always consumes less energy than the system based on Cloud technology. This result can be inferred from the architecture of the two systems.

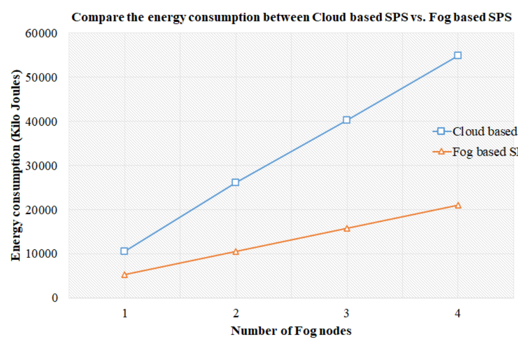

FIGURE 9. Cloud vs. Fog energy consumption comparison.

We also compare the execution time of the Fog-based and Cloud-based systems for the above scenario with a changing number of Fog nodes. The results obtained are as depicted in Fig. 10. We can see that as the size of the parking lot increases, the execution time also increases, and the execution time of the Fog system is smaller than that of the Cloud system in all cases. This is because calculating parking lot data through nearby Fog nodes saves time compared to transmitting data straight to the cloud server at a distance.

Additionally, to evaluate the quality of smart parking services, the parameters of average parking waiting time and user's parking cost need to be considered. In this study, we compare the average waiting time and the user's parking cost in consideration of the cost function as mentioned above. The comparison is shown in Fig. 11. The alpha's

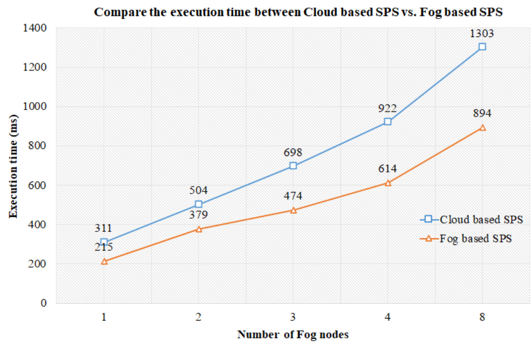

FIGURE 10. Cloud vs. Fog execution time comparison.

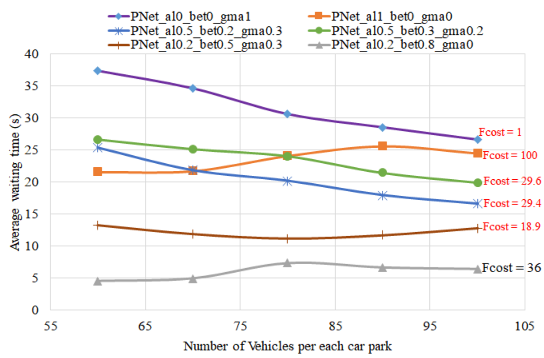

FIGURE 11. Illustrate the cost function that depends on distance, parking costs, and parking availability.

value changes from $\alpha = 0, 0.2, 0.3, 0.5, 1$. The beta's value changes from $\beta = 0, 0.2, 0.3, 0.5, 1$, and the gamma's value changes from $\gamma = 0, 0.2, 0.3, 0.5, 1$. It can be assumed that the price for parking at car park 1 is $1\$/h$, car park 2 is $1.5\$/h$, car park 3 is $1.2\$/h$, car park 4 is $2\$/h$, and car park 5 is $0.8\$/h$. We can see that with the shortest path option ($\alpha = 1, \beta = 0, \gamma = 0$), the average waiting time of the user is greater than 20 minutes, which is not the optimal value. Besides, the parking fee is highest with $F_{cost} = 100$. If we choose the least cost option regarding parking fee, the average waiting time that vehicles wait for parking is the longest and more than 25 minutes, but the parking fee that users should pay is the smallest with $F_{cost} = 1$. And with the best space option, Fig. 11 shows the average waiting time is minimal. With the set of values ($\alpha = 0.2, \beta = 0.8, \gamma = 0$), it can be seen that the user has the smallest average waiting time and smaller than 10 minutes. This set of values coincides with the optimal values of the waiting time as in [15], but we could see the parking fee users should pay is still not optimal with $F_{cost} = 36$. This is because the value of α is 0, which means that users only care about the wait time for parking without regard to the fee they should pay. In this paper, the set of values ($\alpha = 0.2, \alpha = 0.5, \gamma = 0.3$) was offered to utilize because the waiting time may be acceptable by users, and the cost of parking fees is near optimal values. This will bring a lot of significance for users with long parking times.

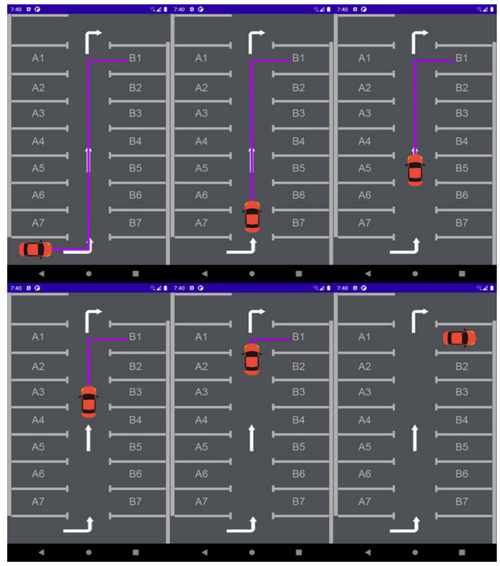

FIGURE 12. Implementation of indoor navigation service.

C. IMPROVE THE ACCURACY OF INDOOR NAVIGATION SERVICE FOR SP SYSTEM

To enhance the accuracy of the navigation service inside the parking lot, as described above, we propose to implement a positioning enhancement system using Bluetooth Low Energy (BLE) technology. These indoor navigation systems help improve the SP service in the case of parking locations located in the building or the basement of the garage. The navigation process is depicted in Fig. 12 below. The user will be assigned a given parking space when entering the system.

To evaluate our proposed indoor guidance method, we have deployed it on the actual parking lot with the size of $30m \times 50m$; the layout diagram is shown in Fig. 13. In which the car is at the entrance hall and begins to move to parking slot B1. We place three fixed anchors at the corners of the parking lot, so the coordinates of the anchors on the map are always known. We implement a system evaluation scenario where the error margin value is set to 10, 20, and 30 cm, respectively, and the vehicle speed changes to 20km/h and 30km/h, respectively. With the distance traveled by car in the parking lot in the shape of an L, after using three cases of the above error distance to filter the data, we get the distance of the car traveling as points with coordinates (X_1, Y_1, Z_1), (X_2, Y_2, Z_2), etc. in fact, as follows:

From Fig. 13, we can see that when the margin error cases increase from 10 cm, 20 cm to 30 cm and with the case of the vehicle moving at 20 km/h, the vehicle's moving path will deviate much from the actual path. When choosing an error margin of 10 cm, the measured results show us a higher accuracy of the vehicle's position than the other cases because the significant error has been eliminated. Therefore, to calculate the vehicle position, we can take the average value of the points located close to each other and give the

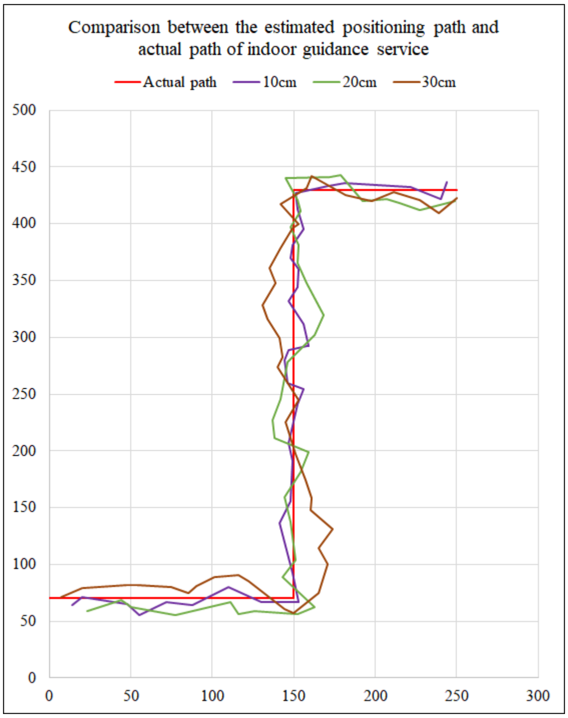

FIGURE 13. Implementation of indoor navigation service.

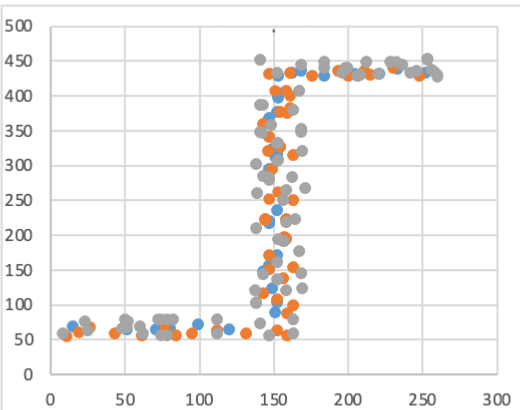

FIGURE 14. Illustrate vehicle position data at different travel speeds.

results closest to reality. Still, when removing the high error, the parking location data points will be reduced.

Fig. 14 describes the measured results with the same distance error of 10 cm, but vehicles move with different speeds of 10 km/h, 20 km/h, and 30 km/h, respectively. The faster the vehicle moves, the fewer dots will be obtained due to fewer positions.

V. CONCLUSION

This paper constructs a complete group of services for smart parking lots, from when the driver searches and makes a reservation to successfully parking at the desired parking spot. To enhance system performance, we propose to deploy these services on parking lots based on IoT technology and Fog computing. The application of Fog computing helps to reduce the delay time of services and increase the throughput of the system compared to parking lots that are only deployed based on Cloud computing technology. An improvement in the indoor navigation system based on BLE technology enhances the performance of the indoor guidance service. The analysis and evaluation of the implementation of these services in our study demonstrated that our method worked well in the experiment. In the near future, we will execute several advancements in the quality of the SP system, such as improving the parking system organization algorithms to optimize the use of parking resources and reduce the user's parking costs. In addition, to enhance the performance of the SP system and speed up the processing of parking requests, AI algorithms and camera vision will be applied to satisfy smart parking assistance services to the users' desires.

ACKNOWLEDGMENT

Ministry of Education and Training, Vietnam (Grant number: B2020-TNA-18).

REFERENCES

[1] K. Sha, T. A. Yang, W. Wei, and S. Davari, "A survey of edge computing-based designs for IoT security," *Digit. Commun. Netw.*, vol. 6, no. 2, pp. 195–202, May 2020.

[2] M. Lee, "An empirical study of home IoT services in South Korea: The moderating effect of the usage experience," *Int. J. Hum.-Comput. Interact.*, vol. 35, no. 7, pp. 535–547, Apr. 2019.

[3] I. Khajenasiri, A. Estebsari, M. Verhelst, and G. Gielen, "A review on Internet of Things solutions for intelligent energy control in buildings for smart city applications," *Energy Procedia*, vol. 111, pp. 770–779, Mar. 2017.

[4] A. Sumalee and H. W. Ho, "Smarter and more connected: Future intelligent transportation system," *IATSS Res.*, vol. 42, no. 2, pp. 67–71, Jul. 2018.

[5] N. P. Rocha, A. Dias, G. Santinha, M. Rodrigues, A. Queiros, and C. Rodrigues, "Smart cities and public health: A systematic review," *Procedia Comput. Sci.*, vol. 164, pp. 516–523, Jan. 2019, doi: 10.1016/j.procs.2019.12.214.

[6] M. N. K. Boulos and N. M. Al-Shorbaji, "On the Internet of Things, smart cities and the WHO healthy cities," *Int. J. Health Geograph.*, vol. 13, no. 1, p. 10, Mar. 2014.

[7] Z. Din, D. I. Jambari, M. M. Yusof, and J. Yahaya, "Critical success factors for managing information systems security in smart city enabled by Internet of Things," *Int. J. Adv. Res. Eng. Technol.*, vol. 11, no. 12, pp. 1108–1120, Dec. 2020.

[8] R. Hassan, F. Qamar, M. K. Hasan, A. H. M. Aman, and A. S. Ahmed, "Internet of Things and its applications: A comprehensive survey," *Symmetry*, vol. 12, no. 10, p. 1674, Oct. 2020.

[9] *Smart Parking*. Accessed: Sep. 12, 2023. [Online]. Available: https://www.smartparking.com/

[10] *PlacePod*. Accessed: Sep. 12, 2023. [Online]. Available: https://www.pnicorp.com/placepod/

[11] *Sitraffic Scala*. Accessed: Sep. 12, 2023. [Online]. Available: https://www.yunextraffic.com/global/en/portfolio/traffic-management/centrals/sitraffic-scala

[12] E. Ismagilova, L. Hughes, N. P. Rana, and Y. K. Dwivedi, "Security, privacy and risks within smart cities: Literature review and development of a smart city interaction framework," *Inf. Syst. Frontiers*, vol. 22, no. 4, pp. 1–22, Jul. 2020.

[13] C.-C. Huang, Y.-S. Tai, and S.-J. Wang, "Vacant parking space detection based on plane-based Bayesian hierarchical framework," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 9, pp. 1598–1610, Sep. 2013, doi: 10.1109/TCSVT.2013.2254961.

[14] J. Chinrungrueng, S. Dumnin, and R. Pongthornseri, "iParking: A parking management framework," in *Proc. 11th Int. Conf. ITS Telecommun.*, Aug. 2011, pp. 63–68.

[15] Y. Geng and C. G. Cassandras, "New 'smart parking' system based on resource allocation and reservations," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1129–1139, Sep. 2013, doi: 10.1109/TITS.2013.2252428.

[16] C. Matt, "Fog computing: Complementing cloud computing to facilitate industry 4.0," *Bus. Inf. Syst. Eng.*, vol. 60, no. 4, pp. 351–355, Aug. 2018.
[17] X. Chen and L. Wang, "Exploring fog computing-based adaptive vehicular data scheduling policies through a compositional formal method—PEPA," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 745–748, Apr. 2017.
[18] A. Somani, S. Periwal, K. Patel, and P. Gaikwad, "Cross platform smart reservation based parking system," in *Proc. Int. Conf. Smart City Emerg. Technol. (ICSCET)*, Jan. 2018, pp. 1–5, doi: 10.1109/icscet.2018.8537282.
[19] V. Rajvanshi, S. Chaturvedi, D. Yadav, and L. Sharma, "Smart parking system using sensors and cloud based network for smart cities applications," in *Proc. 2nd Smart Cities Symp. (SCS)*, Mar. 2019, pp. 1–4.
[20] S. Saharan, N. Kumar, and S. Bawa, "DyPARK: A dynamic pricing and allocation scheme for smart on-street parking system," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 4, pp. 4217–4234, Apr. 2023, doi: 10.1109/TITS.2022.3230851.
[21] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st Ed., MCC Workshop Mobile Cloud Comput. (MCC)*, Helsinki, Finland, Aug. 2012, pp. 13–16.
[22] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.
[23] A. M. S. Maharjan and A. Elchouemi, "Smart parking utilizing IoT embedding fog computing based on smart parking architecture," in *Proc. 5th Int. Conf. Innov. Technol. Intell. Syst. Ind. Appl. (CITISIA)*, Nov. 2020, pp. 1–9.
[24] C. Lai, Q. Li, H. Zhou, and D. Zheng, "SRSP: A secure and reliable smart parking scheme with dual privacy preservation," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10619–10630, Jul. 2021.
[25] N. Sangeetha, S. Amutha, A. R. Deepa, and S. Joyal, "IoT based smart parking assistant for smart cities," in *Proc. Int. Conf. Edge Comput. Appl. (ICECAA)*, Oct. 2022, pp. 593–598.
[26] M. A. U. Rehman, M. S. U. Din, S. Mastorakis, and B.-S. Kim, "FoggyEdge: An information-centric computation offloading and management framework for edge-based vehicular fog computing," *IEEE Intell. Transp. Syst. Mag.*, vol. 15, no. 5, pp. 78–90, Sep./Oct. 2023, doi: 10.1109/MITS.2023.3268046.
[27] C. Tang, X. Wei, C. Zhu, W. Chen, and J. J. P. C. Rodrigues, "Towards smart parking based on fog computing," *IEEE Access*, vol. 6, pp. 70172–70185, 2018.
[28] K. S. Awaisi, A. Abbas, M. Zareei, H. A. Khattak, M. U. S. Khan, M. Ali, I. U. Din, and S. Shah, "Towards a fog enabled efficient car parking architecture," *IEEE Access*, vol. 7, pp. 159100–159111, 2019.
[29] 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 *Proc. IEEE 90th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2019, pp. 1–5.
[30] T. Fernández-Caramés, P. Fraga-Lamas, M. Suárez-Albela, and M. Vilar-Montesinos, "A fog computing and cloudlet based augmented reality system for the industry 4.0 shipyard," *Sensors*, vol. 18, no. 6, p. 1798, Jun. 2018.
[31] O. Abdulkader, A. M. Bamhdi, V. Thayananthan, K. Jambi, and M. Alrasheedi, "A novel and secure smart parking management system (SPMS) based on integration of WSN, RFID, and IoT," in *Proc. 15th Learn. Technol. Conf. (L&T)*, Feb. 2018, pp. 102–106.
[32] E. C. Anderson, K. C. Okafor, O. Nkwachukwu, and D. O. Dike, "Real time car parking system: A novel taxonomy for integrated vehicular computing," in *Proc. Int. Conf. Comput. Netw. Informat. (ICCNI)*, Oct. 2017, pp. 1–9.
[33] P. Bencak, D. Hercog, and T. Lerher, "Indoor positioning system based on Bluetooth low energy technology and a nature-inspired optimization algorithm," *Electronics*, vol. 11, no. 3, pp. 1–27, 2022, doi: 10.3390/electronics11030308.
[34] S. R. Hassan and M. Rashad, "Cloud computing to fog computing: A paradigm shift," in *Edge Computing—Technology, Management and Integration*, S. Goundar, Eds. IntechOpen, 2023. [Online]. Available: https://www.intechopen.com/chapters/86772
[35] D. Rahbari and M. Nickray, "Scheduling of fog networks with optimized knapsack by symbiotic organisms search," in *Proc. 21st Conf. Open Innov. Assoc. (FRUCT)*, Nov. 2017, pp. 278–283.

THANH-NAM PHAM received the Ph.D. degree in information engineering and computer science from Feng Chia University, Taiwan, in 2018. He is currently with the Department of Electronics and Communications Technology, Thai Nguyen University of Information and Communication Technology, Vietnam. His research interest includes the Internet of Things technologies and applications.

DUC-THO MAI received the B.Sc. degree in electronics and telecommunications and the M.Sc. degree in telecommunications engineering from the Hanoi University of Science and Technology (HUST), Vietnam, in 2010 and 2012, respectively, and the Ph.D. degree from the Computer and Network Engineering Department, University of Electro-Communications (UEC), Tokyo, Japan, in March 2023. He is currently a Lecturer with the Academy of Cryptography Techniques, Hanoi, Vietnam. His research interests include machine learning, deep learning, and their applications in the IoT security.

XUAN-KIEN PHAM received the master's degree in software engineering from the Hanoi University of Science and Technology (HUST), in 2016. He is currently with the Department of Electronics and Communications Technology, Thai Nguyen University of Information and Communication Technology, Vietnam.