Smart Parking: An Efficient System for Parking and Payment

Md Ezaz Ahmed[1]*, Mohammad Arif[2]*, Mohammad Khalid Imam Rahmani[3]*, Md Tabrez Nafis[4], Javed Ali[5]
College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia[1, 3, 5]
School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India[2]
Department of Computer Science and Engineering, Jamia Hamdard, New Delhi, India[4]

***Abstract*—In addition to being a time-consuming and annoying driving experience, searching for a cheap and empty parking space also wastes fuel and pollutes the air. In densely populated cities, there are limited and expensive public parking spaces. On the other hand, private parking spaces are typically underutilized, and the parking space owners are willing to charge higher parking fees to cover the expenses of maintaining their excess parking capacity. In light of these circumstances, it is essential to look for a smart parking system that gathers and allows private parking spaces to ease the worries of public parking. An Internet of Things (IoT) enabled parking space recommendation system is proposed in this paper. It makes recommendations by utilizing IoT technology (traffic and parking sensors). The recommended system helps users automatically pick a spot at the lowest charge by accounting for metrics like distance, availability of vacancy at the slot, and the charges. To accomplish this, the user parking cost is calculated using performance measures. This system provides the user with a way to request a parking spot when one is available, as well as a way to recommend a new parking lot if the present one is filled. The proposed model reduces user waiting time and increases the likelihood of finding an empty slot in the parking, based on the simulation results used besides offering an anonymous payment method. The proposed system also exploited the concept of VANET as it uses onboard and roadside units. The novelty of the research is that apart from calculating the cost function it also maintains the neighbors table at each neighbor which will be shared among all as and when there is a change. We have simulated the environment in Network Simulator 3 (NS3).**

Keywords—Smart parking; Internet of Things; sensors; simulation; NS3; VANET

I. Introduction

There has been a noticeable rise in the number of cars on the road during the past ten years, which has caused significant issues with parking and traffic. Typically, drivers search the streets for open parking spaces by driving about; they only empirically locate a spot through luck and local knowledge. This practice results in significant wastage of fuel and time, and occasionally it makes it hard to find a site unoccupied during periods of high vehicle traffic. Locating a parking lot with sufficient open spaces would be one way to increase the likelihood of finding a place even though the user's destination can be somewhat far from this parking spot. Another option is to create a system that allows the driver to manually select a free parking space after it is displayed to them. This is not the best option, though, as it requires the drivers to choose a parking spot on their own, which is a hassle in and of itself. Second, there's a chance that the driver will find the parking space occupied when they get there due to heavy traffic on the path leading to the designated place.

IoT technology, which exploits enormous technological advancements, has transformed nearly every aspect of everyday existence, such as parking systems. Motivated by these novel opportunities, a smart parking system is proposed to automatically suggest unoccupied parking spaces to drivers in search of them. This reduces the time required to locate unoccupied parking spaces and lowers expenses related to employing personnel to manage the manual parking system [1-3]. One such method is a parking spot booking system that uses wireless networking technologies like ZigBee, radio frequency identification, and the Internet. Drivers can use their devices to book free parking spots nearby and get information about them by using a unique identification assigned to every car in a parking spot booking system [1]. However, due to laws in some regions (like Santander, Spain), reserving parking spaces is often not possible in advance.

Additionally, it is crucial to offer the least congested path to each available parking spot while recommending parking spots, accounting for the traffic volume on the route. Let's examine a rush hour situation where traffic in the city core is at its worst for a better understanding of the significance of traffic congestion. The streets will be extremely clogged in this circumstance due to many drivers searching for parking spaces [4-9]. When a driver gets a vacant spot that is available and moves in that direction, there's a good chance that when they get there, it will already be occupied as other users are also searching for vacant spots, and traffic overcrowding delays their arrival. This issue is addressed by helping users choose the closest parking space and by informing them of the availability (occupancy data) of parking lots.

In contrast to the conventional approaches, it has been found that a sizable percentage of parking spaces are privately held and not under the direct jurisdiction of the local transportation authority. These parking spaces in the private sector (such as homes and workplaces) are never occupied when the occupants are away on vacation or not in the office. Furthermore, suppliers typically shell out a lot of cash for the purchase and upkeep of these exclusive areas. As a cost-effective way to offset their costs, they are thus willing to charge for their parking spaces. These encourage us to consider the time that cruising drivers will save and the amount of traffic congestion that will be reduced.

*Corresponding Author.

A. Scope and Objectives

In this work, an overview and the system architecture of the proposed model have been offered for the endorsements of parking places and the suggested routes using parking and traffic sensors. By allowing drivers to analyze the probable availability of parking lots in an accessible interface and choose a parking lot, the API aims to ease interoperability and reusability while facilitating the integration of smart parking applications with other Internet of Thing applications.

In conclusion, the main obstacles are as follows: (1) Devise a mechanism that incentivizes private parking spot owners to voluntarily make their spaces available to the public, as opposed to public parking spots, which are concealed from view; and (2) Accomplish the payment procedure once drivers have finished parking, even in cases where owners and drivers remain anonymous to the server. (3) The public and private parking systems must be combined into a system, allowing drivers to select and reserve a spot based on factors like cost and distance.

B. Novelty in the Research

1) IoT technology motivates the novel opportunities of automatically suggesting unoccupied parking spaces to drivers. This reduces the time required to locate unoccupied parking spaces and the expenses of employing personnel to manage the manual parking system. The proposed system suggests a parking spot booking system that uses wireless networking technologies like ZigBee, radio frequency identification, and the Internet. Drivers can use their devices to book free parking spots nearby and get information about them by using a unique identification assigned to every car in a parking spot booking system.

2) It offers the least congested path to each available parking spot while recommending parking spots, considering the volume of traffic on the route. As a cost-effective way to offset their costs, they are thus willing to charge for their parking spaces. These encourage us to consider the time that cruising drivers will save and the amount of traffic congestion that will be reduced.

3) By providing timely and reliable parking information people equipped with smart devices can help develop the idea of smart parking.

4) Semantically enabled applications for the IoT, like smart parking systems, should be implemented through semantic web technologies and semantic data modelling to promote interoperability in the IoT.

5) The proposed model allows drivers to analyze the probable availability of parking lots in an accessible interface and choose a parking lot to ease interoperability and reusability while facilitating the integration of smart parking applications with other IoT applications.

The chance to cooperatively detect and exchange crucial information for collaborative welfare has been made possible by the recent surge in the development and use of cell phones. People who own sensors, computers, and storage devices [10-12] can now gather and provide useful data (often in the form of reports) to a server for various uses, like parking spot location. Therefore, by providing timely and reliable parking information, people equipped with smart devices can help develop the idea of smart parking [13-19].

A large number of the IoT apps that are currently in use were created vertically, concentrating on a single use case or scenario, sometimes without taking data reuse and interchange with other IoT applications into account. Because there is insufficient compatibility in IoT information and systems due to a lack of diverse data integration, this overly concentrated attention leads to bad service. Conversely, if IoT apps worked together, sharing and exploiting each other's data, chances to create new, more valuable, and effective services would arise [20]. One possible solution to help achieve the necessary interoperability is the semantic web [21]. Semantically enabled applications for the IoT, like smart parking systems, should be implemented through the use of semantic web technologies and semantic data modeling to promote interoperability in the IoT.

The measurements for parking spots are the nearest or most reliable parking spots, whereas the metrics for routes are the quickest or least congested paths [22]. A parking space that satisfies the user's past parking sensor quality rating and his trust based on experience is considered a trustworthy parking location [23]. A trust-checking component was established to locate the closest trusted parking in a recent study. This component analyses user comments and sensor quality to determine parking spot trust scores. The proposed model uses these trust scores to determine the closest trusted parking space [24, 25]. In study [26], a method for creating an effective parking lot payment system with the least human intervention is suggested. This system runs in real-time over an ESP-8266 connection on an ARM CORTEX M3 Board, which consumes little power. The study method [27] predicts parking lot occupancy using machine learning-based techniques, which are then utilized to determine occupancy-driven charges for arriving cars. The work mentioned in the research [28] contributes to the decision of the technical options and requirements for designing a smart parking system that adheres to the paradigm of efficiency and innovation. After a study of many materials on smart parking and others, the research concluded that the smart parking system has some advantages and disadvantages. Some of them are mentioned below.

Advantages of Smart Parking:

- Convenience: Using applications or real-time displays, drivers may quickly locate open parking spots, saving them time.
- Efficiency: By optimizing parking lot usage, smart solutions create more available spaces and reduce traffic.
- Diminished Emissions: Lower fuel use and emissions result from less time looking for parking.
- Security: Since automated payment systems do not need currency handling, there is a decrease in fraud.
- Data-driven Management: Parking authorities can better allocate resources and set prices thanks to the useful information they obtain about usage trends.

- Paying with a touchless device reduces the chance of spreading germs.

Disadvantages of Smart Parking:

- Cost: The technology may be costly to install and maintain.
- Technology Dependency: Problems in the system may lead to disruptions and stranding of drivers.
- Digital Divide: Some people lack access to cell phones and the necessary technological know-how to operate these devices.
- Data collecting introduces privacy concerns, particularly when not done transparently.

This document is organized as follows. Section II discusses the associated work. Section III describes the system design and architecture; Section IV outlines the system's working. Section V covers the simulation. Section VI summarizes the findings and assessments, whereas Section VII concludes the research.

II. Literature Review

A lot of effort has been put into automated parking systems since the introduction of smart cities. However, the proposed system is considerably different in that it suggests parking spots and routes that are the closest to traffic congestion, trust scores for spots to park, and probable availability of parking lots. The paper outlines the current state-of-the-art and discusses why the proposed model needs the features.

An intelligent parking system that uses IoT communication is proposed to compute the prices of parking requests made by drivers. Data on cars, the quantity of parking spaces available in parking lots, and the separation between parking lots would all be coordinated by this system. They used sensors, Arduino, and cell phones to develop a prototype.

It is suggested to integrate RFID, Wireless Sensor Network (WSN), and Ultra-High Frequency (UHF) technologies [2] to create a smart parking system. This system was created as an application to direct cars to the closest available parking space and includes software tools to track parking spot occupancy. The users can also utilize an e-wallet system based on Near-Field Communication (NFC) to pay their parking costs. The authors did not, however, assess the parking system's effectiveness; instead, they focused mostly on implementing the prototype.

It is recommended that drivers book a spot thirty minutes in advance using a smartphone app via a parking advisory system [3]. SmartValet was created as an interior and outdoor parking solution. The ID of the car is used to reserve a parking space. According to DSRC technology, the vehicle receives a map at the parking lot entry that shows the location of the assigned parking spot. An inertial navigation system fitted by SmartValet directs the car to its designated parking spot. Periodically updating the parking spot's status ensures that it is correct. The authors evaluated system installation and performance using the accuracy of the GPS and the inertial navigation system's accuracy as performance criteria.

Based on ZigBee technology, a smart parking system is suggested [5]. Through a gateway, data is transferred to the server, which updates the database. The system's application layer collects information about available parking places via the Internet, uses web services to compile data about all the distributed parking locations, and then provides the information to cars who are looking for free spots. It is a basic program, though, and does not take into account sophisticated issues like navigation, traffic congestion, or the likelihood of parking spaces being available. Moreover, the authors did not assess the system's functionality.

In study [7], it is advised to implement an automated parking system for the impaired. The IoT and smart city prospects are enhanced by this technology, which integrates smartphones, sensors, and mobile/wireless communications. DisAssist gives disabled drivers the option to reserve parking spots and gives them access to real-time availability data on accessible parking spaces nearby. DisAssist, like other current projects, takes into account the reserve of parking spaces, which isn't always feasible.

A system that takes advantage of the data gathered in smart cities is suggested in [18] and is focused on contextualization and offering users personalized parking recommendations. The contextualized information on users' tastes and behaviors is the authors' primary focus, even though they offer context-aware parking recommendations. It also provides parking area occupancy statistics and a parking recommendation system.

A smart city architecture utilizing IoT-based big data analytics was presented by System in [16]. The authors take into account a variety of sensor deployments, including those used to monitor parking, weather, water, smart homes, and automobiles in addition to surveillance and other areas. However, their primary concentration is on smart city planning employing Big Data analytics, rather than offering suggestions for smart parking.

The main research gap found so far is that in most of the work, methods have been applied to find the available space and the vehicle is parked there. Then payment is initiated. If parking is not available then, the whole algorithm is again executed. The paper proposes to find the cost function to get the nearest parking along with the neighbor table. The neighbor table will be shared among all the neighbors proactively whenever there is a change in the parking slot. All the neighbors will have the latest status of their neighbors.

III. System Model and Architecture

In the IoT, data modeling serves to enable data reuse, sharing, and interoperability amongst the applications which are having cross domains. The proposed model is intended for use with the smart parking system, primarily concerned with creating an anonymous payment system and a smart parking recommendation system.

A. System Model of Parking Sensors

Modern cities have several parking sensors installed to determine a free parking spot. Parking sensors can represent a parking location. Parking lots are organized into different parking regions to offer more useful data. Using data from parking sensors, the proposed model provides expected availability (occupancy statistics) for parking lots. The system model is depicted in Fig. 1. The trusted authority, server, driver, and supplier are the four primary elements that make up the system model. Every component has a different role.

1) *Trust authority:* The powerful Trust Authority (TA) is in charge of initializing the entire system, which entails distributing keys, generating public parameters, and registering drivers and suppliers. Unless there is a dispute in which TA can identify the identity of a targeted user, it will remain offline. A Trust Authority (TA) is an essential component of Vehicular Ad-hoc Networks (VANETs) that helps to guarantee secure and trustworthy communication. The roles of trust authority are:

a) Registration: The TA keeps track of the On-Board Units (OBUs) and Roadside Units (RSUs) of the cars connected to the network. This entails confirming their identity and providing them with special credentials.

b) Authentication: To ensure the legitimacy of messages sent by cars and RSUs and stop hostile actors from inadvertently introducing fake information, the TA authenticates the messages.

c) Key distribution: The TA securely distributes cryptographic keys to enable registered nodes to encrypt and decode messages for safe communication.

d) Revocation: To stop compromised nodes from using the forward-going network and to lessen any security risks, the TA can revoke their login credentials.

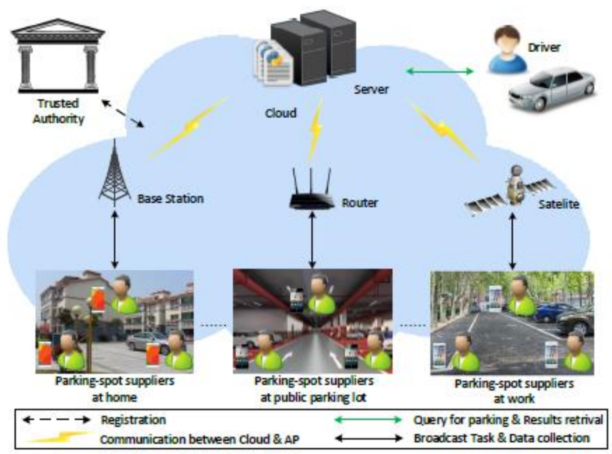

Fig. 1. System architecture.

2) *Cloud/Server:* The roles of the cloud in VANET include:

a) Enhanced scalability: VANETs can manage massive data volumes and accommodate a growing number of connected vehicles because of cloud computing's nearly limitless capabilities.

b) Better traffic management: To maximize traffic flow, lessen congestion, and boost overall transportation efficiency, real-time traffic data analysis from the cloud can be applied.

c) Advanced applications: Route optimization, individualized entertainment services, accident detection, and other advanced applications are made possible by cloud-based processing capacity.

d) Storage and analysis of data: Cloud platforms provide safekeeping and processing power for enormous datasets amassed by automobiles, providing insightful information about traffic patterns and driver conduct.

3) *Vehicle:* The vehicle requires a parking spot in the least amount of time and for an efficient way to make the payments. It has an onboard unit (OBU) that communicates with the network.

4) *Parking-Spot supplier:* The parking spot supplier keeps communicating with the cloud to inform the cloud about the latest status of the parking slots. They can be a base station/roadside unit (RSU), a router in parking in some mall or building, or a satellite to inform about open parking information.

After receiving parking inquiries from drivers and supply reports from suppliers, the server scans the database and provides drivers with matched results. The driver will be required to pay a parking fee to the supplier in exchange for her offering to lend the driver her private parking space. Driver speeds around looking for a spot in a lot that is open to the public or waits for one that is private and requires payment of a parking fee.

5) *Cloud-Based server:* The resource data supplied by local units at each parking lot is stored by this Web entity. The system allows a vehicle to seek and get details of parking lots in each car park directly, bypassing the parking unit, by instantly connecting to the IoT-based cloud server.

6) *Parking unit:* As seen in Fig. 1, this machine is situated in every parking lot and retains data about every parking spot. The following are included in the local unit:

a) Control unit: An RFID reader connects this Arduino module. After verifying the user's identity, the card reader shows the data on the screen. The module of Arduino will regulate to allow the vehicle in if the tag data or the card is accurate. To move data from the nearby parking lot to the cloud server database, the Arduino module establishes an Internet connection with the cloud server.

b) Screen: This shows details on the local parking lot's capacity, the overall percentage of presently available spots, the tag check's status, the driver card used to enter, and a small plot of the parking lot.

c) Tag or identity card: The tag or the card is used to compute the proportion of total available spaces in each parking lot and verify and validate user data.

7) *Parking network:* Assumedly, Fig. 2 depicts the deployment network in an actual setting, with labels for each parking lot.

P1 is the first parking lot, and N1 is the total number of places available. P2 is the second car park, while N2 is the total parking spots in P2. There is a total of Nn parking places in car park number n, denoted by Pn. The system has a total capacity of N = N1 + N2 + N3 +... + Nn (i.e., free spaces).

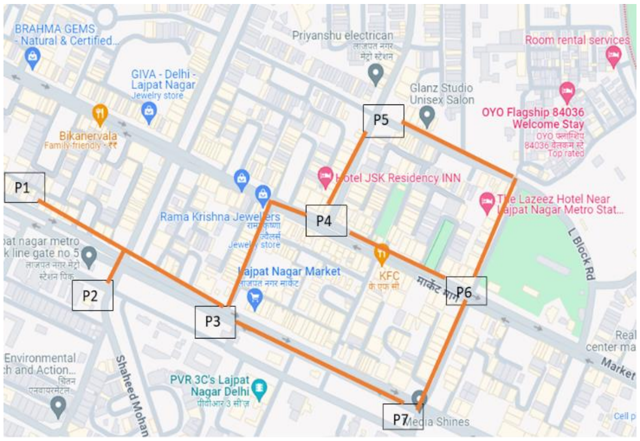

Fig. 2. Actual deployment of cars.

The deployment network is shown in Fig. 3. Every node has a queue with a set length and a neighbor table that keeps track of the network's current state. The actual distance (D) between two network nodes is shown in this picture. The D_{ij} denotes the distance from nodes P_i to P_j. Information about the neighboring nodes that are directly connected to a node can be found in its neighbor table. However, to avoid overloading the node, the vehicle queue regulates the number of vehicles sent to it. Whenever a node enters or departs the suggested system, it broadcasts a message to its adjacent nodes. Included in this message is information on all of its free resources. After receiving this message, the nearby node will update its neighbor tables.

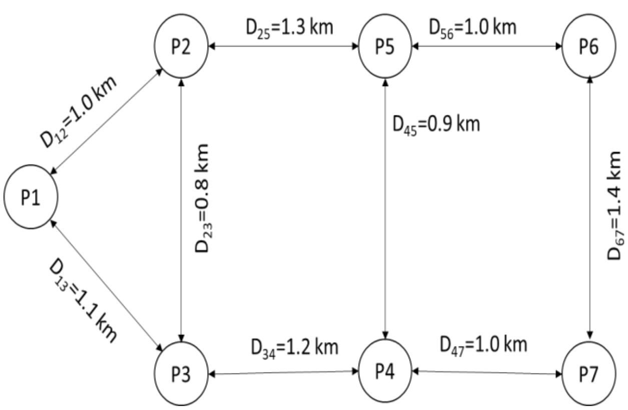

Fig. 3. Parking deployment.

The distances are shown in Fig. 3 and the available vacant slots at each parking lot P1, P2, P3, P4, P5, P6, and P7 are assumed as 10, 20, 30, 40, 50, 60, 70 respectively. In Fig. 4, these parameters are displayed using straightforward neighbor tables.

The neighbor table in every parking node holds the number of available parking vacancies currently available in the neighboring nodes, which improves the outcome of discovering available parking. To identify the neighbor table, Algorithm 1 is applied.

Algorithm 1: To find the neighbor table

1. User log in to the system
2. The user reaches any node of the parking network.
3. The user sends the request to the server through the roadside units
4. Each immediate neighbor will share the information.
5. All the nodes will enter the distance of their neighbors in a table.
6. All nodes will share their table with the source node.

The outcome of this algorithm is to find the neighbor table. This will display the quickest path between a specific parking node and other nodes. This is shown in Fig. 4.

B. Constructing the Cost Function Table of Nodes

The cost is computed between the network's nodes using a cost function called C(a,b,c). The function C(a,b,c) depends on the available parking vacancies at the parking node, the distance between two parking lots, and the charges of each parking lot for each 5 hours. In the proposed system, C(a,b,c) is a weighted link between two nodes. C(a,b,c) = ∞ indicates that two parking nodes are not directly connected. The car should be routed to the adjacent parking lot, which has the lowest value of C(a,b,c) in the neighbor table if it enters a node already full. The cost function C(a,b,c) is determined between nodes P_i and P_j, that is to say,

$$\boldsymbol{C_{ij}} = \boldsymbol{C_{ij}}(a,b,c) = \left(a \times \frac{\boldsymbol{d_{ij}}}{\boldsymbol{D_{ij}}} + c \times \frac{\boldsymbol{s_j}}{\boldsymbol{S_{up}}}\right) \times \left(b \times \frac{\boldsymbol{t_j}}{\boldsymbol{T_{up}}}\right) \quad (1)$$

where, c is the cost for the five-hour parking slot, a is a length factor, where length means the distance between the two lots, and b is an available vacancy factor at every slot.

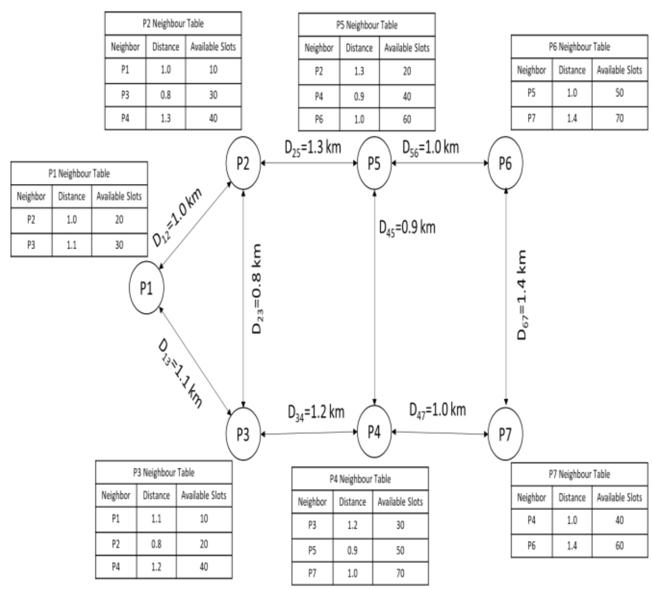

Fig. 4. The neighbor tables.

The tables for the cost function will be created using Algorithm 2 as shown in Table I.

Algorithm 2: To find the cost function table at each node

1. At each node, first find the neighbor through algorithm 1.
2. Set the weightage of distance (a), available slot (b), and the charges (c).
3. Apply the cost function formula as given: $\boldsymbol{C_{ij}} = \boldsymbol{C_{ij}}(a,b,c) = (a \times \frac{\boldsymbol{d_{ij}}}{\boldsymbol{D_{ij}}} + c \times \frac{\boldsymbol{s_j}}{\boldsymbol{S_{up}}}) \times (b \times \frac{\boldsymbol{t_j}}{\boldsymbol{T_{up}}})$
4. Get the cost function values for each neighbor of the parking node
5. Arrange the individual table as per ascending values of C(a,b,c)

It is assumed that the parking space will be reserved for at least five hours. C(a,b,c) is proportional to the total number of open spaces in the destination parking slot, to the costs, and inversely related to the distance between two nodes. a, b, and c can be changed to improve network performance, depending on which of the two parameters viz. the distance, the available vacancies, or the charges, we believe to be most crucial. The experiment yielded the values a, b, and c, which have a value of [0, 1]. We only consider the quantity of available spaces when determining the user's cost if a = 0. The cost to the user is determined only by considering the distance between two nodes if b = 0.

The cost function is obtained from Eq. (1) by considering the parking slot charges, the percentage of available vacancies at each parking lot, and the distance between two nodes. The parameters that make up the parking network are as follows: t_j is the number of available spaces at node P_j; Tup is the maximum available slot capacity and is a global parameter; s is the slot charges per five hours; and S is the maximum slot charges. The d_{ij} is the distance from lot Pi to lot P_j. Assuming the network shown in Fig. 4 has seven nodes, we compute the value of function C using the value of a, as 0.2, the value of b as 0.8, the value of c as 1, the value of D as 2 km, the value of T as 100 and the value of S as 20. Tables I to VII display the cost function tabulated for every lot with the C(a,b,c) function. This demonstrates that each node's new cost function table adheres to Eq. (1). When a parking lot is full, this routing table will determine which node the user should be forwarded to next.

TABLE I. COST FUNCTION TABLE P1 IS SORTED BY ASCENDING VALUES OF C (A, B, C)

Neighbor	Distance	Charges/ 5 hr	Available Slot	C(a, b, c)
P2	1	9	20	0.088
P3	1.1	10	30	0.1464

TABLE II. COST FUNCTION TABLE P2 IS SORTED BY ASCENDING VALUES OF C (A, B, C)

Neighbor	Distance	Charges/ 5 hr	Available Slot	C(a, b, c)
P1	1	10	10	0.048
P3	0.8	11	30	0.1512
P4	1.3	9	40	0.1856

TABLE III. COST FUNCTION TABLE P3 IS SORTED BY ASCENDING VALUES OF C (A, B, C)

Neighbor	Distance	Charges/ 5 hr	Available Slot	C(a, b, c)
P1	1.1	10	10	0.0488
P2	0.8	8	20	0.0768
P4	1.2	9	40	0.1824

TABLE IV. COST FUNCTION TABLE P4 IS SORTED BY ASCENDING VALUES OF C (A, B, C)

Neighbor	Distance	Charges/ 5 hr	Available Slot	C(a, b, c)
P3	1.2	11	30	0.1608
P5	0.9	8	50	0.196
P7	1	9	70	0.308

TABLE V. COST FUNCTION TABLE P5 IS SORTED BY ASCENDING VALUES OF C (A, B, C)

Neighbor	Distance	Charges/ 5 hr	Available Slot	C(a, b, c)
P2	1.3	9	0	0
P4	0.9	10	40	0.1888
P6	1	8	60	0.24

TABLE VI. COST FUNCTION TABLE P6 IS SORTED BY ASCENDING VALUES OF C (A, B, C)

Neighbor	Distance	Charges/ 5 hr	Available Slot	C(a, b, c)
P5	1	10	50	0.24
P7	1.4	8	70	0.3024

TABLE VII. COST FUNCTION TABLE P7 IS SORTED BY ASCENDING VALUES OF C (A, B, C)

Neighbor	Distance	Charges/ 5 hr	Available Slot	C(a, b, c)
P4	1	11	0	0
P6	1.4	8	60	0.2592

IV. SYSTEM OPERATIONS

A user must log into the proposed system to search for a parking space. A message to look for a vacant space is delivered following a successful login. The information, with the slot address of the parking lot and directions to get there, will then be sent back by the system in a response message. The function C(a,b,c), computed using the vehicle's present location and the parking lot's location, determines which parking lot to use. If the parking lot is full, the system will route the automobile to one with a minimum C(a,b,c) value. The user needs permission to enter the parking lot when he gets there. Either RFID technology or a card scan is used to obtain this authorization. This system is straightforward but efficient. The user may park if the information is accurate. If the parking lot is full, the system will send a recommendation message with the address and updated instructions for a new parking lot with a minimal fee.

Algorithm 3: Recommendation System
1. Apply Algorithm 1 to find the distances and the neighbor table.
2. Apply Algorithm 2 to find the Cost function table.
3. If C(a,b,c)= 0, no parking slot is available at that place.
4. Parking is recommended as per a lower value of C(a,b,c).
5. Finally book the parking and make the payment. a) Find Available Space b) Initiate the Booking Process c) Make payment

Three procedures are involved in the suggested system:

1) Process to find the available space and generate the neighbor table: Parking sensors are used in the proposed system to determine whether a slot is available or not. Each parking space in our suggested system will have an ultrasonic sensor installed. The roadside unit will receive the same information from the sensor if the slot is available if the automobile is parked there. The ESP8266 Wi-Fi module will then transfer this information to the server, as shown in Fig. 5. Both the neighbor table and the cost function table will receive an update with the slot's current status data.

2) Booking process: The user uses a smartphone to transmit a message to the system if he is looking for a free parking space. Upon receiving this request, the system will locate parking lot P1 with a lower value of C(a, b, c), at which point it will notify the user. Algorithms 1 and 2 are used to calculate the value of C(a,b,c). The answer message contains directions to the parking lot as well as its address, such as P1. There's a good chance of obtaining a free spot because we propose a new parking lot based on the percentage of all available spaces.

3) Payment system: When to pay the parking fee is determined by convention or rule, which governs the payment mechanism. An advance payment method is supported by the work. We have decided to complete the payment process at the time the five-hour parking slot is reserved. The sensor will notify the server if the vehicle is left parked for longer than five hours. The driver will receive a notice from the server to pay within five hours. The next five hours must be paid for in advance by the driver.

A car initiates a request to access the parking slot. The request message will be forwarded to the server. The server is getting the information (after regular intervals) from all the neighboring parking slots. Each parking point of every parking slot (which will have the sensor) is connected to RSU as shown in Fig. 5 through the IoT network. The RSU will inform the availability of parking slots to the server through cloud communication.

The server will apply Algorithm 1 to find the neighbor table. This will give the info about the distance of the parking slot from the user and the number of available slots. Now server will apply Algorithm 2 to apply the cost function to get the cost. Finally, it will apply Algorithm 3 to recommend the best slot after cost estimating and making the payment.

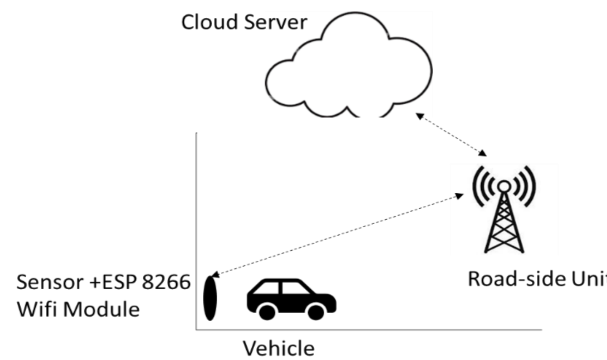

Fig. 5. Updating the current status of the slot.

V. Simulation

We replicated a network deployment complete with the previously discussed parking architecture, to assess the processes' performance. We utilized an ESP8266 Wi-Fi module and an ultrasonic sensor to transmit the current slot state. We simulated this network using the NS3 VANET emulator. We randomly generated automobiles to join the network to replicate the mathematical and queuing models. In the simulation, we used the Poisson distribution to depict the arrival of the cars at the parking lot denoted by P(X). X represents the period between consecutive arriving cars.

The simulation's X values are 15 and 20 sec. We viewed the car as the task and the parking spot as the tool used to do the task. In this simulation, the time essential to complete the task was represented by an exponential distribution, or E(Y). Here Y represents the mean service time a car spends in a parking spot. In this instance, we selected Y = 60 sec. Five parking lots served as five nodes in the simulation of a parking network. As seen in Fig. 6, we supposed that the parking slots are connected. As the resources, we configured each parking lot to hold four spots. Additionally, we generated an equal quantity of arbitrary cars to reach every parking lot. 20, 25, 30, and 70 cars arrive at each parking lot. The simulations were run repeatedly until every car was fixed.

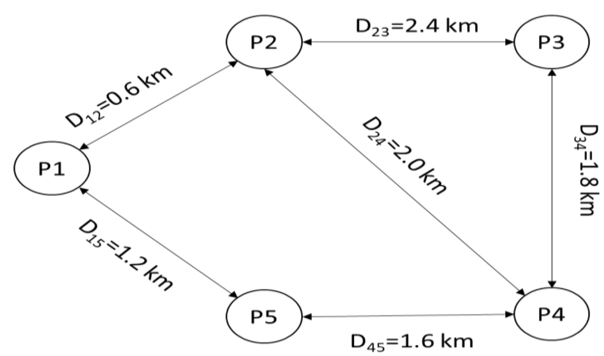

Fig. 6. A five-node network.

A simulation is created based on various a and b values to compare network performance and give the best alternative for the proposed network. All alpha and beta scenarios between zero and one were simulated. Here are a few exceptional values of a and b in the range of 0 to 1. 'a' was previously valued at {0, 0.2, 0.5, 0.8, 1}. Beta was previously valued as b = {0, 0.2, 0.5, 0.8, 1}. We have the distances between the network nodes put up D12 D15 D23 D24 D34 D45 as 0.6, 1.2, 2.4, 2.0, 1.8, and 1.6 respectively in kilometers. We choose Dup = 2.4 km as the maximum value of the distance and Tup = 4 vacancies as the top bound of the capacity. Table VIII contains a summary of all the simulation's setup parameters.

The industry and research community should recently pay attention to NS3 VANET Simulation, an emerging technology. A useful program for simulating various real-time networking systems, like parking and vehicle networking, is called NS3. Particularly in the parking system simulation, NS3 supports Poisson's distribution and exponential distribution. To assess performance, it is possible to do statistical computations and export the parameters' average values. The average automobile wait time for requests to park and the average time a car spends in the parking lot are two examples of these measures. Numerous earlier studies have demonstrated how closely the NS3 simulation findings match real-world outcomes. The above benefits led us to select NS3 as the simulation tool for this investigation.

TABLE VIII. SIMULATION PARAMETERS FOR NS3, SUMO TRAFFIC SIMULATOR

Parameter	Value
Simulation Area	1600m x 1500m
Transmission Range	350m
Model for Propagation	Nakagami (m = 1)
Model of Mobility	Gipps
Data Rate	CBR (constant bitrate)
Transport layer	TCP Lite
MAC and PHY layer	802.11p
Packet size	512 bytes
Transmission Rate	4 packets/second
Interface Queue	20 packets
Simulation Time	300 seconds
Routing Protocol	AODV
Number of vehicles arriving at each car park	{20, 25, 30, ….70}
Inter-arrival rate	P(15), P(20) sec
Service rate	60 sec
Coefficient of distance a	{0, 0.2, 0.5, 0.8, 1}
Coefficient of distance b	{0, 0.2, 0.5, 0.8, 1}
Number of runs	15 times
Confidence Interval	95%
Parking spaces	7
Vehicles	70

1) Example case: Five nodes make up the simulated network that was constructed. Two different assessment models were utilized to contrast the performance of the recommended algorithm with the existing parking system. The proposed first network model, depicted in Fig. 7(a), was created using the conventional approach; cars that arrive at a full parking lot will be parked in a line and held on to be served until no more spaces are available. The FIFO queue is this one. The classical approach is a widely used technique that illustrates the conventional parking system without any preparation to address this issue. Vehicles are looped until a parking spot becomes available at the node.

As illustrated in Fig. 7(b), the networked parking model is employed to tackle this issue and shorten the time cars in the system must wait. According to this network model, a vehicle will be sent to a different parking lot Py that has available spots when it arrives at a parking lot Px that is currently full. The algorithms have been provided to serve as the foundation for the forwarding. The simulation of two network topologies has been employed in the proposed system, using the NS3 simulator. The mean waiting time is examined for different values of parameters and the different arrival times.

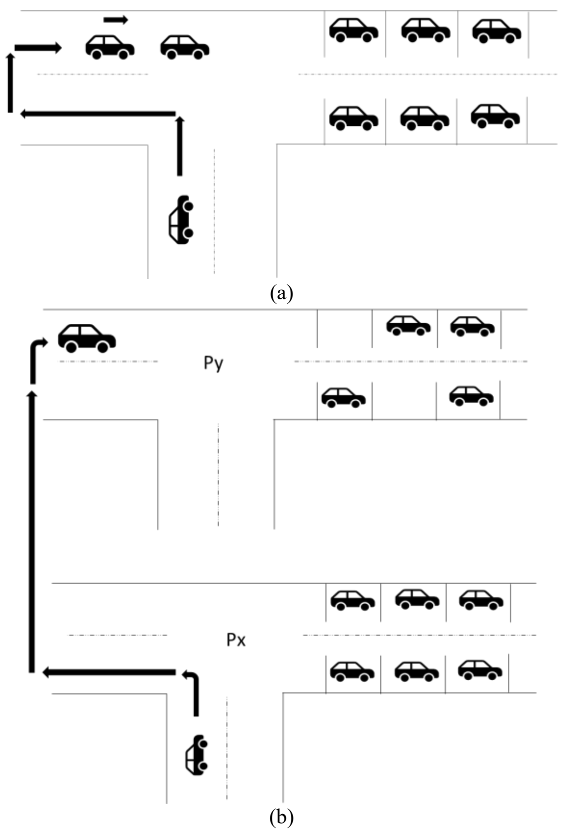

Fig. 7. (a) The Classical Model. (b) Networked model using a mechanism of forwarding.

VI. RESULT ANALYSIS

To evaluate the efficacy of the recommended system, firstly the system's recommendation for car parking is considered based on coefficients and the values that the parking slot collected in line with Tables I to VII. Here, we analyze the advice using just one parking space, P2. P2, P1, P3, and P4 are its neighbors. Ten open spots are maintained in each parking space to compare the suggestions. According to Tables I to VII, P4 is advised for C(0.2, 1, 0.8), P1 and P3 are advised for C(0.5, 1, 0.5), and P3 is advised for C(0.8, 1, 0.2). Similarly, as the value of coefficients a, b, and c are changed we get different recommendations in each case as shown in Fig. 8 (a) to (i).

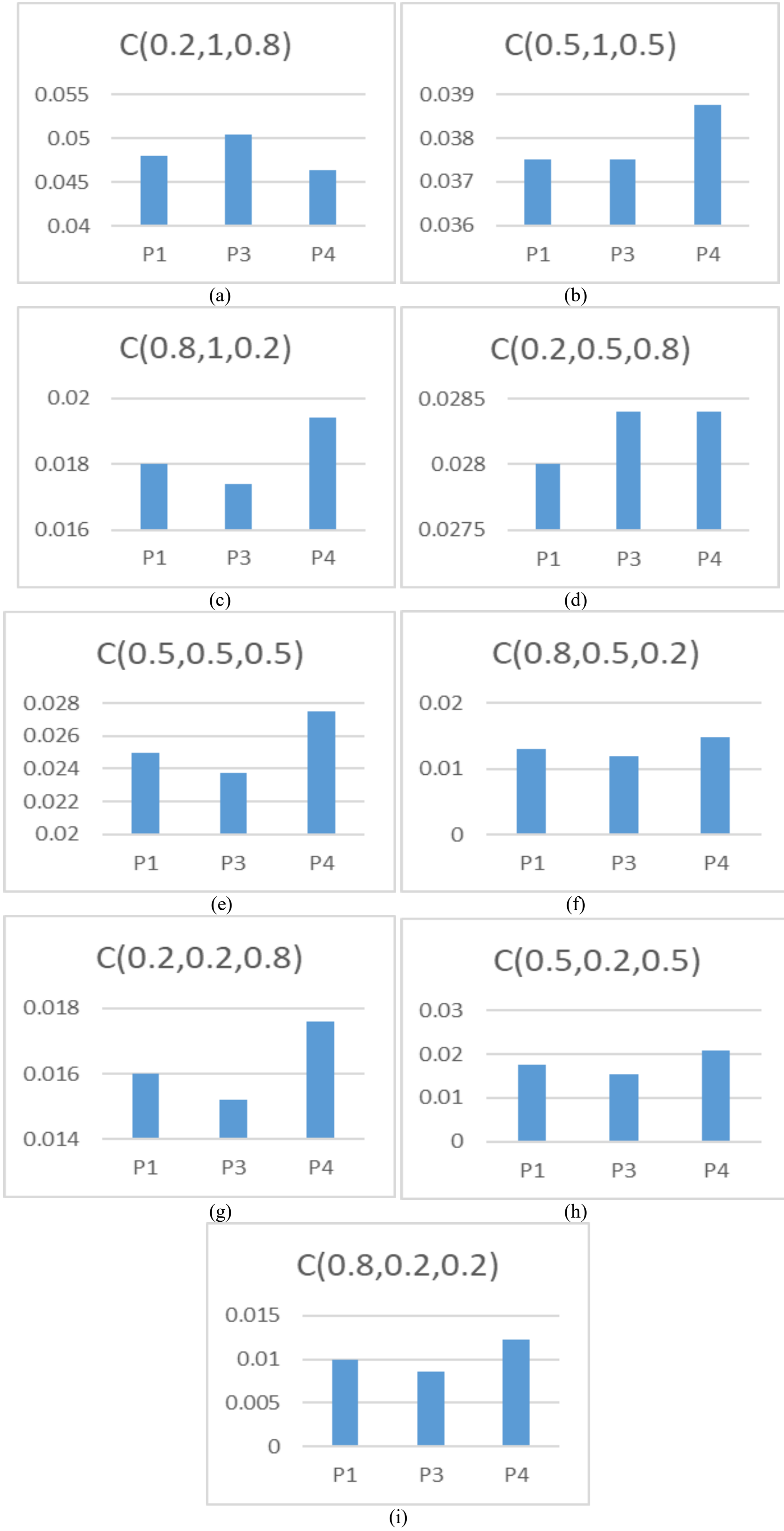

Fig. 8. (a) to (i): The recommendations on varying the values of a, b and c.

We have now shown that the cost of driver time spent in the parking system is the performance metric. The time a driver spends using the parking system to receive service is the cost to the user. Expenses like money, gasoline, and pollution can be cut by minimizing these costs. The study's time measurement is the average wait time for the user's service and the average time the user spends using the parking system with travel, waiting, and service periods. Improved system performance is the result of a lower cost value.

The parameters along with the lowest cost value, given the parameters we simulated, will be regarded as the best option and are used as a suggestion to implement a model similar to this one in real life. We have compared the average waiting time for the suggested network and a classical network with a loop is shown in Fig. 7. We used 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, and 70 automobiles that arrived at each node in the experiment. According to the distribution of the inter-arrival times, P = 15 and P = 20 seconds, as shown in Fig. 9 and Fig. 10 respectively, four cars arrive in each parking lot per minute.

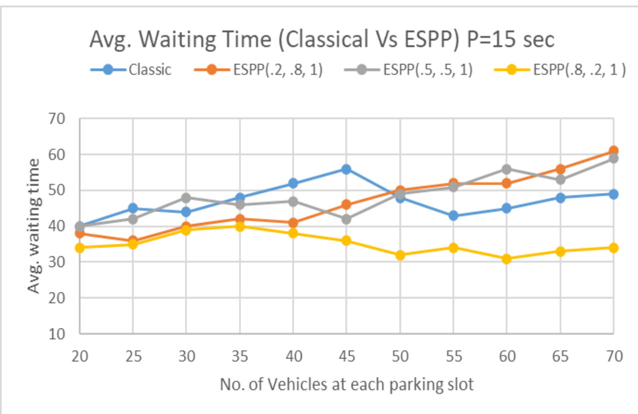

Fig. 9. Average waiting time at P = 15 sec.

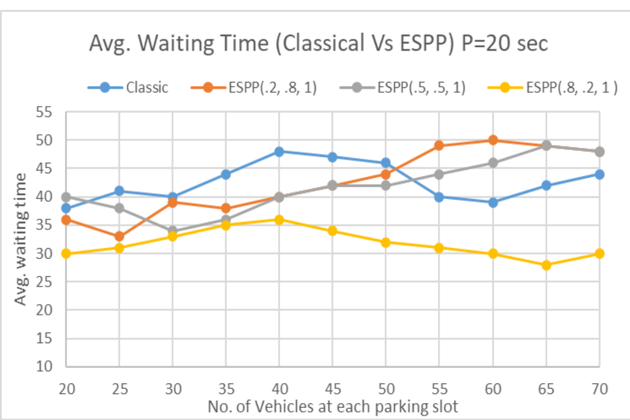

Fig. 10. Average total time at P = 20 sec.

The outcomes demonstrate that the method outperforms the network model with performance even without planning. The optimal performance of the suggested network is achieved with the least waiting time when a and b have values of 0.8 and 0.2, respectively. In the worst-case situation, when a = 1 and b = 0, the network we suggest has the greatest average waiting time since we used only the travel distance parameter for calculating C(a,b,c). Because the proportion of available parking spots is not considered, there is a good chance that the user will still not find a vacant slot at the next car park if they are merely sent to the one with the shortest distance. In this instance, the network performance is not comparable to that of a typical network. We recognize that using the percentage of available spots in every parking lot as a criterion for allocating user forwarding would significantly reduce the user's waiting time for the service compared to a standard network.

The anticipated inter-arrival time in this instance is 20 seconds, is greater than 15 seconds. Therefore, Fig. 9 and Fig. 10 illustrate that the mean wait time will be greatly lowered (about 10 folds for the greatest situation where a is 0.2 and b is 0.8). The simple explanation is that each node has a fixed total quantity of parking spaces, so the more cars that join the network every hour, the greater the wait time for service.

We have simulated the design for seven parking spaces and 70 vehicles. It can further be scaled in a real-world environment. We believe that there will not be any issue in a larger setup.

VII. Conclusion

The parking strategy suggested by this study reduces the number of users unable to locate a spot and lowers the expense of shifting to one. The planned architecture and system in an actual scenario have been successfully simulated and executed. The findings demonstrate how much the proposed algorithm shortens users' typical wait times for parking. The research findings closely match the predictions of the mathematical models proposed. When the majority of vehicles could locate a parking place without difficulty, the proposed system's simulation reached the best possible outcome. The average wait time for services in each parking lot and the overall amount of time spent by each automobile in each lot are decreased.

As a future extension further research on the security elements of the proposed system would be considered and put the suggested system into large-scale real-world implementation.

Conflict of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

Acknowledgment

The authors extend their appreciation to the Deanship of Scientific Research at Saudi Electronic University for funding this research (9484).

References

[1] 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, 2019.

[2] N. P. Rocha, A. Dias, G. Santinha, M. Rodrigues, A. Queiros et al., “Smart cities and public health: a systematic review,” Procedia Comput. Sci., vol. 164, pp. 516–523, 2019.

[3] M. N. Kamel Boulos, N. M. Al-Shorbaji, “On the internet of things, smart cities and the WHO healthy cities,” Int. J. Health Geogr., vol. 13, no. 1, 2014.

[4] Y. Rahayu and F. N. Mustapa, “A secure parking reservation system using GSM technology,” Int. J. Comput. Commun. Eng., pp. 518–520, 2013.

[5] Masmoudi, A. Wali, A. Jamoussi and A.M. Alimi, "Vision based system for vacant parking lot detection: VPLD," in: VISAPP 2014- Proceedings of the 9th International Conference on Computer Vision Theory and Applications, vol. 2, pp. 526–533, 2014.

[6] A. Yadav and M. Arif, "An approach to smart parking algorithm using ant colony optimization and decision tree algorithm," International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 12, no. 10, pp. 1-6, 2021.

[7] M. Y. I. Idris, E. M. Tamil, N. M. Noor, Z. Razak and K. W. Fong, "Parking guidance system utilizing wireless sensor network and ultrasonic sensor," Inf. Technol. J., vol. 8, no. 2, pp. 138–146, 2009.

[8] M. Arif and S. Ahmad, "Security issues in vehicular adhoc network: a critical survey," in: Proceeding of Intelligent Communication, Control and Devices 2017 (ICCD 2017), pp. 27-29, pp. 527-536, 2017.

[9] Z. Chen, J. C. Xia and B. Irawan, "Development of fuzzy logic forecast models for location-based parking finding services," Math. Probl Eng., vol. 2013, 2013.

[10] Y. Geng and C. G. Cassandras, "A new 'smart parking' system infrastructure and implementation," Procedia- Soc. Behav. Sci., vol. 54, pp. 1278–1287, 2012.

[11] M. Sweet, "Traffic congestion's economic impacts: evidence from US metropolitan regions," Urban Stud., vol. 51, no. 10, pp. 2088–2110, 2014.

[12] G. Fontaras, N.G. Zacharof and B. Ciuffo, "Fuel consumption and CO2 emissions from passenger cars in Europe– laboratory versus real-world emissions," Prog. Energy Combust. Sci., vol. 60, pp. 97–131, 2017.

[13] K. Zhang and S. Batterman, "Air pollution and health risks due to vehicle traffic," Sci. Total Environ., vol. 450, no. 451, pp. 307–316, 2013.

[14] N. Zhong, J. Cao and Y. Wang, "Traffic congestion, ambient air pollution, and health: evidence from driving restrictions in Beijing," J. Assoc. Environ. Resour. Econ., vol. 4, no. 3, pp. 821–856, 2017.

[15] J. I. Levy, J. J. Buonocore and K. V. Stackelberg, "Evaluation of the public health impacts of traffic congestion: a health risk assessment," Environ. Heal. A Glob. Access Sci. Source, vol. 9, no. 1, p 65, 2010.

[16] Infrastructure Australia, Urban Transport Crowding and Congestion the Australian Infrastructure Audit 2019 Supplementary Report, 2019.

[17] K. Hassoune, W. Dachry, F. Moutaouakkil and H. Medromi, 'Smart Parking Systems: A Survey," Dec. 2016.

[18] S. Nene, S. Mundle, S. Mahajan, S. Yeginwar and L. Panchal, "A study of vehicular parking systems," in: ICICCT 2019– System Reliability, Quality Control, Safety, Maintenance and Management, Springer Singapore, pp. 207–215, 2020.

[19] V. Paidi, H. Fleyeh, J. Håkansson and R.G. Nyberg, "Smart parking sensors, technologies and applications for open parking lots: a review," IET Intell. Transp. Syst., vol. 12, no. 8, pp. 735–741, 2018.

[20] A. Hilmani, A. Maizate and L. Hassouni, "Designing and managing a smart parking system using wireless sensor networks," J. Sens. Actuator Netw., vol. 7, no. 2, p 24, 2018.

[21] 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. Front, vol. 22, no. 4, pp. 1–22, 2020.

[22] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini et al., "Deep learning for decentralized parking lot occupancy detection," Expert Syst. Appl., vol. 72, pp. 327–334, 2017.

[23] Z. Suryady, G. R. Sinniah, S. Haseeb, M. T. Siddique and M. F. M. Ezani, "Rapid development of smart parking system with cloud-based platforms," Jan. 2014.

[24] S. Y. Chou, S. W. Lin, C. C. Li, "Dynamic parking negotiation and guidance using an agent-based platform," Expert Syst. Appl., vol. 35, no. 3, pp. 805–817, 2008.

[25] M. E. Karbab, D. Djenouri, S. Boulkaboul and A. Bagula, "Car park management with networked wireless sensors and active RFID," in: IEEE International Conference on Electro Information Technology, 2015-June, pp. 373–378, 2015.

[26] D. Hunusekattte, "Smart Parking With Automated Billing System," 2020 15th International Conference for Internet Technology and Secured Transactions (ICITST), London, United Kingdom, 2020, pp. 1-5, doi: 10.23919/ICITST51030.2020.9351312.

[27] Sandeep Saharan, Neeraj Kumar, Seema Bawa, An efficient smart parking pricing system for smart city environment: A machine-learning based approach, Future Generation Computer Systems, Volume 106, 2020, https://doi.org/10.1016/j.future.2020.01.031.

[28] Rocco G, Pipino C, Pagano C. An Overview of Urban Mobility: Revolutionizing with Innovative Smart Parking Systems. Sustainability. 2023; 15(17):13174. https://doi.org/10.3390/su151713174 IoT-Based Smart Parking System: A Comprehensive Development Guide https://www.rishabhsoft.com/blog/iot-based-smart-parking-system-development.