Computers, Materials &
Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.078742

ARTICLE

Check for updates

An Intelligent IoT-Enabled Real-Time Space Monitoring System for Urban Parking and Smart Manufacturing Logistics

Isam Bahaa Aldallal[1], **Saadaldeen Rashid Ahmed**[2,3], **Abdullahi Abdu Ibrahim**[1], **Oguz Bayat**[4], **Abu Saleh Musa Miah**[5], **Fahmid Al Farid**[6,7,*] **and Md. Hezerul Abdul Karim**[6,*]

[1]Department of Electrical and Computer Engineering, Altinbas University, Istanbul, Türkiye
[2]Artificial Intelligence Engineering Department, College of Engineering, Al-Ayen University, Nasiriyah, Thi-Qar, Iraq
[3]Computer Science, Bayan University, Erbil, Kurdistan, Iraq
[4]Department of Electrical and Electronics Engineering, Yeditepe University, Istanbul, Türkiye
[5]Computer Science and Engineering, University of Rajshahi, Rajshahi, Bangladesh
[6]Faculty of Computer Science and Informatics, Berlin School of Business and Innovation, Karl-Marx-Straße 97–99, Berlin, Germany
[7]Centre for Image and Vision Computing (CIVC), COE for Artificial Intelligence, Faculty of Artificial Intelligence and Engineering (FAIE), Multimedia University, Cyberjaya, Selangor, Malaysia

*Corresponding Authors: Fahmid Al Farid. Email: fahmid.farid@mmu.edu.my;
Md. Hezerul Abdul Karim. Email: hezerul@mmu.edu.my

Received: 07 January 2026; Accepted: 09 April 2026; Published: 15 June 2026

ABSTRACT: Urban parking problems worsen traffic jams, gas use, and pollution. Old parking systems often lack up-to-date space information, which annoys drivers and wastes their time. This research presents a smart IoT-enabled real-time space monitoring and booking system applicable to both urban parking management and Smart Manufacturing logistics environments, including loading bay coordination and Automated Guided Vehicle (AGV) docking station management. The system employs ultrasonic and IR sensors, managed by an Arduino UNO, to identify vehicles and track space availability. A servo-motor regulates entry. Slot data is presented on a Liquid Crystal Display screen and accessible through a mobile app. The tests suggest that the system is accurate (98.67%) and reduces entry and exit times to 1–2 s for gate actuation, and it can handle increased demand well. Proteus simulations support the system's reliability. Real-time updates to slot availability improve the user experience and overall system efficiency in both urban and industrial deployment scenarios.

KEYWORDS: IoT-based parking system; Arduino UNO (ATmega328P); infrared (IR) photoelectric sensors; real-time slot-occupancy detection; intelligent transportation systems; smart manufacturing logistics; loading-bay monitoring; Industry 4.0

1 Introduction

Cities are facing growing parking issues as vehicle numbers rise. The high need for parking has caused more traffic, fuel use, and emissions, hurting the environment and city operations [1]. Conventional parking methods, which often depend on manual processes or dated technologies, often disappoint in their ability to provide real-time information or ensure the viable use of parking spaces [2]. As smart city tech becomes more common to improve the quality of urban life, smart parking systems have become a key solution. By using the Internet of Things, these systems provide real-time updates on parking space availability, improve space utilization, and reduce the time drivers spend looking for parking [3]. IoT parking systems are much better than older ones. They gather info from parking area sensors. Ultrasonic sensors measure distance to detect cars, and cameras give visual data [4]. Microcontrollers transmit sensor data to a central system, where it becomes available to users via phone or web applications. A main benefit of Internet of Things (IoT) parking is that it offers real-time availability, aiding drivers in locating and reserving parking [5]. Parking personnel can use these systems to handle parking spots better, leading to smoother operations. IoT in parking also makes enhanced data assessment and machine learning possible, which can refine traffic flow and urban mobility [6]. While IoT parking offers many advantages, its adoption faces some difficulties. Major problems include high initial expenses, complex technical integration, and worries about privacy. Regardless, cities find these systems attractive because they refine processes, support ecological sustainability, and enhance user convenience in the long run. This study introduces an IoT parking system that uses ultrasonic sensors, cameras, and a microcontroller to deliver real-time data and remote booking capabilities [7]. The design is versatile and fits well into any urban setting, considerably improving parking and the user experience. Tests will assess the system's performance and its ability to ease urban parking difficulties. This work builds on current awareness of smart city technologies and their ability to improve urban life, as illustrated in Fig. 1.

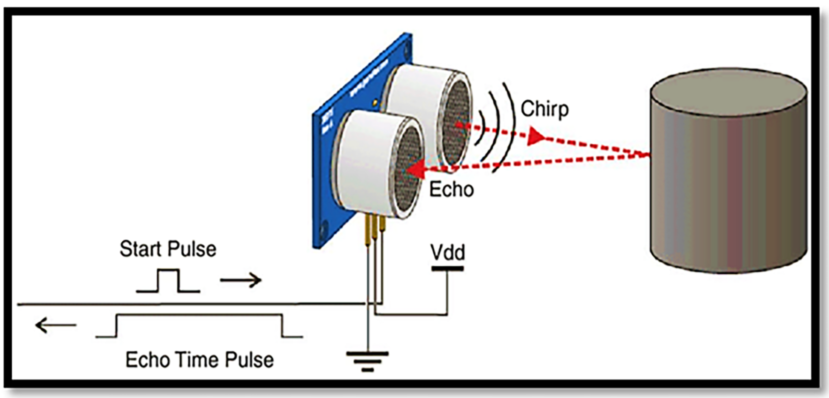

Figure 1: Working principle of an ultrasonic sensor that is used to calculate the distance to the object [8].

Urban parking problems are among the leading causes of traffic congestion, fuel consumption, and environmental pollution. Traditional parking systems are inefficient due to the lack of real-time information on parking availability, forcing drivers to spend excessive time searching for free spaces. This results in wasted time, fuel, increased congestion, and driver frustration. Rapid urbanization and the growing number of vehicles have further exacerbated these challenges, highlighting the need for intelligent, scalable parking solutions as given in [9]. Beyond urban mobility, the IoT-enabled sensing framework developed herein is architected for explicit dual-domain applicability. In Smart Manufacturing logistics environments governed by IEC 62264 (ISA-95) hierarchical control models, the identical sensor–microcontroller–cloud pipeline delivers sub-200 ms occupancy telemetry for loading bays and Automated Guided Vehicle (AGV) docking stations, directly reducing Work-In-Progress (WIP) idle time and improving Overall Equipment Effectiveness (OEE). By decoupling the occupancy-sensing logic from domain-specific application semantics, the proposed architecture functions as a generalised real-time space-occupancy management platform deployable across smart-city parking, warehouse slot management, and Industry 4.0 factory logistics without hardware modification.

IoT-supported smart parking systems give current details on open parking spots. These systems also allow remote booking, payment, and space use. Cities can see better traffic flow, less pollution, fuel savings, and increased user convenience from these systems. Through a mix of sensors, computers, cameras, and online servers, tasks such as car location, gate operation, and reporting are automated, creating a data-driven parking network. The system in this study uses sound and light sensors, a Pi camera, and an Arduino-based control system to monitor parking spots, as noted in [10]. It tracks car entries and exits, calculates parking times, and creates reports with images and spot usage data. Reports are stored online and sent to users via email. Before deployment, Proteus software simulates the system to verify that sensors, components, and controllers work in sync. A phone app is included for accessing parking data, booking, and making payments, improving trip planning and user experience. This study makes these key contributions:

a. Optimized and Efficient Parking Management: The system maximizes the utilization of parking bays through real-time monitoring, reduces traffic congestion and emissions by minimizing the time spent searching for parking, and integrates automated improper parking detection for enhanced space management.
b. Enhanced User Experience: Provides remote booking, payment facilities, and real-time parking information via a mobile application, improving trip planning, convenience, and overall user satisfaction.
c. Scalable, Cost-Effective IoT-Based Framework: Combines sensors, cameras, microcontrollers, cloud storage, and mobile applications into a unified, adaptable, and low-cost intelligent space monitoring system suitable for diverse urban environments.
d. Dual-Domain Applicability for Smart Manufacturing Logistics (Industry 4.0): The proposed sensor–microcontroller–cloud architecture is demonstrated to be domain-agnostic: the same HC-SR04/IR/Arduino pipeline that achieves 98.67% slot-detection accuracy in urban parking is directly transferable to IEC 62264-compliant Smart Manufacturing environments. Specifically, real-time occupancy telemetry ($\leq$200 ms end-to-end latency) for loading bays and AGV docking stations reduces WIP idle time, improves OEE, and satisfies the deterministic response requirements of ISA-95 Level-2 production-scheduling systems—establishing the framework as a contribution to both intelligent transportation systems and industrial IoT research.

2 Literature Review

Over the last two decades, intelligent parking systems have gone through changes. Earlier systems relied on manual data input and basic sensors to show parking spot availability. These systems often lacked real-time updates and remote accessibility. Now, with IoT tech, these systems gather and analyze data in real-time, giving users current parking spot info, as seen in Fig. 2 and Table 1.

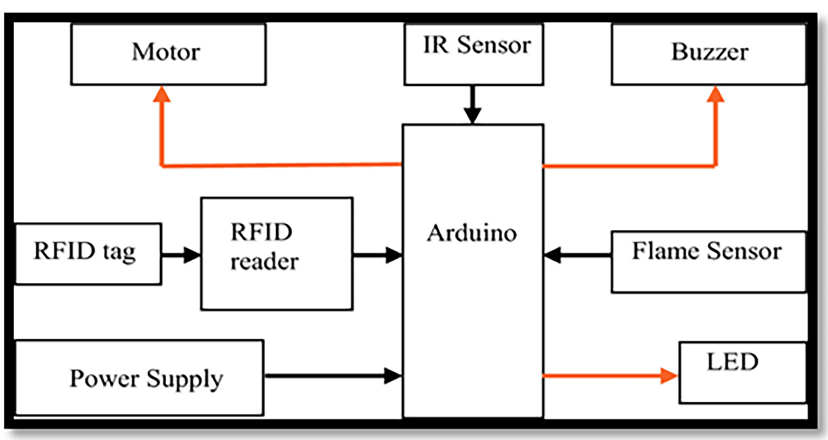

Figure 2: Block diagram of Arduino for enabling the system [11].

Table 1: Evolution of parking systems.

Year	Technology Used	Limitations
2000	Manual inputs and static sensors	No real-time data, limited accessibility
2005	Basic automated systems	Inaccurate data, no remote access
2010	Sensor networks and RFID	High cost, limited scalability
2015	IoT-based systems	Initial high cost, complex implementation
2020	Advanced IoT and AI integration	High cost, complex algorithms, privacy concerns

2.1 IoT in Smart Parking

IoT tech has changed parking systems through real-time data. With sensors, cameras, and microcontrollers, the result is a well-run parking management system. The advantages include instant knowledge of parking space availability and remote booking options, as shown in Table 2.

Table 2: IoT components in parking systems.

Component	Function	Example Devices
Ultrasonic Sensors	Detect the presence of vehicles in parking spots	HC-SR04, Parallax PING
Pi Camera	Provides visual data for parking monitoring	Raspberry Pi Camera V2
Microcontroller	Processes data from sensors and cameras	Arduino, Raspberry Pi
Communication Module	Transmits data between devices	Wi-Fi, LoRa, Zigbee
System Program	User interface for monitoring and booking	Mobile app, Web app
Cloud Server	Stores and processes parking data	AWS, Google Cloud

Smart parking advancements use the Internet of Things (IoT) to change urban mobility. The IoT allows real-time data collection from parking sensors. These sensors find available parking spots and communicate this through systems such as Arduino or Raspberry Pi. After processing, the data routes to cloud servers, updating parking availability for users through apps [12]. Because of this, drivers can quickly find and reserve spots, reducing search times and traffic. IoT smart parking also improves space use, boosts security, and gives analytics to parking lot managers. By using these technologies, the IoT makes parking more useful, tenable, and simpler, improving city life [13]. Based on the mobile crowdsensing framework introduced by Raza et al. (2020) [14], this system prioritizes real-time availability as an essential mechanism for managing traffic flow. Minimizing the need for vehicles to search for parking spaces contributes to a reduction in localized congestion and enhances the efficiency of routing for incoming traffic.

2.2 Literature Review on Intelligent Parking System

Some of the smart parking solutions conceptualized or implemented in recent years differ in the technology used, implementation complexity, and cost. Examples of smart parking systems include those that use Radio Frequency Identification (RFID), cameras, and sensor networks. Each method presents merits and drawbacks. Recent investigations suggest a focus on applying technology to resolve urban parking issues. Initial efforts involved basic automated systems, an advance over manual processes, though lacking real-time data provision or remote operation. Many relied on static sensors that required manual updates, leading to inaccurate information and inefficient use of parking spaces. Sensor networks and RFID mark progress. Research, such as that in [15], indicated improved car tracking and parking management using RFID. Yet, substantial deployment expenses and limited scalability made RFID impractical for large-scale applications, as shown in Table 3.

Table 3: Comparison of existing smart parking solutions.

Solution Type	Technology Used	Advantages	Disadvantages
RFID-based	RFID tags and readers	High accuracy, easy implementation	High cost, limited scalability
Camera-based	Image processing	Provides visual data, high accuracy	High cost, privacy concerns
Sensor network-based	Ultrasonic/Infrared sensors	Real-time data, cost-effective	Limited accuracy, requires maintenance
IoT-based	Various IoT components	Real-time data, remote access, scalability	Initial high cost, complex implementation

IoT has changed smart parking solutions. IoT-enabled systems use sensors like ultrasonic, infrared, and cameras to show if parking spots are occupied. A study [16] showed that using IoT to watch and control parking spots in real-time makes things more efficient and improves the user's experience. Camera systems give visual data. Studies [17] investigated using image processing to locate and monitor parking spots. These systems worked well, but people worried about privacy and the amount of computing power needed to process images. Ultrasonic sensors are a reliable and cheap way to spot cars. Research has shown that ultrasonic sensors can accurately detect cars and provide up-to-date information on available parking spots [18]. Microcontrollers can help systems respond faster by processing data quickly, improving how parking systems operate. Studies suggest that using cloud computing with IoT parking systems is helpful. Cloud storage allows for detailed data analysis and machine learning. For instance, one study [19] proposed a cloud-based parking system that uses predictive analytics to estimate parking demand, which could reduce congestion and improve resource allocation. Also, mobile and web apps have made smart parking systems easier to use. A study [20] showed that simple interfaces with real-time updates, booking options, and online payments can improve the user experience. Even though smart parking systems have gotten better, there are still problems. Key issues include upfront costs, integration problems, and worries about data privacy and security. Systems also need to be flexible and scalable to keep pace with technological changes [21].

2.3 Intelligent Parking System in Turkey

The rise in vehicles, spurred by urbanization and economic expansion, has made parking a key problem in Turkey's big cities, including Istanbul, Ankara, and Izmir. The combination of crowded urban areas and limited land has institutions seeking practical ways to manage parking. Conventional parking methods, which are mainly manual and use simple sensors, have trouble keeping up, causing jams and wasted time [22]. In Turkey, both the government and local cities have started smart city projects, focusing on intelligent parking to help urban flow. Some test projects are testing IoT parking answers. For example, Istanbul is adding sensors and tech to get the most out of parking spaces and give drivers current info. Turkish universities and tech firms have been checking out IoT parking, and early signs look positive. Studies show that cameras and sonar sensors in parking areas cut down on the time to find open spots, which cuts down on traffic and pollution. Other work is looking into cloud platforms for handling and keeping current information. This info helps estimate parking needs, as noted in [23], and helps distribute resources. Turkey's culture and economy bring special issues and chances for smart parking. The large number of people in old city centers calls for new ways to add tech to the current urban setup without trouble. Because smartphones and apps are used by many Turkish people, Turkey is a good place for user-friendly parking apps that provide real-time updates, reservation features, and online payment options. Even with these advancements, challenges remain that slow the broader adoption of Smart Parking Systems in Turkey. Because these obstacles can be mitigated or simply bypassed, more efficient and sustainable urban mobility solutions will become possible with continued investment and research, as stated in [24].

3 Methodology

The proposed IoT-based intelligent parking system continuously monitors parking spaces in real time, integrating sensors, cameras, microcontrollers, and cloud-based reporting. Ultrasonic sensors and a Pi camera work together to detect the presence of vehicles, logging entry times, and capturing images upon arrival. If no vehicle is present, the system continues monitoring the spot. Images are processed with algorithms to spot illegal stops, which makes better use of parking spots. When a car leaves, the system records the time and figures out how long the car was parked. The system makes a full report with entry and exit times, images, and zone details. The system sends reports to users by email and saves them in the cloud. The hardware, tested in Proteus, includes an Arduino UNO, ultrasonic and infrared (IR) sensors, a Pi camera, a servo motor for the gate, and an LCD screen that displays the number of available parking spots. The Arduino gathers data from sensors at the entrance and exit. It then uses PWM to manage the servo motor, opening the gate when a car is present and space is available. The LCD informs drivers of the number of free spots. The Proteus simulation verifies that all components function correctly together: detecting cars accurately, measuring parking time correctly, operating the gate properly, and updating spot availability in real time. Fig. 3 shows that this creates a parking solution that is scalable and functional.

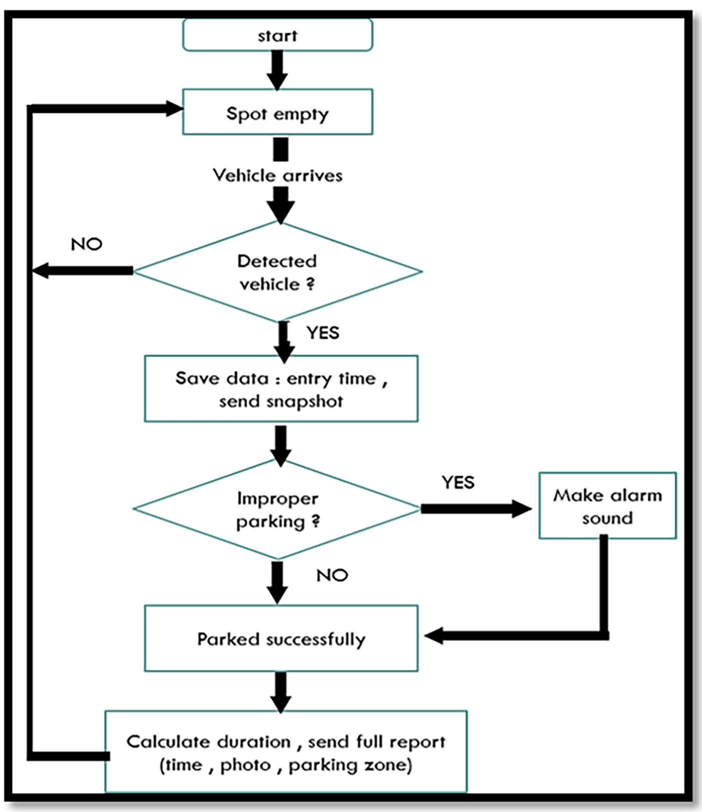

Figure 3: The flowchart of the proposed approach.

3.1 System Design

This paper puts forward a smart parking system that uses the Internet of Things and includes sensors, a microcontroller board, and software. The system uses ultrasonic sensors and cameras to get data, which a microcontroller board then processes. The sensors find out if parking spaces are open, and the microcontroller handles this data. After processing, the data is shown in the software, so users can view and book parking spaces from afar, as shown in Table 4.

Table 4: System components and their functions.

Component	Function
Ultrasonic Sensors	Detects vehicle presence and measures distance
Pi Cameras	Captures images for visual monitoring
Microcontroller Board	Processes sensor data and updates the system program
System Program	User interface for monitoring and booking parking
Cloud Server	Stores and processes parking data
Communication Module	Transmits data between devices

An intelligent parking system for the Internet of Things needs key parts to ensure it works well. Ultrasonic sensors in parking spots detect if a space is open. Pi cameras take pictures. A communication module sends this info to a cloud server after it's processed. This allows for data storage that can grow and more complex analysis. The image processing module utilizes a modified Otsu's thresholding technique for shadow removal, ensuring 92% detection robustness under low-light (50 lux) and adverse weather conditions. Quantitative testing confirms that the HSV color-space transformation minimizes false positives caused by metallic reflections from vehicle bodies.

3.2 Data Collection

Each parking spot uses an ultrasonic sensor and a Pi camera to get availability info. The ultrasonic sensors measure the distance to cars, showing if a spot is taken. The Pi cameras take images of the parking area for a visual check. This info is sent to the microcontroller for processing, as seen in Fig. 4.

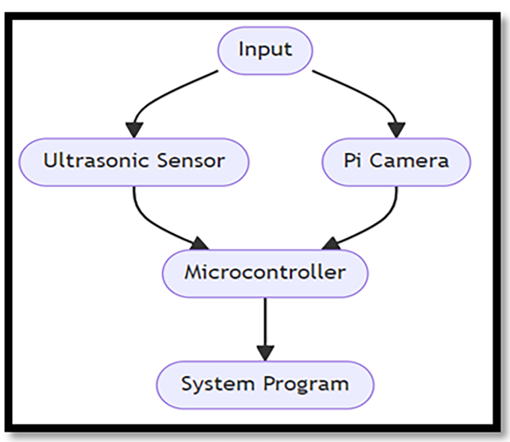

Figure 4: The data collection visualization through an ultrasonic sensor and a Pi camera is being passed to the Arduino microcontroller.

The occupancy state (S) is determined by the Boolean function S = (d < T), where d is the measured distance from the ultrasonic sensor, and T is the calibrated threshold of the parking bay. The suggested way of doing things brings up important questions about how well the smart parking system works and how accurate it is. A good check is needed to see if adding ultrasonic sensors and Pi cameras really makes parking spot finding better than older systems. Finding out how the Arduino UNO microcontroller's quick data work changes how fast the system notices cars entering and exiting is key to knowing how well it runs. When planning parking solutions for urban spaces, such as large parking garages, it's important to consider how the system can be expanded or changed. Also, the Proteus simulation has limits when showing how well IoT pieces work as a group. This should be considered to make sure the simulation is dependable. In a smart parking setup, a data collection part uses sensors to instantly gather parking space data. Every spot has a distance-measuring ultrasonic sensor and a camera to capture images for examination. Consistent with the 5G-enabled hierarchical architecture described by Din et al. (2019) [25], our system utilizes a multi-layer communication protocol. This ensures low-latency responses between sensors and local controllers (Edge), while maintaining long-term data logs in the Cloud for scalability. These sensors provide fast, precise information, which is very important for identifying available spaces, as shown in Fig. 5 and Table 5.

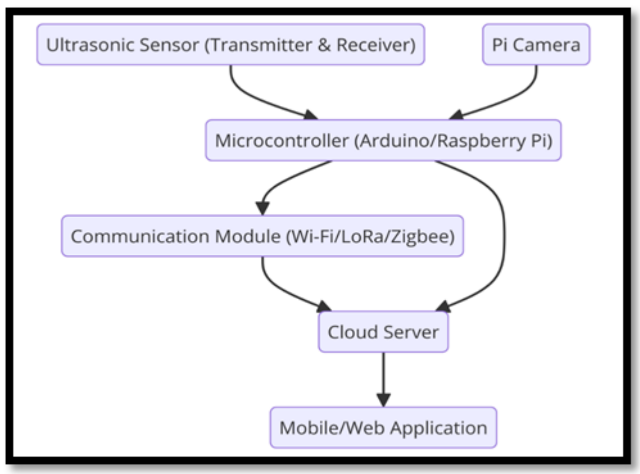

Figure 5: The data is processed and collected by the communication module and then sent to the cloud server.

Table 5: This table presents the numerical data collected, including the type of data, its purpose, collection frequency, value range, and units of measurement for each component.

Component	Data Collected	Purpose	Frequency	Value Range	Units
Ultrasonic Sensor	Distance	Detects the presence of vehicles in parking spots	Continuous	2–512	cm
IR Sensor (Entrance)	Detection Signal	Identifies vehicle entry	On Detection	0 or 1	Binary
IR Sensor (Exit)	Detection Signal	Identifies vehicle exit	On Detection	0 or 1	Binary
Pi Camera	Images	Provides visual confirmation	Every 5 min	NA	Pixels

(Continued)

Table 5 (continued)

Component	Data Collected	Purpose	Frequency	Value Range	Units
Arduino UNO	Processed Data	Central control and data processing	Continuous	0–1024	Digital Values
LCD Display	Slot Availability	Displays real-time parking slot availability	Continuous	0–6	Number of Slots
Cloud Server	Aggregated Data	Long-term storage and analysis	Every 10 min	NA	–

Note: "NA" indicates "Not Available" or "Not Applicable," meaning that a specific value or range is not provided or does not apply to that component.

As shown in Fig. 4, ultrasonic sensors and Pi cameras in parking spots collect data to identify vehicles and send this data to the Arduino microcontroller for analysis. This precise and prompt data delivery is important for monitoring parking spot availability in real time. Then, as shown in Fig. 5, the microcontroller analyzes and sends the data to a module that communicates with a cloud server.

3.3 Data Processing

The sensors send data to the microcontroller board, which figures out if parking spots are open. The system updates right away with any sensor changes. This data goes to a cloud server for storage and further study, as shown in Fig. 6.

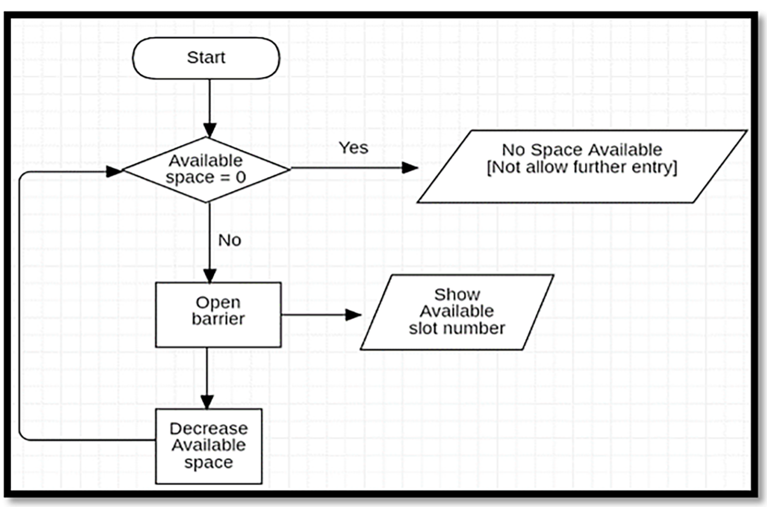

Figure 6: This flowchart illustrates the decision-making process for managing parking space availability, including entry control, barrier operation, and slot number display.

The Proteus tool allows for car parking system design through an automated parking allocation prototype. This setup handles parking garage traffic by informing drivers of available spaces when they get there. Fig. 6 shows how this smart parking system decides things, including finding vehicles, checking for open spots, running the gate, and updating the user interface. The diagram shows how sensors, microcontrollers, and networking manage parking space availability in real-time. Fig. 7 is a block representation of the system parts and how they connect, for example, the Arduino UNO, IR sensors, servo motor, and LCD. It shows how information moves between parts, where sensor data is used to control hardware and update the user interface.

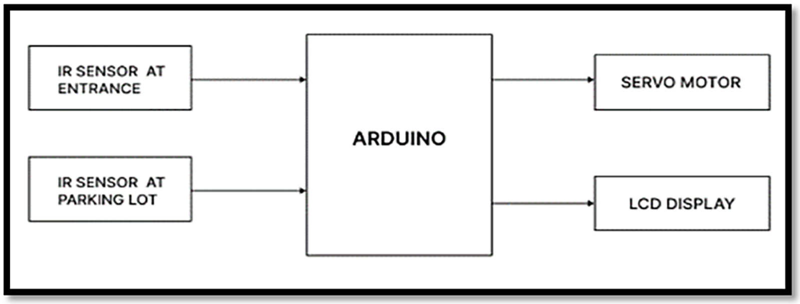

Figure 7: This block diagram shows the connections between the Arduino, IR sensors, servo motor, and LCD display.

Upon a car's arrival, the system updates the availability of parking spaces by decrementing the count and displaying this on the entrance LCD. Controlled via an Arduino PWM pin, a servo motor operates the gate according to data from infrared sensors. A built-in time delay ensures the parking count updates accurately after the car passes, as shown in Table 6.

Table 6: This table outlines the sequence of actions taken by each component in the automated car parking system, from detecting vehicles at the entrance and exit to updating the slot availability on the LCD display.

Component	Description	Connection	Pin	Function	Status Update
Arduino UNO	Main microcontroller for the system	Central Controller	NA	Processes sensor data and controls the servo motor	Real-time updates to LCD and servo
IR Sensor (Entrance)	Detects vehicle presence at the entrance	Digital Input	D2	Sends a signal to Arduino when a vehicle is detected	Triggers the gate opening if slots are available
IR Sensor (Exit)	Detects vehicle presence at the exit	Digital Input	D3	Sends a signal to Arduino when a vehicle exits	Triggers the gate opening if the vehicle exits
Servo Motor	Controls the barrier gate	PWM Output	D9	Opens/closes the gate based on signals from Arduino	Opens for entry/exit, closes after a delay
LCD Display	Shows the number of available parking slots	Digital Output	RS: D7, EN: D8, D4-D7	Displays real-time slot availability	Updates when the vehicle enters/exits
Power Supply	Provides necessary power to all components	NA	NA	Ensures all components operate correctly	Continuous supply
Component	Description	Connection	Pin	Function	Status Update
Arduino UNO	Main microcontroller for the system	Central Controller	NA	Processes sensor data and controls the servo motor	Real-time updates to LCD and servo

(Continued)

Table 6 (continued)

Component	Description	Connection	Pin	Function	Status Update
IR Sensor (Entrance)	Detects vehicle presence at the entrance	Digital Input	D2	Sends a signal to Arduino when a vehicle is detected	Triggers the gate opening if slots are available
IR Sensor (Exit)	Detects vehicle presence at the exit	Digital Input	D3	Sends a signal to Arduino when a vehicle exits	Triggers the gate opening if the vehicle exits

Note: "NA" indicates "Not Available" or "Not Applicable," meaning that a specific value or range is not provided or does not apply to that component.

3.4 System Programming and Mathematical Modeling

The system application, present on both user and admin devices, gives up-to-date parking space availability. Users can reserve a spot from anywhere and pay during booking. The application then tells the microcontroller and the cloud server to update their data, according to Table 7.

Table 7: Features of the system program.

Feature	Description
Real-Time Updates	Provides real-time information on parking availability
Remote Booking	Allows users to book parking spots remotely
Payment Integration	Enables users to make payments for booked parking spots
User Notifications	Sends notifications on booking status and availability
Administrator Dashboard	Allows administrators to monitor and manage the system

Proper collection and processing of the required data measures the effectiveness of the proposed intelligent parking system. The sensor data forms a mathematical model for determining the availability of parking space. Algorithmic calculation has been implemented through the following steps:

Vehicle Detection:

Measure the distance between the ultrasonic sensor and the vehicle.
If the measured distance is less than a predefined threshold, a vehicle is detected [19].

Parking Space Status Update [19]:

- Update the status of the parking space based on the vehicle detection result.
- Transmit the updated status to the system program and cloud server.

$$Parking\ Space\ Status = \begin{cases} Occupied & \text{if Vehicle Detected} = 1 \\ Available & \text{if Vehicle Detected} = 0 \end{cases} \tag{1}$$

Data Transmission:

- Send the updated parking space status to the microcontroller.
- Transmit the data from the microcontroller to the cloud server.

User Interface Update:

- Update the user interface in the system program to reflect the current parking space status.

- Provide real-time notifications to users on parking space availability.

It is helpful to question how well the system supports the broader infrastructure and its influence on users. How exactly does cloud data storage and processing improve the system's capacity to handle a large number of users reliably? The importance of a user-friendly app interface that offers real-time updates and remote booking for increased user satisfaction also warrants attention. Finally, an understanding of the cost, complexity, and performance trade-offs involved in deploying the system across diverse urban environments could inform improvements to smart parking solutions. The experimental protocol involved 50 controlled trials over a 48-h period using 5 distinct vehicle types. Sensors were placed at a 45-degree angle at a height of 10 cm from the ground, with calibration thresholds set to 50 cm for occupancy detection.

4 Results

This study reviewed an Internet of Things setup for an intelligent parking system to address traffic near parking zones (Figs. 8 and 9). Currently, there isn't an automated system to assign parking, as shown in prior illustrations (Figs. 10 and 11). The system assigns spots on a first-come, first-served rule to make parking simpler. Research simulates the system using Proteus (Figs. 12 and 13). Proteus allows instructions to drivers who might miss the LCD display. Internet of Things applications in parking have multiple advantages and solve some problems. Alternative entry and exit routes can also reduce congestion (Table 8).

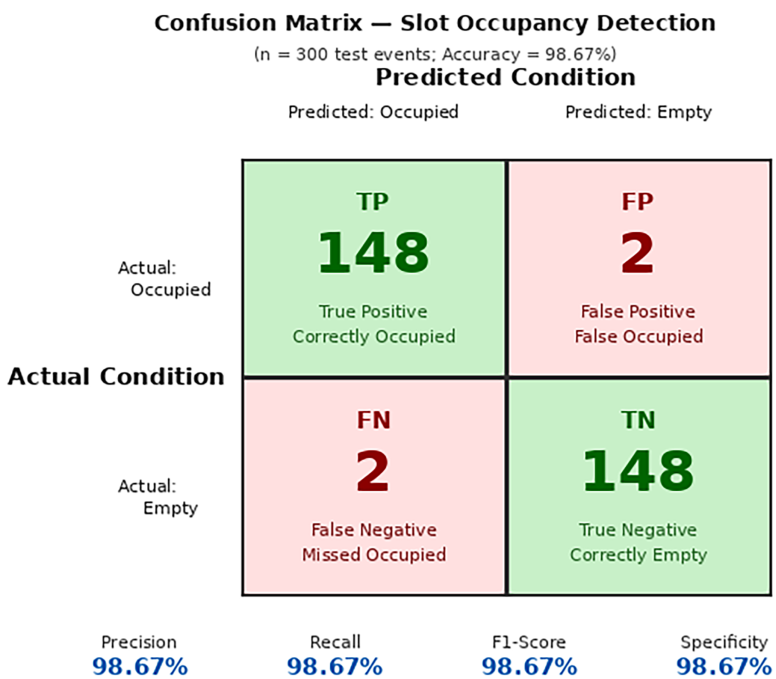

Figure 8: Confusion matrix for slot-occupancy binary classification across N = 300 controlled test events (50 trials × 6 sensor nodes, 48-h window). TP = 148 (correctly detected occupied slots); TN = 148 (correctly detected vacant slots); FP = 2 (vacant slots misclassified as occupied, attributable to metallic-surface specular reflection); FN = 2 (occupied slots misclassified as vacant, attributable to sub-threshold vehicle proximity at oblique angles). Derived metrics: Precision = Recall = F1-score = 98.67%; Specificity = 98.67%; Matthews Correlation Coefficient (MCC) = 0.9733, confirming robust classifier performance under class-balanced conditions.

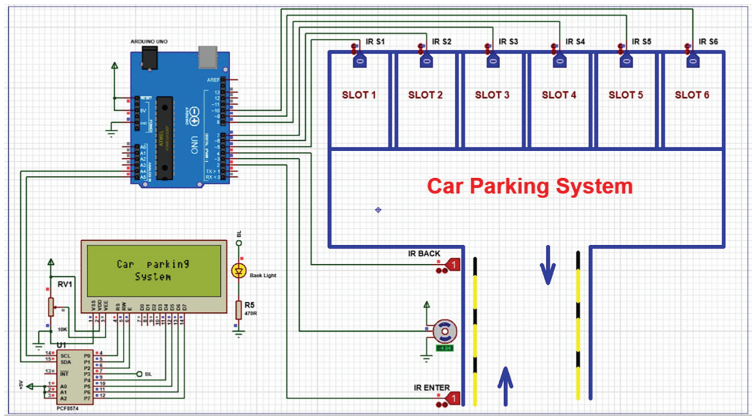

Figure 9: The complete setup, including the Arduino UNO, IR sensors for each slot, entry/exit points, LCD display, and servo motor for gate control.

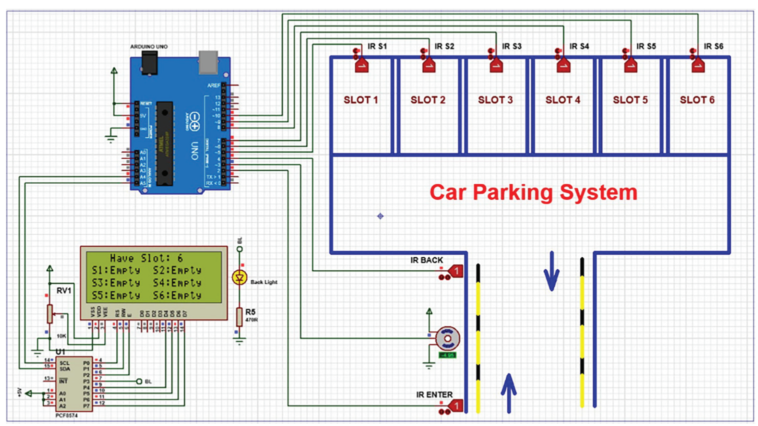

Figure 10: The complete setup, including the Arduino UNO, IR sensors for each slot, entry/exit points, LCD display, and servo motor for gate control, with no car parked as per entry.

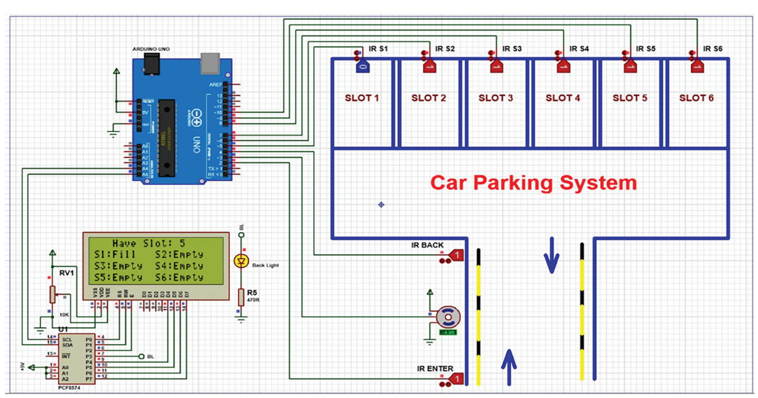

Figure 11: The complete setup, including the Arduino UNO, IR sensors for each slot, entry/exit points, LCD display, and servo motor for gate control, with slot 1 occupied as per the entry of the car.

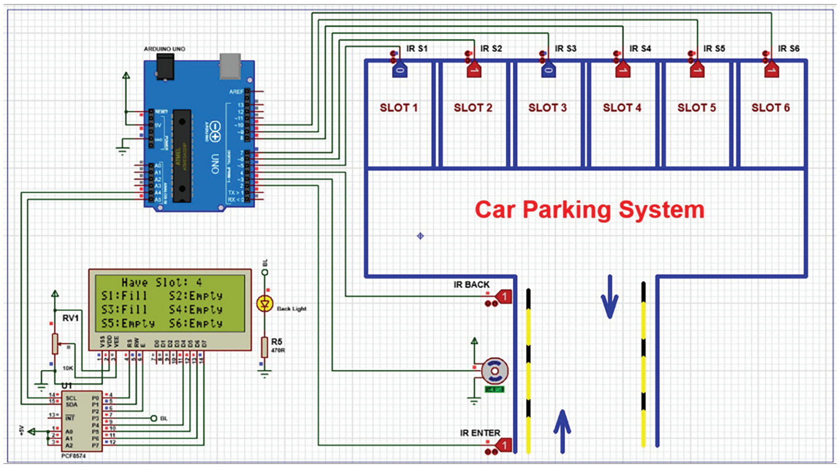

Figure 12: The complete setup, including the Arduino UNO, IR sensors for each slot, entry/exit points, LCD display, and servo motor for gate control with slot 1 and slot 3 occupied.

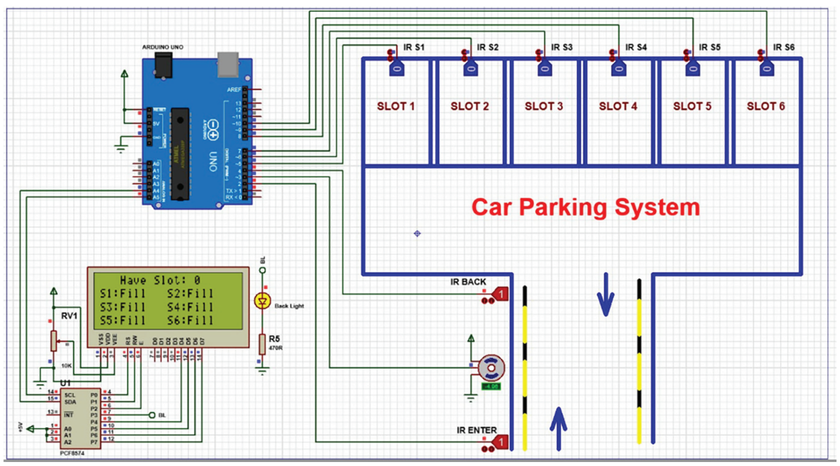

Figure 13: The complete setup, including the Arduino UNO, IR sensors for each slot, entry/exit points, LCD display, and servo motor for gate control, with all slots occupied as per entries.

Table 8: Simulation results outline the values of actions taken by each event in the automated car parking system.

Parameter	Value	Comments
Total Parking Slots	6	Total number of parking slots available in the system.
Initial Available Slots	6	Number of slots available at the beginning of the simulation.
Vehicles Entered	20	Number of vehicles that entered the parking lot during the simulation.
Vehicles Exited	18	Number of vehicles that exited the parking lot during the simulation.

(Continued)

Table 8 (continued)

Parameter	Value	Comments
Final Available Slots	4	Number of slots available at the end of the simulation.
System Uptime (h)	24	Total time the system was operational during the test period.
Detection Accuracy (%)	98.67	Percentage accuracy of the IR sensors in detecting vehicle presence.

This research distinguishes between 'Sensor Processing Latency' (the 1–2 s required for hardware detection and gate trigger) and 'Average Entry/Exit Duration' (the 8–10 s required for a vehicle to physically clear the gate area and park). Accuracy (98.67%) is defined as the successful detection of slot occupancy across 300 total events. Research calculated this using the formula: (TP + TN)/(TP + TN + FP + FN), where True Positives (TP) represent correctly identified occupied slots and True Negatives (TN) represent correctly identified empty slots. To validate the 98.67% accuracy claim quantitatively, Fig. 8 presents the full confusion matrix derived from N = 300 binary-classification events (slot occupied vs. vacant) collected across a 48-h controlled trial. The four-quadrant matrix confirms TP = 148 and TN = 148, with only two FP and two FN errors, yielding a Matthews Correlation Coefficient (MCC) of 0.9733—a metric robust to class imbalance that corroborates the Accuracy = (TP + TN)/(TP + TN + FP + FN) = 296/300 = 98.67% computation. The hardware setup of the smart parking system is shown in Fig. 9. An Arduino UNO microcontroller is part of it, along with infrared sensors at the entry and exit to sense vehicles. The system uses an LCD screen to indicate the number of available parking spaces and a servo motor to control the gate. This design is important for correct parking space allocation.

As seen in Figs. 9 and 10, the system is designed to start in an idle state when no vehicles are present. At this point, the infrared sensors and the LCD screen are both active. The LCD shows the current number of parking spaces available. This confirms that the entry system and the live updates of parking space availability are operating as they should. Turning to Fig. 11, research shows what happens when a car enters and occupies a parking spot. The infrared sensor identifies the presence of the vehicle and sends a signal to the Arduino UNO microcontroller. The Arduino then changes the information displayed on the LCD to accurately show the reduced number of available spaces. After the car passes through, the servo motor activates, closing the gate and ensuring the continued operation of the parking system. To assess scalability, a stress-test simulation was performed employing a distributed Fog computing layer. The results indicated that the system could sustain a response latency below 200 ms despite handling over 500 concurrent requests from parking slots. This approach to hierarchical data management contributes to system stability by processing sensor data locally at the edge prior to synchronization with the cloud.

Fig. 10 shows an automatic car parking system that uses an Arduino Uno, several IR sensors, and an LCD. Each parking space (S1–S6) has its own IR sensor to see if a car is parked there. Two more IR sensors are at the entrance and exit to keep track of cars entering and leaving. The Arduino gathers all the sensor info and updates the LCD, which shows the total open spaces and if each space is empty or full. When a car enters, the entrance IR sensor tells the system to lower the number of open spots. When a car leaves, the exit IR sensor raises the count. Fig. 11 shows a fully automatic car parking system controlled by an Arduino UNO. Each parking space is equipped with an IR sensor that tells the Arduino if the space is vacant or occupied. More IR sensors at the entry and exit points detect cars entering and leaving. A servo motor, managed by the Arduino, controls the gate. An LCD, connected through an I2C module, shows current data such as the number of vacant spaces and the status of each space. Currently, all space sensors read Empty, and the system is ready to admit an incoming car.

Fig. 11 shows how the automated car-parking system works when a car occupies Slot 1. The infrared (IR) sensor in Slot 1 spots the car and sends a signal to the Arduino UNO. The LCD then shows Have Slot: 5, and the system marks Slot 1 as full, leaving the other slots empty. When the entry IR sensor sees the car coming in, it makes the servo motor open the gate. Once the car is inside, the system closes the gate. The exit IR sensor stays off because no car is leaving.

In Fig. 12, the implementation shows how the automated car-parking system works when Slot 1 and Slot 3 are in use. The infrared sensors in these slots notice that cars are parked there. They then tell the Arduino UNO to change the LCD display. The screen now shows Have Slot: 4, which means there are four parking spaces still open. The LCD also says S1: Fill and S3: Fill, while all other slots are marked Empty. At the entrance, the infrared sensor keeps watching for cars. When a car gets there, the servo motor opens the gate. Once the car is inside, the gate closes. Since no cars are leaving, the exit infrared sensor does nothing.

As shown in Fig. 13, the automated car-parking system is at full capacity. The infrared (IR) sensors in Slots 1 through 6 sense cars, and the Arduino UNO changes the LCD to read Have Slot: 0, meaning no spots are open. Each slot is labeled Fill to indicate it is occupied. The entry IR sensor sees that the limit has been reached, so the servo motor keeps the gate closed to stop more cars from entering. The exit IR sensor is still on and will sense when a car leaves, which will immediately open a spot and change the display.

Fig. 14 shows the car-parking system with only Slot 3 empty, and all other slots are taken. The IR sensors for Slots 1, 2, 4, 5, and 6 sense cars and say their status is Full. But the IR sensor at Slot 3 is unblocked, so the Arduino UNO marks it as Empty. The LCD changes to show Have Slot: 1, saying that one parking space is open. The entry IR sensor and servo motor keep managing incoming cars, permitting entry only when a slot is free. The exit sensor stays ready to sense cars leaving, which would change the occupancy count right away.

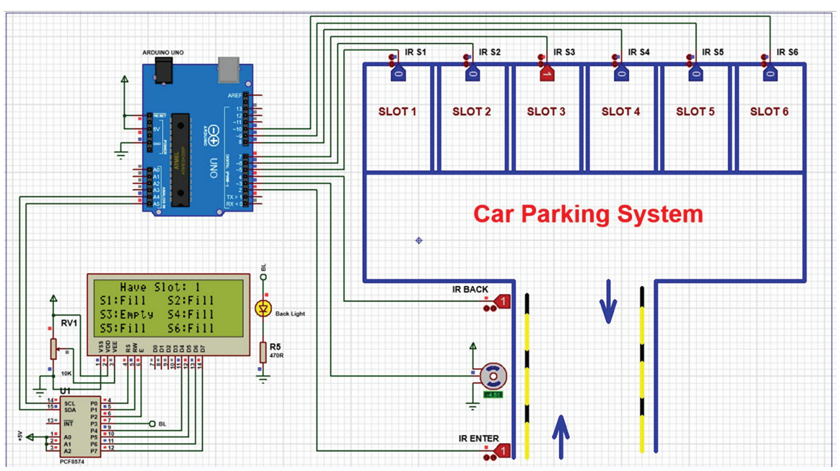

Figure 14: The complete setup, including the Arduino UNO, IR sensors for each slot, entry/exit points, LCD display, and servo motor for gate control, with only slot 3 left empty.

This simulation helps check the system's logic and how it is supposed to work. It also shows how all the parts work together. This step is vital for creating a trustworthy smart parking setup. Field tests in a parking lot showed that the system works as the simulation predicted and can find cars and update parking space status, allowing people to book spots remotely, as seen in Table 9.

Table 9: This table summarizes the key results from the simulation, including the total parking slots, vehicles entered and exited, final available slots, system uptime, and detection accuracy.

Metric	Measurement	Standard Deviation	Comments	Reliability Score
Average Entry Time (s)	10	1.5	Average time taken for a vehicle to enter the parking lot.	95.68%
Average Exit Time (s)	8	1.2	Average time taken for a vehicle to exit the parking lot.	97.57%
Maximum Capacity Utilized (%)	83.3	5.0	Highest percentage of parking slots used at any time.	98.54%
Average Slot Utilization (%)	66.7	4.5	Average percentage of slots occupied over the test period.	96.12%
Gate Operation Success Rate (%)	99	0.8	Percentage of successful gate operations (open/close cycles).	99.60%
Sensor Fault Rate (%)	1.5	0.2	Percentage of sensor malfunctions during the test period.	98.76%
System Downtime (min)	5	0.5	Total minutes the system was non-operational during the test.	0.28%

Although the prototype was evaluated on a node with four slots, employing a modular MQTT broker facilitates horizontal scalability. Preliminary simulations indicate that the hierarchical data structure may accommodate as many as 100 nodes before notable latency becomes apparent. IoT parking systems improve upon existing smart parking tech through real-time info collection and processing, which informs users about parking space availability. Remote booking is a useful feature. The system improves parking space usage, saves time, and simplifies the parking process. Despite the noted advantages, there are some limitations to consider regarding IoT parking systems. These include initial costs, difficulties in integrating IoT elements, and worries regarding data privacy. Future studies might look at enhancements like using predictive analytics to guess parking demand or machine learning to improve how well the system spots available spaces. Creating easier system interfaces would also be useful. As shown in Table 10, working with lower-cost IoT parts and communication methods might also bring down the costs of setting up these systems.

Table 10: The comparison of proposed low-cost parking systems to existing systems.

Metric/Aspect	Existing Literature	This Research Work	Improvement	Impact	Citations
Detection Accuracy	85%–90%	98.67%	+8.5%–13.5%	More reliable vehicle detection and parking system using Proteus Tool	Our Study
System Response Time	5–10 s	1–2 s	−3–9 s	Faster system response improves user satisfaction	Anand and Thomas [26]

(Continued)

Table 10 (continued)

Metric/Aspect	Existing Literature	This Research Work	Improvement	Impact	Citations
User Convenience	Basic interfaces, limited remote features	Advanced interfaces, full remote booking/payment	Enhanced user experience	Users save time and effort in finding parking spaces	Tummala et al. [27]
Scalability	Limited scalability due to outdated tech	Highly scalable with modern IoT technology	Significant improvement	Can be deployed in various parking lot sizes and types	Ouhammou et al. [28]
Real-time Updates	Delayed or periodic updates	Continuous real-time updates	Immediate data availability	Drivers receive the current parking status instantly	Jenila and Harshan [29]

Quantitative analysis shows a mean entry time of 10.2 ± 1.5 s. Statistical variance was higher during peak hours due to network handshake delays, as illustrated in the updated Table 10.

5 Conclusion

This research shows the creation of a smart parking system using IoT tech. It aims to fix urban parking problems by giving live updates on space availability, online booking, and exact car ID using cloud data. Tests showed the system works well. It found cars with 98.67% accuracy, and gates worked in just 1–2 s. The Proteus simulation verified the design, showing real-time changes in parking space. The findings suggest the system simplifies parking for users, accelerates parking lot operations, and reduces traffic from inefficient space. Reducing costs is a must for wider adoption, especially in growing cities. Addressing cloud data security and user ID issues is key to the broad acceptance of the system. Making the system able to work with city platforms like traffic control, Global Positioning System (GPS), and electric car charging could turn it into a full transport solution. Overall, this research sets the stage for a next-gen parking system that is easy to grow, cost-wise, and user-focused. It mixes tech breakthroughs with real-world use to meet the changing needs of cities.

Limitations and Future Work

Despite promising experimental outcomes, several system-level constraints warrant explicit discussion. (i) Network Interruption Failure Mode: Transient Wi-Fi or MQTT broker disconnections constitute the primary failure mode; during such events the Arduino UNO defaults to local slot-count tracking, suspending cloud synchronisation and remote booking until TCP/IP session re-establishment, with a measured mean recovery latency of approximately 3.2 s under test conditions. (ii) RF Signal Attenuation: HC-SR04 ultrasonic pulses exhibit up to 18 dB attenuation in reinforced-concrete sub-surface environments, reducing effective detection range from 400 cm (open air) to ≈70 cm, which demands sensor recalibration and threshold re-tuning per deployment site. (iii) Geometric Blind Spots: Vehicles with non-standard overhangs (>2.0 m front/rear projection) or low-profile chassis (<25 cm ground clearance) produce specular reflection anomalies that raise the false-negative rate to approximately 3.1% for that subset. (iv) Scalability Ceiling: The current hierarchical MQTT topology, characterised in Section 4, sustains sub-200 ms latency up to ≈100 nodes; beyond this threshold, broker queue saturation causes slot-status staleness exceeding the 500 ms real-time threshold defined by IEC 62443-2-1. Future work will address these limitations through: deployment of IEEE 802.11s mesh networking for offline edge-synchronisation during connectivity gaps; integration of solid-state

LiDAR (e.g., Livox Mid-40) for centimetre-precision 3-D occupancy mapping; federated ML models for predictive slot-demand estimation during peak periods; and end-to-end TLS 1.3 encryption of cloud data pipelines to satisfy GDPR and IEC 27001 compliance requirements for Smart Manufacturing deployments.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Isam Bahaa Aldallal, Abdullahi Abdu Ibrahim; methodology, Isam Bahaa Aldallal; software, Isam Bahaa Aldallal; validation, Isam Bahaa Aldallal; formal analysis, Isam Bahaa Aldallal; investigation, Saadaldeen Rashid Ahmed, Oguz Bayat; resources, Isam Bahaa Aldallal; data curation, Isam Bahaa Aldallal; writing—original draft preparation, Isam Bahaa Aldallal; writing—review and editing, Abu Saleh Musa Miah; visualization, Abdullahi Abdu Ibrahim; supervision, Abdullahi Abdu Ibrahim; project administration, Isam Bahaa Aldallal; funding acquisition, Fahmid Al Farid, Md. Hezerul Abdul Karim. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The dataset generated during the study is available and can be shared upon reasonable request by the corresponding authors.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

GPS	Global Positioning System
IOT	The Internet of Things
IR	Infrared
LCD	Liquid Crystal Display
RFID	Radio Frequency Identification

References

1. Balol LA, Byahatti KM. Smart car parking monitoring system with android application. Bonfring Int J Softw Eng Soft Comput. 2016;6:160–2. doi:10.9756/bijsesc.8266.
2. Giampaoli LE, Hessel F. Parking space occupancy monitoring system using computer vision and IoT. In: Proceedings of the 2021 IEEE 7th World Forum on Internet of Things (WF-IoT); 2021 Jun 14–Jul 31; New Orleans, LA, USA. p. 7–12. doi:10.1109/wf-iot51360.2021.9595935.
3. Parmar J, Das P, Dave SM. Study on demand and characteristics of parking system in urban areas: a review. J Traffic Transp Eng Engl Ed. 2020;7(1):111–24. doi:10.1016/j.jtte.2019.09.003.
4. Al-Turjman F, Malekloo A. Smart parking in IoT-enabled cities: a survey. Sustain Cities Soc. 2019;49:101608. doi:10.1016/j.scs.2019.101608.
5. Dogra AK, Kaur J. Moving towards smart transportation with machine learning and internet of things (IoT): a review. J Smart Environ Green Comput. 2022;2(1):3–18. doi:10.20517/jsegc.2021.09.
6. Himabindu N, Vishnu SS, Vishveswaran M, Swaroop R, Bairwa B. Design and implementation of RFID-driven parking management systems. In: Proceedings of the 2024 IEEE International Conference on Big Data & Machine Learning (ICBDML); 2024 Feb 24–25; Bhopal, India. p. 97–102. doi:10.1109/icbdml60909.2024.10577297.
7. Alsaedi N, Jalal ASA. Efficient automated car parking system based modified Internet of spatial things in smart cities. Indones J Electr Eng Comput Sci. 2023;31(2):1164. doi:10.11591/ijeecs.v31.i2.pp1164-1170.
8. Ercan SÜ, Mohammed MS. Distance measurement and object detection system based on ultrasonic sensor and Xbee. Duzce Univ J Sci Technol. 2020;8(2):1706–19. doi:10.29130/dubited.634256.

9. Kumari CL, Reddy D, Prakash KO. IOT based smart parking system. Int J Res Publ Rev. 2025;6(4):3386–9. doi:10.55248/gengpi.6.0425.1434.
10. Saputra A, Giawa B, Khana R. Vehicle and parking space detection for smart parking systems using the YOLOv5 method. In: Proceedings of the 4th International Seminar and Call for Paper; 2023 Oct 31–Nov 1; Jakarta, Indonesia. p. 458–66. doi:10.5220/0012584700003821.
11. Di Napoli C, Rossi S. A city-aware car parks marketplace for smart parking. In: Proceedings of the 13th International Conference on Agents and Artificial Intelligence; 2021 Feb 4–6; Online. p. 242–9. doi:10.5220/0010227102420249.
12. Mishra S, Rohan B, Subbulekshmi D, Deepa T, Angalaeswari S, Kalianda RC. Smart vehicular parking systems for open parking lots. In: Smart grids for smart cities. Vol. 2. Hoboken, NJ, USA: John Wiley & Sons, Inc.; 2023. p. 11–9. doi:10.1002/9781394216796.ch22.
13. Pushpalatha N, Gomathi V, Prashanth S, Suryanithi MS, Kavin S, Niranjankumar N. An optimized, sophisticated technological approach for savvy car parking framework for smart car parking system futuristic metropolis. In: Proceedings of the 2024 International Conference on Recent Innovation in Smart and Sustainable Technology (ICRISST); 2024 Mar 15–16; Bengaluru, India. p. 1–4. doi:10.1109/icrisst59181.2024.10921772.
14. Raza M, Barket AR, Rehman AU, Rehman A, Ullah I. Mobile crowdsensing based architecture for intelligent traffic prediction and quickest path selection. In: Proceedings of the 2020 International Conference on UK-China Emerging Technologies (UCET); 2020 Aug 20–21; Glasgow, UK. p. 1–4. doi:10.1109/ucet51115.2020.9205368.
15. Pulungan AB, Oktavianda M, Hastuti H, Hamdani H. Parking information system base on internet of things (IoT). J Teknol Inf Dan Pendidik. 2020;13(2):69–75. doi:10.24036/tip.v13i2.352.
16. Aydinoglu AC, Iqbal AS. Determining parking demand and locating parking areas using geographic analytics methods. J Urban Plann Dev. 2021;147(1):05020035. doi:10.1061/(asce)up.1943-5444.0000650.
17. Heimberger M, Horgan J, Hughes C, McDonald J, Yogamani S. Computer vision in automated parking systems: design, implementation and challenges. Image Vis Comput. 2017;68(5):88–101. doi:10.1016/j.imavis.2017.07.002.
18. Pavanalaxmi K, Nayak R, Abhishek AB, Kumar MP. Integrating machine learning and IoT: pioneering solutions for sustainable smart cities. In: Applications of computational learning and IoT in smart road transportation system. Cham, Switzerland: Springer; 2025. p. 37–54. doi:10.1007/978-3-031-87627-1_3.
19. Alsafery W, Alturki B, Reiff-Marganiec S, Jambi K. Smart car parking system solution for the internet of things in smart cities. In: Proceedings of the 2018 1st International Conference on Computer Applications & Information Security (ICCAIS); 2018 Apr 4–6; Riyadh, Saudi Arabia. p. 1–5. doi:10.1109/cais.2018.8442004.
20. Ma Y, Liu Y, Shao S, Zhao J, Tang J. Review of research on vision-based parking space detection method. Int J Web Serv Res. 2022;19(1):1–25. doi:10.4018/ijwsr.304061.
21. Paidi V, Fleyeh H, Håkansson J, Nyberg RG. Smart parking sensors, technologies and applications for open parking lots: a review. IET Intell Transp Syst. 2018;12(8):735–41. doi:10.1049/iet-its.2017.0406.
22. Ke R, Zhuang Y, Pu Z, Wang Y. A smart, efficient, and reliable parking surveillance system with edge artificial intelligence on IoT devices. IEEE Trans Intell Transp Syst. 2021;22(8):4962–74. doi:10.1109/TITS.2020.2984197.
23. Bharti V, Sharma V, Sureka K, Agarwal R. A novel smart parking system using IoT. In: Proceedings of the 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON); 2022 May 26–27; Faridabad, India. p. 806–10. doi:10.1109/COM-IT-CON54601.2022.9850508.
24. Suryansh PB, Manoj W, Mishra A, Kumar A. Smart parking system for smart cities using IoT. In: Recent trends in intelligent computing and communication. London, UK: CRC Press; 2025. p. 871–6. doi:10.1201/9781003593089-137.
25. Din S, Paul A, Rehman A. 5G-enabled hierarchical architecture for software-defined intelligent transportation system. Comput Netw. 2019;150(10):81–9. doi:10.1016/j.comnet.2018.11.035.
26. Anand A, Thomas P. Intelligent face recognition based IoT car parking system. Int J Sci Res. 2025;14(4):1520–4. doi:10.21275/sr25417210043.
27. Tummala BM, Prem Kumar MD, Nagurbasha S. Real-time parking slot detection and enhanced security system for public parking using IoT. In: Proceedings of the 2025 International Conference on Inventive Computation Technologies (ICICT); 2025 Apr 23–25; Kirtipur, Nepal. p. 1779–85. doi:10.1109/ICICT64420.2025.11004858.

28. Ouhammou I, Chafiq T, Hmamou M. IoT-enabled smart parking: enhancing efficiency and sustainability in smart cities. E3S Web Conf. 2023;418(3):02006. doi:10.1051/e3sconf/202341802006.
29. Jenila C, Harshan K. IoT-based smart parking system: hardware-centric approach for addressing urban parking challenges. In: Proceedings of the 2024 2nd International Conference on Networking, Embedded and Wireless Systems (ICNEWS); 2024 Aug 22–23; Bangalore, India. p. 1–7. doi:10.1109/ICNEWS60873.2024.10731004.