IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

IoT Based Smart Parking Prediction System

Eswaran K and S. Kamalakkannan
Depaertment of Computer Applications
VELS Institute of Science, Technology and Advanced Studies (VISTAS), Pallavarm, Chennai, India
23105405@vistas.ac.in | kannan.scs@vistas.ac.in

Abstract: *With rapid urbanization, finding available parking spaces in congested areas has become a significant challenge for vehicle owners worldwide. Conventional parking systems lack real-time availability information, resulting in wasted time, increased fuel consumption, and heightened traffic congestion. This paper proposes an IoT-Based Smart Parking Prediction System that leverages HCSR04 ultrasonic sensors, NodeMCU ESP8266 microcontrollers, and a Random Forest machine learning algorithm to detect, monitor, and predict parking space availability in real time. Sensor data is transmitted to an AWS EC2 cloud server via the MQTT protocol over Wi-Fi, and slot information is presented to users through a ReactJS web dashboard and mobile interface. The system was piloted across 20 parking slots over a two-week period. Experimental results demonstrate 98.6% detection accuracy, 92.4% prediction accuracy for 30-minute forecasts, and an end-to-end latency of 1.2 seconds. Pilot users reported approximately 60% reduction in parking search time, confirming the system's potential to improve urban mobility significantly.*

Keywords: IoT, Smart Parking, Ultrasonic Sensor, NodeMCU ESP8266, Machine Learning, Random Forest, Real-Time Monitoring, MQTT, Cloud Computing, Prediction System

I. INTRODUCTION

The rapid growth of urban populations has led to an unprecedented surge in vehicle ownership, making parking management one of the most pressing challenges in modern smart city development. In densely populated metropolitan areas, drivers can spend up to 30% of total travel time searching for parking, contributing directly to traffic congestion, air pollution, fuel wastage, and driver frustration [1].

Traditional parking systems are static and lack real-time feedback mechanisms for drivers. These systems do not provide advance information about slot availability, leading to inefficient circulation patterns within parking facilities. Moreover, manual monitoring approaches are labor-intensive and prone to errors, making them unsuitable for large-scale smart city deployment.

The emergence of the Internet of Things (IoT) offers transformative opportunities to convert conventional parking infrastructure into intelligent, connected systems capable of monitoring and predicting parking availability dynamically. IoT-enabled systems can integrate low-cost sensors, wireless communication, cloud platforms, and machine learning to deliver actionable real-time insights to both parking operators and drivers.

This paper proposes an IoT-Based Smart Parking Prediction System that deploys HC-SR04 ultrasonic sensors connected to NodeMCU ESP8266 microcontrollers to detect vehicle presence in individual parking slots. The collected binary occupancy data is transmitted over Wi-Fi using the MQTT protocol to a cloud platform hosted on AWS EC2. A machine learning model based on the Random Forest algorithm analyzes historical occupancy patterns to predict future slot availability. Users access both real-time and predicted slot information through a ReactJS web dashboard or mobile application.

The remainder of this paper is structured as follows: Section II reviews related work; Section III presents the proposed system architecture; Section IV describes implementation details; Section V discusses experimental results; Section VI outlines application domains; and Section VII concludes the paper with future research directions.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

II. LITERATURE REVIEW

Considerable research has been directed toward IoT-based parking solutions in recent years. Kotb et al. [1] introduced iParker, a smart parking guidance system that used wireless sensor networks to detect slot occupancy and guide drivers through dynamic resource allocation. While effective for real-time detection, their system lacked predictive capability and was constrained in scalability due to high infrastructure costs.

Suresh et al. [2] developed a cloud-connected parking system integrating RFID tags and Arduino microcontrollers to automate entry/exit logging and slot allocation. Although this approach improved operational efficiency, it relied on RFID-based identification rather than continuous sensor monitoring, and did not incorporate any machine learning component for occupancy forecasting.

Lin et al. [3] conducted a comprehensive survey of smart parking solutions and demonstrated that machine learning algorithms—including Random Forest and Long Short-Term Memory (LSTM) neural networks—can predict parking occupancy with high accuracy when trained on historical sensor data. Their findings confirmed the feasibility of combining IoT sensor inputs with predictive analytics but did not present an end-to-end integrated prototype.

Despite these contributions, existing systems frequently suffer from limited scalability, high hardware costs, or a lack of tight integration between the detection, communication, and prediction layers. The proposed system addresses these gaps by employing low-cost NodeMCU hardware, the lightweight MQTT protocol, and a Python-based Random Forest prediction engine, all integrated into a full-stack cloud architecture.

Table I: Comparison of Related Work with the Proposed System

Feature	Kotb et al.	Suresh et al.	Lin et al.	Proposed System
Real-Time Detection	Yes	Yes	Yes	Yes
ML Prediction	No	No	Yes	Yes
Cloud Integration	Partial	Yes	No	Yes
Mobile Dashboard	No	No	No	Yes
Low-Cost Hardware	No	Partial	N/A	Yes
Scalability	Limited	Limited	N/A	High

III. PROPOSED SYSTEM ARCHITECTURE

The proposed system adopts a three-tier IoT architecture comprising a Perception Layer, a Network Layer, and an Application Layer. Each layer is responsible for a distinct set of operations, and the three layers interact seamlessly to provide an end-to-end smart parking solution. Figure 1 illustrates this architecture.

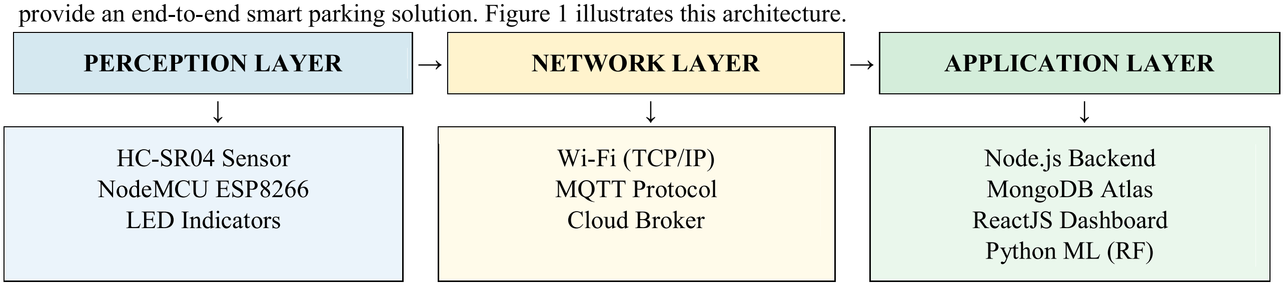

Fig. 1. Three-Tier IoT Architecture of the Smart Parking Prediction System

DOI: 10.48175/IJARSCT-34445

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

A. Perception Layer

The Perception Layer constitutes the physical hardware deployed at each parking slot. An HC-SR04 ultrasonic sensor is mounted at the entry level of every slot. The sensor emits ultrasonic pulses at 40 kHz and measures the echo return time to calculate the distance between the sensor and any object in its field of view. A threshold distance of 20 cm is configured: if the measured distance is less than 20 cm, the slot is classified as occupied; otherwise, it is classified as vacant. This binary classification is highly robust under varying lighting conditions, as ultrasonic sensing is inherently light-independent.

The NodeMCU ESP8266 module reads the sensor output through its GPIO pins and encodes the occupancy status as a binary value (0 for vacant, 1 for occupied). LED indicators at each slot provide immediate visual feedback to drivers in the vicinity.

B. Network Layer

The Network Layer handles wireless communication between IoT nodes and the cloud backend. The MQTT (Message Queuing Telemetry Transport) protocol is employed due to its lightweight publishsubscribe architecture, low bandwidth requirements, and suitability for constrained IoT devices. Each NodeMCU publishes slot status updates to a dedicated MQTT topic (e.g., parking/slot/1) at five-second intervals. The MQTT broker receives all published messages and routes them to the Node.js backend subscriber in real time.

C. Application Layer

The Application Layer encompasses the cloud backend, the machine learning prediction module, and the user-facing interface. The Node.js backend subscribes to MQTT topics, processes incoming messages, and persists timestamped occupancy records to a MongoDB Atlas cloud database. The Python-based prediction module exposes a REST API endpoint that the backend queries to obtain 30-minute occupancy forecasts. The ReactJS frontend renders a color-coded slot map that is updated in near real time.

IV. SYSTEM IMPLEMENTATION

A. Hardware Setup and Sensor Interfacing

Each of the 20 parking slots in the pilot facility was equipped with one HC-SR04 ultrasonic sensor and one NodeMCU ESP8266 module. The sensor's TRIG pin is connected to GPIO pin D1 of the NodeMCU, and the ECHO pin is connected to GPIO pin D2. The NodeMCU firmware, written in C/C++ using the Arduino IDE, triggers the sensor every 5 seconds, measures the echo time, converts it to a distance value in centimeters, and determines slot occupancy using the 20 cm threshold. The firmware then constructs a JSON payload containing the slot identifier, occupancy status, and a local timestamp before publishing it to the MQTT broker via the built-in Wi-Fi module.

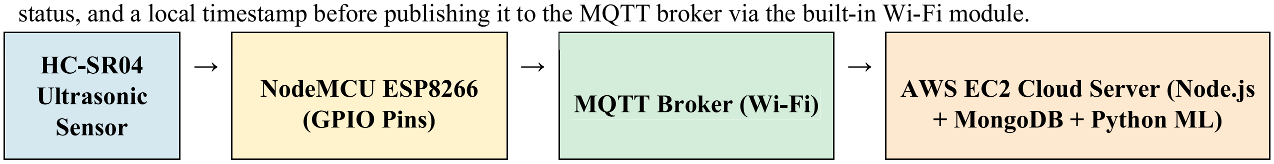

Fig. 2. Hardware Connection and Data Flow Diagram

B. MQTT Communication Protocol

The MQTT protocol operates over TCP/IP and is well-suited for IoT deployments where network bandwidth and device processing power are limited. In the proposed system, the Mosquitto MQTT broker is hosted on the AWS EC2 instance. Each NodeMCU publishes to a topic structured as parking/slot/{slotID}. The Node.js backend, acting as a persistent subscriber, receives messages from all slot topics simultaneously and invokes a data-persistence function to store records in MongoDB Atlas with millisecond-precision timestamps. Quality of Service (QoS) level 1 is configured to ensure at-leastonce delivery of slot updates.

C. Machine Learning Prediction Module

Historical occupancy records accumulated in MongoDB Atlas are periodically exported and used to train the Random Forest Classifier implemented using the scikit-learn library [5] in Python. The feature vector for each training sample

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

includes: (i) hour of day (0–23), (ii) day of week (0–6), (iii) slot identifier, and (iv) rolling occupancy rate over the preceding 30-minute window. The target label is the occupancy status 30 minutes into the future. The trained model is serialized using the joblib library and deployed as a Flask REST API endpoint on the same AWS EC2 instance. When the backend queries this endpoint with current context features, the model returns a predicted occupancy probability in real time.

Raw Occupancy Data (MongoDB)	→	**Feature Engineering (Hour, Day, Slot ID)**	→	**Random Forest Classifier (scikit-learn)**	→	**Predicted Occupancy Probability (REST API)**

Fig. 3. Machine Learning Prediction Pipeline

D. Workflow Overview

Figure 4 illustrates the complete operational workflow of the proposed system, from vehicle arrival at a parking slot to the display of real-time and predicted information on the user dashboard.

Step 1: Vehicle Arrives at Parking Slot

↓

Step 2: HC-SR04 Sensor Detects Presence (Distance < 20 cm)

↓

Step 3: NodeMCU Publishes Slot Status via MQTT (Every 5 sec)

↓

Step 4: Cloud Backend Subscribes & Stores in MongoDB Atlas

↓

Step 5: Random Forest ML Model Predicts Availability (30 min)

↓

Step 6: ReactJS Dashboard Updates Color-Coded Slot Map

↓

Step 7: User Views Real-Time & Predicted Slot Information

Fig. 4. End-to-End Operational Workflow of the Smart Parking System

E. Frontend Dashboard

The ReactJS frontend communicates with the backend over WebSockets for real-time slot updates and REST API calls for prediction data. The dashboard renders a color-coded grid layout of the parking facility, where green cells represent available slots and red cells represent occupied slots. Predicted availability is displayed as a percentage probability for each slot for the next 30-minute window. Figure 5 shows the simulated dashboard layout for the 20-slot pilot facility.

	Smart Parking Dashboard — Real-Time Slot Map			
S1 ●	S2 ○	S3 ●	S4 ○	S5 ●
S6 ○	S7 ○	S8 ●	S9 ○	S10 ●
S11 ●	S12 ○	S13 ○	S14 ●	S15 ○
S16 ○	S17 ●	S18 ○	S19 ○	S20 ●
● Available (Green)			● Occupied (Red)	

Fig. 5. Simulated ReactJS Dashboard Layout — 20-Slot Pilot Facility

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

V. RESULTS AND DISCUSSION

The system was deployed in a pilot parking facility consisting of 20 slots arranged in 4 rows of 5 slots each. Comprehensive testing was conducted over a two-week period to evaluate detection accuracy, system latency, prediction performance, scalability, and user experience. Table II summarizes the key performance metrics obtained during the pilot evaluation.

Table II: System Performance Metrics from Pilot Evaluation

Metric	Value	Remarks
Detection Accuracy	98.6%	Varying lighting & weather
End-to-End Latency	1.2 seconds	MQTT-cloud pipeline
Prediction Accuracy (30 min)	92.4%	Random Forest, 14-day training
Scalability	200+ nodes	No performance degradation
Parking Search Time Reduction	~60%	Reported by pilot users

A. Detection Accuracy

The HC-SR04 ultrasonic sensor demonstrated a detection accuracy of 98.6% across the 20-slot facility under varying environmental conditions, including different ambient lighting levels, slight variations in vehicle positioning, and intermittent weather changes. The sensor's light-independent operation was a key advantage over camera-based detection systems. The 1.4% error rate was primarily attributed to rare edge cases where motorcycle handlebars or bicycles were parked at the boundary of the 20 cm threshold distance.

B. Latency Analysis

The average end-to-end latency from sensor reading to dashboard update was measured at 1.2 seconds. This latency encompasses sensor sampling time (approximately 60 ms), MQTT message propagation over Wi-Fi (approximately 80–120 ms), server-side processing and database write (approximately 300 ms), WebSocket push to the frontend (approximately 200 ms), and React rendering time (approximately 150 ms). This latency is well within acceptable bounds for a parking management use case, where subsecond precision is not required.

C. Prediction Performance

The Random Forest Classifier trained on 14 days of historical occupancy data achieved a prediction accuracy of 92.4% for 30-minute ahead forecasts. Feature importance analysis revealed that the hour of day was the most influential predictor, followed by day of week and rolling occupancy rate. The model was particularly accurate during predictable peak hours (morning 8–10 AM and evening 5–7 PM) and showed slightly reduced accuracy during irregular occupancy patterns on weekends.

D. Scalability Testing

Load testing demonstrated that the MQTT broker and MongoDB Atlas architecture could support up to 200 concurrent sensor nodes without measurable performance degradation. Response times remained below 2 seconds even at peak load, confirming that the architecture is well-suited for deployment in larger parking facilities.

E. User Feedback

Feedback collected from 35 pilot users indicated that approximately 60% reported a significant reduction in average parking search time compared to their experience with conventional parking systems in the same facility. Users highlighted the real-time color-coded slot map and the 30-minute prediction display as the most valuable features of the dashboard.

DOI: 10.48175/IJARSCT-34445

ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

VI. APPLICATIONS

The IoT-Based Smart Parking Prediction System has broad applicability across multiple urban and institutional contexts:

- Urban Parking Lots: Real-time slot availability and navigation guidance for shopping malls, hospitals, office complexes, and public transit hubs, reducing circling time and traffic congestion in surrounding roads.
- Smart Campus Parking: Automated monitoring and dynamic allocation of parking slots in university campuses and corporate parks, with integration into access control systems.
- Municipal Traffic Management: Integration with city-level traffic management systems to provide early warning of parking saturation in high-density zones, enabling traffic signal adjustments and dynamic routing.
- Paid Parking Automation: Dynamic pricing engines can leverage occupancy prediction data to implement surge pricing during peak hours and discounted rates during off-peak times, maximizing revenue and optimizing utilization.
- Disability Access Management: Priority slot tracking and advance reservation capabilities for differently-abled users, ensuring that accessible slots are available when needed.
- Event Venue Management: Temporary deployment at stadiums, conference centers, and outdoor venues for real-time monitoring during high-demand events.

VII. CONCLUSION

This paper presented an IoT-Based Smart Parking Prediction System that successfully integrates embedded ultrasonic sensing, NodeMCU microcontroller firmware, MQTT-based cloud communication, MongoDB Atlas persistent storage, and a Random Forest machine learning prediction engine into a cohesive, scalable solution for modern urban parking challenges.The pilot evaluation demonstrated 98.6% occupancy detection accuracy, 92.4% prediction accuracy for 30-minute forecasts, and an end-to-end latency of 1.2 seconds. These results confirm that the proposed system meaningfully improves the efficiency of urban parking infrastructure and has the potential for wide-scale deployment in smart city environments.

Future work will focus on the following enhancements: (i) integration of computer vision modules for license plate recognition and vehicle type classification; (ii) development of a mobile application with turn-by-turn navigation to available slots; (iii) replacement of the Random Forest model with Long ShortTerm Memory (LSTM) networks for improved long-horizon prediction; (iv) implementation of dynamic slot reservation and payment processing within the mobile application; and (v) deployment at scale across multiple facilities with federated learning to improve the global prediction model without centralizing sensitive data.

REFERENCES

[1] A. O. Kotb, Y. Shen, X. Zhu, and Y. Huang, "iParker — A New Smart Car-Parking System Based on Dynamic Resource Allocation and Pricing," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 9, pp. 2637–2647, Sep. 2016.

[2] P. Suresh, J. V. Daniel, V. Parthasarathy, and R. H. Aswathy, "A State of the Art Review on the Internet of Things (IoT) History, Technology and Fields of Deployment," in Proc. International Conference on Science, Engineering and Management Research (ICSEMR), Chennai, India, 2014, pp. 1–8.

[3] T. Lin, H. Rivano, and F. Le Mouël, "A Survey of Smart Parking Solutions," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 12, pp. 3229–3253, Dec. 2017.

[4] NodeMCU, "NodeMCU ESP8266 Documentation," [Online]. Available: https://nodemcu.readthedocs.io/. [Accessed: Apr. 2026].

[5] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

DOI: 10.48175/IJARSCT-34445

[6] MongoDB, Inc., "MongoDB Atlas Cloud Documentation," [Online]. Available: https://www.mongodb.com/docs/atlas/. [Accessed: Apr. 2026].

[7] G. J. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression," IEEE Signal Processing Magazine, vol. 15, no. 6, pp. 74–90, 1998.

[8] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," IEEE Internet of Things Journal, vol. 1, no. 1, pp. 22–32, Feb. 2014