

Intelligent Smart Parking Management Using Hybrid Deep Learning in IoT Cities

Raniah Hazim^{1*} 

Department of Computer Engineering, University of Diyala, Diyala, 32001, Iraq
Rania_hazim_eng@uodiyala.edu.iq

Abstract

Urbanization and the Internet of Things (IoT) are on the rise, and managing urban parking has become a difficult challenge for existing smart parking systems, as they have failed to manage traffic effectively, lack security, have low user adaptability, inaccurately predict parking demand, store unsecured data, create unfair pricing, and suffer from improper parking detection. This paper introduces a new hybrid deep learning-based Artificial Intelligence (AI) smart parking management system for the IoT city. We design a simulator environment and add 10 vehicles to simulate real-time data input. User data privacy is secured with Elliptic Curve Cryptography (ECC) and Local Differential Privacy (LDP). Dynamic pricing is formulated as a Stackelberg Leader-Follower Game Framework (SL-FGF) for optimization. At the same time, we propose the Acar Agent Multi-Criteria Decision-Making (AcarA-MCDM) framework for intelligent space allocation. A hybrid CNN-LSTM performs parking slot prediction with the Lion Optimization Algorithm (CNN-LSTM-LOA), and an Intelligent Parking Reservation System (IPRS) is deployed. Mis-parking detection is based on an autoencoder with a GPS Navigation system to ensure proper space utilization. Evaluation of the system on 10 performance metrics yielded the best results, i.e., detection accuracy of 98.5%, precision of 99.2%, recall of 97.3%, F1-score of 98.3%, allocation efficiency of 95.5%, latency of 57 ms, and energy consumption of 99 mJ, establishing significant improvements over state-of-the-art methods.

Keywords: Smart Parking, Deep Learning, IoT Cities, hybrid CNN-LSTM, Vehicle Allocation

Article history: Received: 1 Jul 2026, accepted: 23 Aug 2026, published: 15 Sep 2026.

This article is open-access under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Rapid urbanization has significantly increased the number of vehicles in metropolitan areas, creating considerable challenges for urban transportation infrastructure, particularly parking management [1]. Limited parking availability often increases traffic congestion, lengthens parking search times, raises fuel consumption, and increases carbon emissions, undermining the sustainability of modern cities [2] and [3]. Most existing parking management solutions rely on manual monitoring or basic sensor networks, which fail to efficiently adapt to the dynamic, real-time demands of urban mobility [4]. Consequently, Smart Cities have increasingly adopted

Internet of Things (IoT) technologies to develop intelligent parking systems that collect, transmit, and process real-time parking information through interconnected sensors, actuators, and communication networks [5] and [6].

The continuous generation of large volumes of parking data has created new opportunities for intelligent data analytics, enabling parking management to evolve from traditional reactive systems to proactive, intelligent decision-making frameworks [7,8]. However, parking occupancy prediction remains a challenging task because parking demand is highly nonlinear and varies with traffic conditions, time of day, weather, public events, and driver

behavior [9]. Deep learning techniques have demonstrated superior capability in learning complex spatio-temporal parking patterns, where Convolutional Neural Networks (CNNs) effectively extract spatial characteristics and Long Short-Term Memory (LSTM) networks capture temporal dependencies. Consequently, hybrid CNN-LSTM architectures have attracted considerable attention because they combine the strengths of both models to improve parking occupancy prediction accuracy under dynamic urban environments [10–17].

Although intelligent parking management has advanced significantly, existing studies generally address only individual aspects of the parking problem. Several studies focus on parking occupancy prediction, while others emphasize parking allocation, reservation management, pricing mechanisms, vehicle detection, or data security independently [18–35]. However, practical smart parking environments require these functionalities to operate collaboratively within a unified framework. The lack of an integrated architecture often limits prediction accuracy, resource utilization, parking allocation efficiency, pricing adaptability, and user privacy, thereby reducing the overall effectiveness of smart parking management systems.

Furthermore, several important research challenges remain unresolved in the current literature. First, many existing systems do not adequately preserve user privacy during data collection and communication, exposing sensitive parking information to potential security threats [36]. Second, dynamic parking pricing strategies are often implemented without jointly considering real-time parking demand and occupancy variation, resulting in inefficient resource utilization and reduced revenue optimization [37]. Third, several parking prediction models rely primarily on occupancy information while neglecting influential contextual factors such as roadside parking conditions and traffic congestion [38,39]. Finally, to the best of the authors' knowledge, existing intelligent parking frameworks do not integrate secure privacy preservation, dynamic pricing, intelligent parking allocation, hybrid deep learning-based prediction, reservation management, and mis-parking detection into

a single optimization framework that can address multiple parking management objectives concurrently [40].

To overcome these research limitations, this study proposes an intelligent smart parking management framework that integrates multiple complementary technologies into a unified architecture. The framework combines Elliptic Curve Cryptography (ECC) and Local Differential Privacy (LDP) to preserve user privacy and secure data communication. A Reinforcement Learning (RL)-based Stackelberg Leader–Follower Game Framework (SL-FGF) dynamically optimizes parking prices based on real-time parking demand. The AcarAgent Multi-Criteria Decision-Making (AcarA-MCDM) framework performs intelligent parking allocation. At the same time, a hybrid CNN-LSTM model optimized with the Lion Optimization Algorithm (LOA) predicts parking occupancy with improved accuracy. In addition, an Intelligent Parking Reservation System (IPRS) integrated with Autoencoder-based mis-parking detection and GPS navigation enhances parking compliance and efficient space utilization.

The primary novelty of this work lies in unifying privacy preservation, dynamic pricing, intelligent decision-making, deep learning-based parking prediction, parking reservation, and mis-parking detection within a single optimization framework, rather than treating these components as independent modules. By enabling these techniques to operate collaboratively, the framework simultaneously enhances prediction accuracy, parking allocation efficiency, user privacy, resource utilization, and overall smart parking performance. The major contributions of this work are summarized as follows:

1. A unified AI-enabled smart parking management framework integrating ECC-LDP privacy preservation, RL-based dynamic pricing, AcarA-MCDM intelligent parking allocation, CNN-LSTM-LOA parking prediction, and Autoencoder-based mis-parking detection.
2. A secure privacy-preserving mechanism that combines ECC and LDP to protect sensitive user information in IoT-enabled smart parking environments.

3. A Reinforcement Learning-based Stackelberg Leader–Follower Game Framework for adaptive real-time parking price optimization.
4. A hybrid CNN-LSTM model optimized using the Lion Optimization Algorithm for accurate parking occupancy prediction.
5. An intelligent parking reservation and mis-parking detection mechanism that improves parking utilization and traffic management.
6. A comprehensive experimental evaluation using multiple performance metrics to demonstrate the effectiveness of the approach compared with existing approaches.

The remainder of this paper is organized as follows. Section 3 presents the related work. Section 4 describes the proposed smart parking framework and mathematical formulations. Section 5 presents the experimental evaluation and discussion. Finally, Section 6 concludes the paper and outlines future research directions.

2. Related work

Vakrinou et al. [21] use LSTM and Transformer models to predict off-street parking occupancy, with LSTM performing better in short-term prediction and Transformer in long-term prediction.

The problem is that this work did not use on-street data, which account for a larger portion of parking and influence off-street occupancy. Grbić and Koch [22] introduce an Automatic Parking Slot Detection and Occupancy Classification algorithm using ResNet34 with the PKLot and CNRPark-EXT datasets. However, it did not sufficiently address the spatial dependencies between parking slots. Yuldashev et al. [23] achieve 98.01% accuracy with an optimized MobileNetV3 model for occupancy detection. However, performance degrades with different weather conditions and partial occlusions.

Dahiya et al. [24] evaluate various ML algorithms for real-time IoT-enabled parking prediction. However, their

approach is limited by the dependency on a single dataset, which may not represent the diversity of parking scenarios. Rajyalakshmi and Lakshmana [25] propose an IoT-based smart parking system with face recognition and automatic vehicle identification. The study is limited by sensor reliability and poor network infrastructure in certain environments. Kadusić et al. [26] discuss integrating NB-IoT with 5G networks for smart parking solutions. However, they do not address scalability, interoperability, or AI-based decision-making.

Amato et al. [27] present a CNN-based camera-only automated parking management framework with unsatisfactory full coverage of large or complex parking lot areas. Shaikh et al. [28] propose an IoT-integrated smart parking solution for Bagging Regression with a 0.06% MAE. Still, it is restricted to parking behavior prediction only. Kotb et al. [29] introduce seven algorithms for a fair parking lot reservation algorithm with 2430 combinations of preference criteria but without meta-heuristic reinforcement to avoid local optima. Biscardi et al. [30] describe a Social IoT roadside parking solution using magnetometer sensors without range coverage beyond WiFi limits.

Xu et al. [31] present a hybrid SVM-ANN trust management framework, which has not yet been optimized for computational efficiency. Ismail et al. [32] describe an ATM system using the DBSCAN clustering method, which still needs real-world validation.

Pramanik et al. [33], Raza et al. [34], and Acharya et al. [35] present IoT-based and vision-based parking detection systems that lack advanced parking reservation, integrated payment, and adverse-weather condition robustness features. Table 1 compares relevant works in this area.

Table 1: Summary of literature survey

Authors	Objective	The model used or the algorithm	Limitations
Vakrinou et al. [21]	To forecast off-street parking occupancy.	LSTM, Transformer, SHAP	- Issues with off-street data focus. - Problems with model transferability.
Grbić and Koch [22]	To propose an algorithm for automatic parking slot detection.	Vehicle detection, ResNet34	- Issues with unreliable vehicle detection distribution. - Problems with limited spatial-relation exploration.

Yuldashev, Y., et al. [23]	To detect vehicle occupancy.	Optimized MobileNetV3	- Challenges with varied weather and partial obstructions. - Issues with hidden car areas in images.
Dahiya et al. [24]	To build a model for real-time vehicle parking prediction.	Multiple ML algorithms	- Problem with single dataset usage. - Issues with scope for broader data sources.
Rajyalakshmi and Lakshmana [25]	To design an IoT-based smart car parking system.	IoT integration	- Major limitations in sensor precision and facial recognition reliability. - Environmental factors affecting the system.
Kadusic, E., et al. [26]	To provide a smart parking solution using NB-IoT.	NB-IoT technology	- Limitations in scalability and interoperability. - Lack of AI integration for decision-making.
Alsheikhy, A. et al [27]	To propose an intelligent parking management system.	CNN	- Limitations of single-camera usage for large areas. - Issues with lighting and camera placement impacts.
Tekouabou, et al. [28]	To suggest an efficient IoT-integrated solution for smart parking.	Bagging Regression	Limited to predicting parking behavior only.
Jemmali, M, et al. [29]	To construct a framework for fair allocation of parking lots.	Seven different algorithms	- Limitation of scope without an expanded class list or comparative analysis. - Problems with algorithms not reinforced with meta-heuristics.
Floris, A, et al. [30]	To propose an innovative smart roadside parking solution.	Social IoT, magnetometer sensors	- Scalability challenged by WiFi range. - Sensor accuracy issues in diverse urban settings.
Ali, J., & Khan, M. F. [31]	To describe a hybrid trust management technique for system security.	SVM and ANN	Issues in energy consumption.
Lilhore, U. et al. [32]	To develop an Adaptive Traffic Management (ATM) system.	DBSCAN clustering	- Reliance on a simulator. - Issues with energy efficiency and security.
Zahid, A., et al. [33]	To develop an IoT-enabled vacant parking slot detection system.	UHF RFID tags	- Absence of advanced features like pre-booking. - Lack of detailed parking area maps.
Dheeven, T. A, et al. [34]	To address urban congestion using sensors.	ESP8266 WiFi module, sensors	- Reliance on a web page display. - Scope limited to basic slot detection.
Satyanath, G., et al. [35]	To investigate dehazing networks for parking space classification.	Dehazing networks	- Real-time deployment complexity not detailed. - Performance under other adverse weather conditions beyond haze not explored.
Alsaedi, N., & Jalal, A. S. A. [36]	Automated parking system with reduced latency and network usage.	Reinforcement Learning (RL), Q-learning, image processing, altered Internet of Spatial Things (IoST) framework.	Lacks data privacy measures for cloud storage.
Lyu, M., et al. [37]	Predict block-level parking occupancy.	Deep-learning model, Boruta, LSTM.	No consideration for parking price changes.
Ali, G., et al. [38]	Predict available parking space.	Deep LSTM-based framework, IoT, cloud computing, sensor networks.	- Excludes influencing factors beyond occupancy data ; neglects roadside parking and traffic congestion.
Canli, H., & Toklu, S. [40]	Deep learning mobile smart parking application.	Deep learning mobile smart parking application, LSTM-based model on Cloud Computing.	Does not consider parking's influence on traffic volume.
Aditya, A., et al. [41]	IoT-based solution for parking availability.	IoT-based solution, Intelligent Parking System	- No direct guidance/reservation keeping - Improper parking detection is lacking.

3. System formulation & Proposed Work

The proposed smart parking management framework integrates privacy preservation, dynamic pricing, personalized space allocation, short-term occupancy prediction, and mis-parking detection into a unified pipeline, as illustrated in Fig. 1. Initially, the ECC-LDP privacy preservation module securely processes user requests to protect sensitive information. The processed data are then used by the reinforcement learning-based Stackelberg Leader-Follower Game Framework (SL-FGF) to determine dynamic parking prices based on real-

time demand. Subsequently, the AcarAgent Multi-Criteria Decision-Making (AcarA-MCDM) module identifies the most suitable parking space. The hybrid CNN-LSTM model optimized using the Lion Optimization Algorithm (LOA) predicts parking occupancy. At the same time, the Intelligent Parking Reservation System (IPRS) manages parking reservations. Finally, the Autoencoder-based mis-parking detection module identifies parking violations and provides GPS-based navigation assistance for corrective actions.

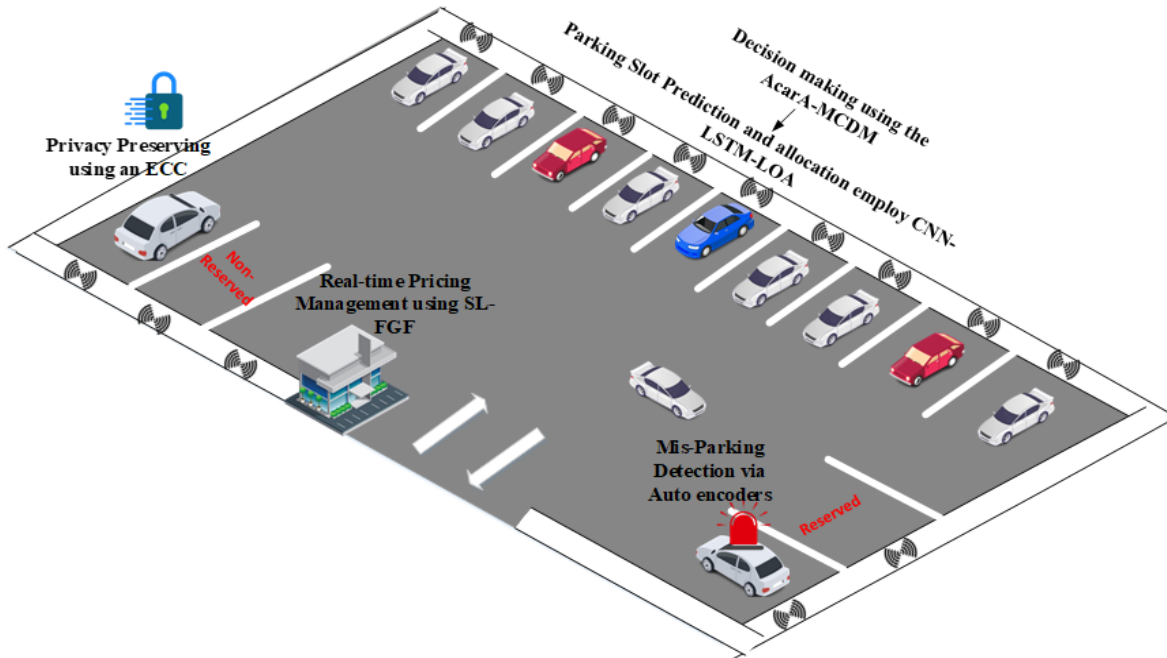


Fig. 1 Overall Architecture of this Research

Global system optimization model: To mathematically link these distinct operational components into a single unified framework, the dynamic global system utility Ψ_{total} at time step t is formulated as:

$$\Psi_{total}(t) = \max \left(\sum_{j \in J} occupancy(j, t) \cdot p_j^t - \sum_{i \in N} \Phi_{j,od}^{l,n}(i) - \lambda \cdot E(x) \right) \quad (1)$$

Where, $occupancy(j, t) \in [0, 1]$ represents the predicted occupancy state of the parking facility j at time t , forecasted by the CNN-LSTM-NILA prediction module.

p_j^t is the dynamic pricing rate for the facility j determined by the SL-FGF game-theoretic framework.

$\Phi_{j,od}^{l,n}(i)$ denotes the multi-attribute user disutility (monetary cost, walking distance, user health constraints) evaluated by the AcarA-MCDM allocation module for driver i assigned to spot j .

$E(x)$ represents the reconstruction error calculated by the autoencoder anomaly detection module for vehicle positioning

λ is a dynamic penalty scaling weight applied to improperly parked or unauthorized vehicles.

The above equation serves as the master objective function, ensuring facility revenue is maximized, user

disutility is minimized, and mis-parking anomalies are penalized under strict differential privacy guarantees.

3.1 Car Parking Simulation and Data Collection

In the parking simulation, the process starts by setting up the virtual parking area, drawing boundaries, slots, and entry/exit points. Vehicles are then drawn up with specific characteristics: size and behaviour. Pathfinding enables them to follow their given paths to the parking area efficiently. While running the simulation, vehicle movements will be recorded, and collision detection will be implemented to avoid accidents. Performance metrics such as the time taken to park and parking efficiency will be collected for analysis. Based on these results, real-world parking systems can be optimized, with useful insights available for further improvement. Here, we will collect simulation data, such as the vehicle's position in the parking lot over time.

3.2 Privacy Preservation

Following simulation data collection, the next stage addresses privacy preservation within the smart parking framework. The Key Distribution Center (KDC) forms the foundation of this framework by establishing the operational infrastructure, registering drivers, and generating the cryptographic parameters and keys needed to ensure secure operation. The KDC maintains central information processing without centralized control of user information in the actual parking management process. However, to further protect sensitive data and maintain user privacy, we include layers of Local Differential Privacy (LDP) to enhance both. LDP ensures that specific data is anonymized, thus restricting adversaries from tracking and generating personal information. The key principle of LDP is covered in the equation below: a randomized algorithm L ensures local ϵ -differential privacy if, for every two different data input elements u and u' in the dataset, and any realization u^*

$$\Pr[L(u) = u^*] \leq e^\epsilon \Pr[L(u') = u^*] \quad (2)$$

Here, ϵ is the privacy budget; smaller ϵ values indicate greater privacy but can yield lower data utility. Furthermore, Elliptic Curve Cryptography (ECC) secures communications throughout the system. As one of the most efficient public-key methods, ECC provides a high level of security with smaller key sizes, making it well

suited to the constrained resources of IoT devices. Algorithm 1 defines the privacy-preserving scheme. The points on the elliptic curve must satisfy these fundamental equations:

$$4a^3 + 27b^2 \neq 0 \quad (3)$$

$$\text{ECp}(a, b): y^2 = x^3 + ax + b(\text{mod}P) = 0 \quad (4)$$

Algorithm 1: Privacy-preserving.

Input: User data U , privacy budget ϵ , ECC key pair, network status, strong type.
Output: Secure parking dataset U' , Encrypted communication keys, and privacy-preserved parking information.

1. Initialize
2. KDC-generated ECC keys
3. **If** user data contains PII
4. Apply LDP with budget using $\Pr[L(u) = u^*] \leq e^\epsilon \Pr[L(u') = u^*]$
5. **Else**
6. -Skip perturbation
7. **End if**
8. **If** network==untrusted:
9. Secure with HMAC:
End if
10. **If** storage==IOTA_{DLT}
11. Store anonymized data
Else
 -use local encrypted DB
End if
12. Return Anonymized Data

Personally Identifiable Information (PII). In addition to ECC and LDP, the system employs a mutual authentication mechanism based on Hash-based Message Authentication Code (HMAC) to verify the authenticity of users and parking providers. These combined techniques - LDP, ECC, and HMAC- along with the immutability and scalability offered by IOT. A Distributed Ledger Technology (DLT) collectively ensures the privacy, integrity, and security of all data exchanged within the Key Management System (KMS), thereby establishing a trust-based Smart Parking Network.

3.3 Real-time Pricing Management

Once the system is secure, we focus on dynamic pricing management, another key element of the smart parking system. Traditional static pricing methods often do not

suit fluctuating demand for parking spaces; as a result, space utilization is inefficient, and revenue is not optimized. To overcome this disadvantage, we introduce RL algorithms developed using a Stackelberg leader-follower game framework (SL-FGF). In real time, RL enables the system to continuously adapt parking prices by learning current demand and parking-lot occupancy.

Thus, it can optimally adjust prices by matching demand with available space, using resources efficiently while maximizing revenue. The system can raise prices when periods of increased demand are predicted and lower them during periods of lower demand to remain competitive and attract more drivers. The integration of RL ensures that parking pricing remains flexible and responsive to real-time conditions, offering a pricing strategy that balances the economic sustainability of the parking system with user convenience.

3.3.1 Upper-Level Problem (Parking Agency's Objective Function)

In the context of dynamic pricing for our smart parking system, the process after privacy preservation focuses on optimizing revenue and space utilization, primarily through Reinforcement Learning (RL) algorithms developed using a Stackelberg leader-follower game framework. The parking agency, acting as the leader, sets prices to achieve its objectives, which the following objective function can define:

$$\min \beta \sum_{t,j} |f_j^t - K_j c_j| + \alpha_s \sum_{t,j} (\text{economic surplus}) + \alpha_r \sum_{t,j} (\text{revenue}) \quad (5)$$

Where f_j^t is the occupancy of the parking area j at time t , K_j is the target percentage occupancy, c_j is the capacity, and $\beta, \alpha_s, \alpha_r$ are weights for the respective objectives.

3.3.2 Upper-Level Problem (Constraints)

These pricing decisions are subject to constraints such as price variation limits,

$$-\epsilon_l \leq p_j^t - p_j^{t-1} \leq \epsilon_r \quad (6)$$

and the price bounds equations are given below,

$$\gamma_j \leq p_j^t \leq \eta_j \quad (7)$$

Where p_j^t is the price at time t , and $\epsilon_l, \epsilon_r, \gamma_j, \eta_j$ are the respective limits. The system also adheres to the conservation of parking spaces, given by

$$f_j^t = f_j^{t-1} - g_j^t + \sum_{n \in \{1,2,\dots,N\}} q_j^{t,n} \quad (8)$$

Where g_j^t represents departing users, and $q_j^{t,n}$ represents arriving users. The total demand arriving and demand leaving in eq. (8) and (9)

$$q_j^{t,n} = \sum_{o \in O} \sum_{d \in D} h_{j,od}^{t,n} \quad (9)$$

$$g_j^t = \sum_{m=\max(1,t-N)}^{t-1} q_j^{m,t-m} \quad (10)$$

With $h_{j,od}^{t,n}$ being the number of users from origin o to destination d choosing area j for duration n .

3.3.3 Lower-Level Problem (User Disutility and Decision)

Simultaneously, users (the followers) make decisions based on minimizing their disutility, defined as

$$\phi_{j,od}^{l,n} = n \cdot p_j^t + \theta w_{jd} + \theta' v_{oj} \quad (11)$$

Where n is the parking duration, w_{jd} is walking time, v_{oj} is travel time, and θ, θ' are value-of-time coefficients. There is also a disutility for a dummy lot (ζ).

$$\lambda_{\zeta,od}^{t,n} = \min \left\{ H^{-1}(0), \max_{j \in J} \{ n[p_j^{t-1} + (t+1-t_s)\epsilon_r] + \theta w_{jd} + \theta' v_{oj} \} \right\} \quad (12)$$

Users ultimately choose a parking area by minimizing $\min_{z_l} \sum_{j \in J \cup \{\zeta\}} \phi_{j,ad}^{l,n} \cdot z_{jl}$, subject to

$$z_{jl} \in \{0,1\} \sum_{j \in J \cup \{\zeta\}} z_{jl} = 1 \quad (13)$$

And

$$z_{jl} + \sum_{-l \in L_{od}^{t,n}} z_{j,-l} + f_j^{t-1} - g_j^t \leq c_j \quad (14)$$

Where z_{jl} indicates user choice, and $L_{od}^{t,n}$ is the set of relevant users. Through these equations, the RL algorithms learn to dynamically adjust prices by matching demand with available space, optimizing resource utilization, and maximizing revenue in real time. Algorithm 2 indicates the Stackelberg leader-follower game framework (SL-FGF).

Algorithm 2: Stackelberg leader-follower game framework (SL-FGF).

Input: Parking demand D_t , available parking space P_t , parking price bounds, maximum episodes N .

Output: Optimal parking price, optimal parking allocation policy.

1. Initialize:

- Set price bounds (Eq. 5-6):

- RL agent (Leader) with Q-table

2. while not converged:

3. Parking Agency (Leader):

4. Observe current demand and occupancy

```

5.      if
6.          Increase price using RL (Eq. 4):
7.      else if
8.          Decrease price to attract users
9.      else
10.         Maintain current price
11.     end if
12.     Enforce Constraints (Eq. 5-6)
13. Users (Followers):
14.     for each user
15.         Calculate disutility
16.         if
17.             Accept Parking Slot
18.         else
19.             Reject and seek alternatives
20.         end if
21.     end for
22. Check convergence
23.     if
24.         Converged = True
25.     end if
26. end while
27. Return dynamic prices
    
```

3.4 Decision Making

After pricing optimization comes the planning inside the system. A new approach combines AcarAgent with Multi-Criteria Decision Making (AcarA-MCDM) methods. The Multi-Criteria Decision Making (MCDM) framework evaluates multiple parking space bids based on various criteria, aiming to select the optimal parking solution for a user. This approach addresses drivers who feel overwhelmed by information and lack organized guidance when finding suitable parking. The framework

considers several key criteria for parking space selection, including:

- **Traffic situation: encompassing the driving distance to the parking lot and estimated time of arrival.**
- **Availability: the real-time status of parking spaces.**
- **Cost: the parking fees.**
- **Walking distance: from the parking space to the driver's final destination.**
- **Driver's physical condition: a novel criterion introduced to prioritize vulnerable drivers.**

To objectively determine the weight of each criterion in the decision-making process, methods such as the CRITIC method can be used.

3.4.1 Driver's Physical Condition (Health Criterion):

The driver's physical condition can be assessed using a predicted 6-minute walking distance (6MWD), calculated based on anthropometric and demographic data, as in the following regression equation

$$6MWD_{pred} = -(age) + (gender) - (BMI) \quad (15)$$

Where the $6MWD_{pred}$: Predicted 6-minute walking distance; age: Driver's age; gender: Driver's gender (1 for man, 0 for woman); BMI: Body Mass Index. This score helps classify drivers into profiles based on bodily fragility, allowing the system to prioritize parking allocations for more vulnerable individuals. Fig. 2 depicts the Smart Parking Allocation.

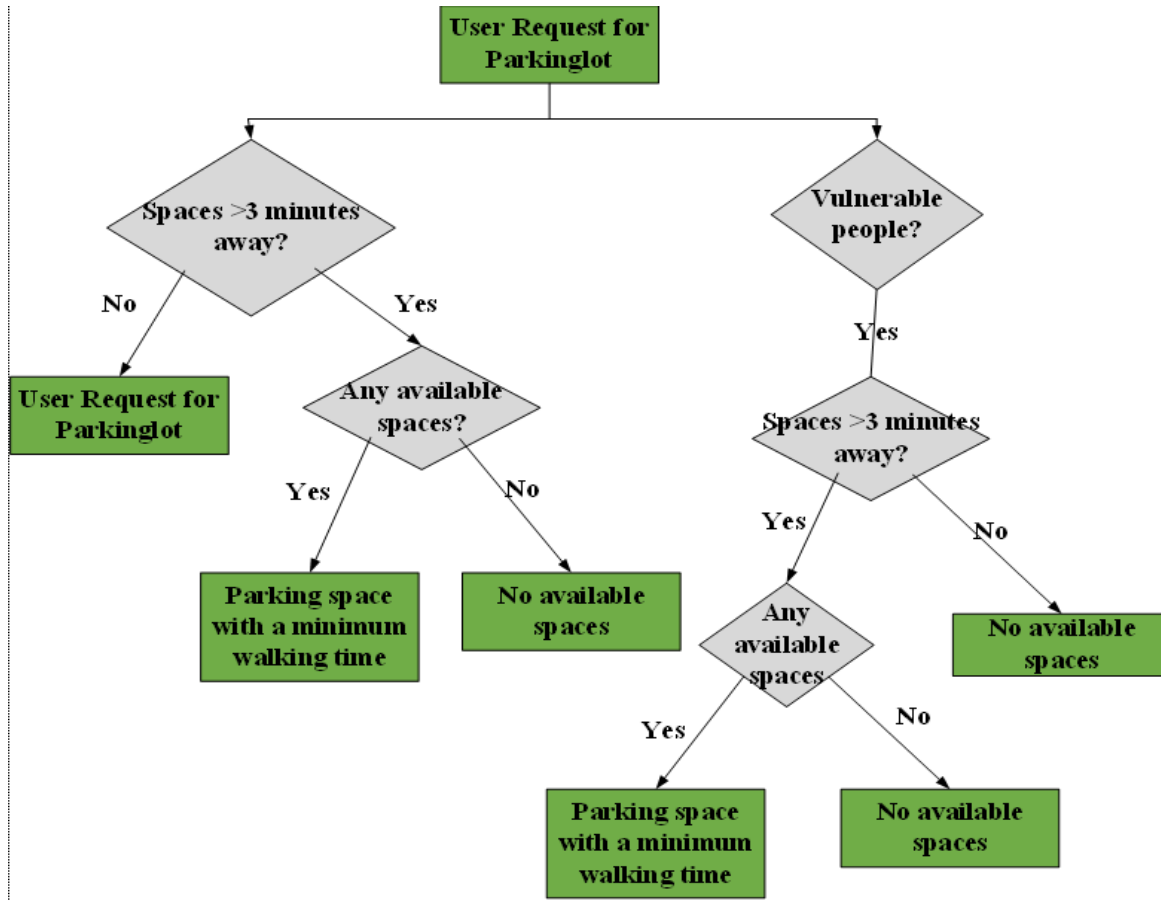


Fig. 2 Smart Parking Allocation

3.4.2 AcarAgent and Multi-Agent System (MAS) Framework

The "AcarAgent" concept aligns with the "carAgent" within a Multi-Agent System (MAS), where intelligent agents represent real-world entities like vehicles ("carAgents") and parking spaces ("parking Agents"). This decentralized approach offers advantages in managing complex problems like parking allocation. In this framework, a "carAgent" acts as an intelligent negotiator. It initiates a negotiation process with "parking

Agents" using a protocol such as the Contract Net Protocol (CNP). The "carAgent" sends a call-for-proposals message (CFP) to parking agents, detailing the car's characteristics and destination. Parking agents then respond with proposals. The "carAgent" evaluates these proposals using a utility function designed to optimize its objective. This function prioritizes proposals that minimize the distance from the parking space to the passenger's destination and ensure proximity to the car's current location. Fig. 3 indicates the multi-agent system.

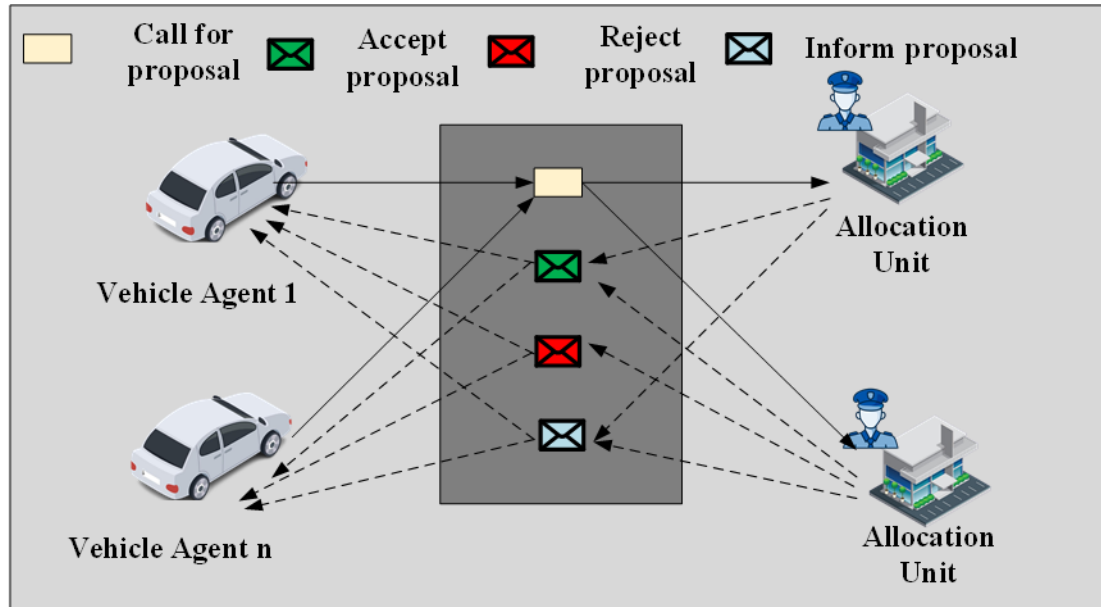


Fig.3. Multi-agent system

i. General Objective Function for Parking Allocation in MAS:

A general objective function for optimal parking assignment in a multi-agent system, aiming to minimize a weighted sum of distances, can be represented as:

$$\text{Min} \sum_{i=1}^n \sum_{j=1}^n (w_1 \cdot d_{ai,pj} + w_2 \cdot d_{pj,ai}) \cdot x_{ij} \quad (15)$$

Where the w_1, w_2 are the Weights for the distances.
 $d_{ai,pj}$: Distance from car agent i to parking space j .
 $d_{pj,ai}$: Distance from parking space j to car agent i .
 x_{ij} : Binary variable indicating if car agent i is assigned to parking space j . The utility function of the "carAgent" is designed to promote proposals that minimize the distance from the parking space to the passenger's destination and are closer to the cars.

ii. Hybrid Functionality (AcarA-MCDM)

The dynamic system integration brings together the intelligent negotiation capabilities of the "AcarAgent" (Multi-Agent System) with the comprehensive evaluation framework of Multi-Criteria Decision Making (MCDM). In this hybrid model, after the real-time pricing management phase, the system initiates planning. When a "carAgent" (representing a driver) needs a parking space, it communicates with available "parking Agents". The "carAgent" evaluates the proposals it receives based on more than distance or cost. Instead, the MCDM framework provides the structured criteria and their

weighted importance that the "carAgent" uses in its internal utility function. For example, the "carAgent's" utility function, in addition to minimizing distance to destination and current car proximity, would incorporate the various criteria defined in the MCDM framework, such as:

- Cost of parking **pjt** from pricing management.
- Walking distance **from the parking to the final destination.**
- Availability **of the parking spot.**
- The driver's physical condition (**derived from the6MWDpred equation**).

The "carAgent" would perform a real-time, personalized MCDM analysis on the bids from "parking Agents". The weights of these criteria, potentially determined by methods like CRITIC (as mentioned in the MCDM section), would guide the "carAgent's" decision. This dynamic allocation, guided by the MCDM's multi-faceted evaluation within the MAS negotiation, improves the likelihood of accurate parking predictions, reduces driver search time, and alleviates traffic congestion by ensuring more efficient and personalized parking space utilization. This integration allows the system to dynamically allocate parking spaces based on real-time

demand, improving parking-area utilization and facilitating better traffic flow across the city.

3.5 Parking Slot Prediction and Allocation

3.5.1 CNN-LSTM

In LSTM neural networks, $C(t)$ is the next long-term state, determined along with the short-term state $S(t)$, based on three main components: the input gate, output gate, and forget gate. Each gate helps the model control what information to put into the cell. The forget gate works by:

$O_{(t)} = \sigma(W_{ko}k_{(t)} + W_{so}S_{(t-1)} + W_{co}C_{(t)} + b_{(o)})$	(16)
$i_{(t)} = \sigma(W_{ki}k_{(t)} + W_{si}S_{(t-1)} + W_{ci}C_{(t-1)} + b_i)$	(17)
$f_{(t)} = \sigma(W_{kf}k_{(t)} + W_{sf}S_{(t-1)} + W_{cf}C_{(t-1)} + b_f)$	(18)

In an LSTM network, the current input state (x_t) and the output from the previous state (h_{t-1}) are crucial. The previous cell state (C_{t-1}) is also considered, along with various biases (as detailed in Eqs. 16-19). The cell state is the central component of information flow, allowing information to pass through largely unchanged. Information can then be added to or discarded from the cell using sigmoid gates. Each gate in an LSTM operates as a layer or a series of matrix multiplies, and each gate can have its own set of weights specific to the memorization process. The forget gate (f_t) receives the previous time's output (h_{t-1}) and the current time's input (x_t) as its inputs, and it generates output as in Eq. (17), where (W_f) is the weight of the forget gate, and b_f is the bias. Eq (16)-(19) receive the previous time's output value and the current time's input value to calculate the output value and candidate cell state of the input gate. The input gate has weights. (W_i, W_c) and biases (b_i, b_c) to update

the recent cell state C_t . Eq. (18) shows that at time t , the output gate (o_t) uses the output (h_{t-1}) and input (x_t) as inputs to create the output. Here, W_o signifies the output gate weight, and b_o is the output gate bias. Finally, the LSTM's final value is determined from Eq. (20) after computing the output gate and the cell state. The symbol σ consistently denotes the sigmoid function throughout these operations.

$C_{(t)} = f_{(t)}C_{(t-1)} + i_{(t)} \tanh(W_{kc}k_t + W_{ac}S_{(t-1)} + b_{(c)})$	(19)
$S_{(t)} = O_{(t)} \tanh(C_{(t)})$	(20)

A CNN-LSTM model can predict parking occupancy percentage based on both dynamic provincial conditions and long-term trends. This model uses multilayer feed-forward neural networks to analyze enduring patterns alongside time-value patterns in parking occupancy. LSTM cells can represent short-term dependencies in parking occupancy. The novel approach begins by creating subsequences of parking occupancy parameters. This subseries is then sequentially fed into an LSTM model for processing, initially using a CNN model to partition the input sequences into subseries. Parking occupancy time-series data samples are partitioned into four-step input samples and one output sample. Each sample is broken into two sub-samples, which effectively allows multiple cycles per sample. An LSTM network takes each down-sampled two-time-step subsequence, which a time series can grow, for a CNN model to extract meaning from. This process contours the input data into different samples, subsequences, time steps, and features. Fig. 4 shows the CNN-LSTM architecture.

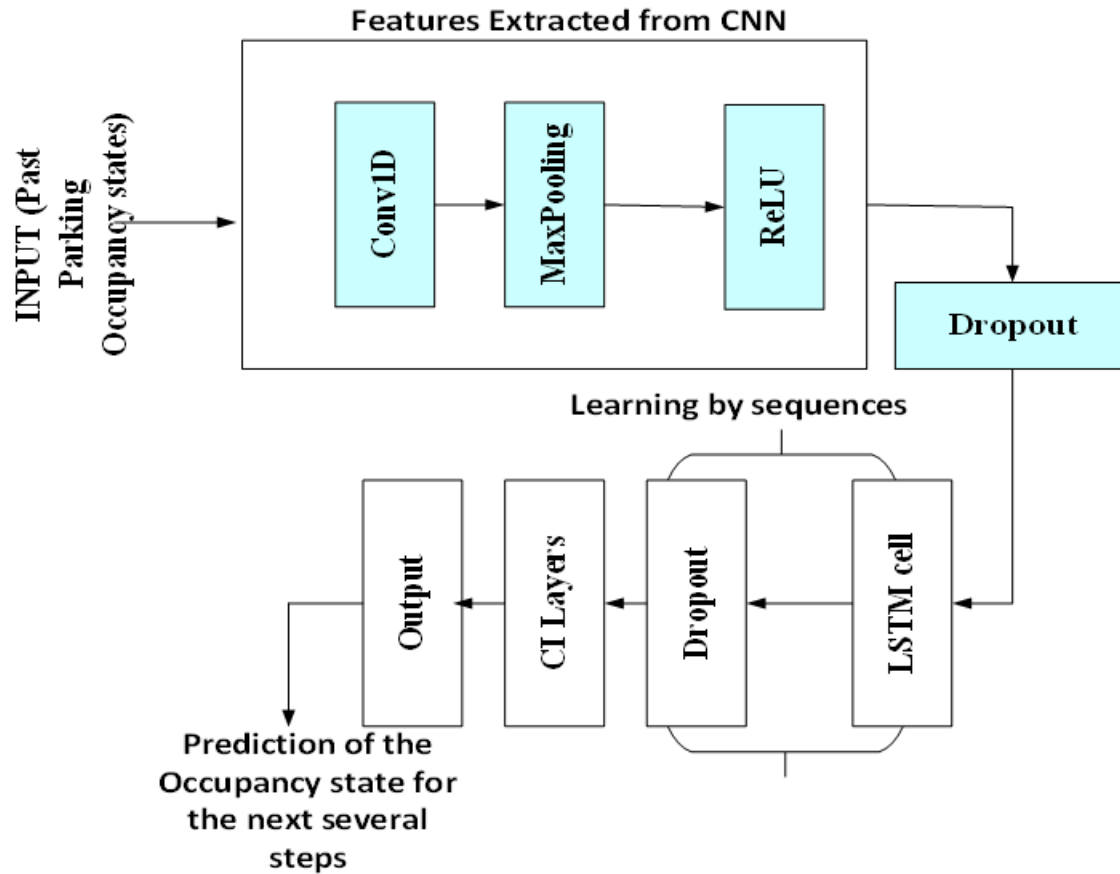


Fig. 4. Architecture of CNN-LSTM

3.5.2 Lion Optimization Algorithm

The optimization uses the Lion Optimization Algorithm (LA), specifically improved with Niche Immunity (NILA), to search for the optimal weights and thresholds of a Convolutional Neural Network. This optimization aims to minimize an objective function, which in our context would relate to the prediction error of the CNN's parameters:

$$\min f(x_1, x_2, \dots, x_n) \quad (21)$$

The overall fitness or value of the lion population ($f(\psi^{\text{pride}})$) during the optimization process is calculated based on the fitness of male lions, lionesses, and their cubs:

$$f(\psi^{\text{pride}}) = \frac{1}{2(1 + \|\psi^{\text{m_cub}}\|)} \left(f(\psi^{\text{m}}) + f(\psi^{\text{f}}) + \frac{\text{age}_{\text{mat}}}{\text{age}_{\text{cub}} + 1} \cdot \sum_{k=1}^{\|\psi^{\text{m_ab}}\|} \frac{f(\psi_k^{\text{m_cub}}) + f(\psi_k^{\text{f_cub}})}{\|\psi^{\text{m_cub}}\|} \right) \quad (22)$$

Furthermore, Niche Immunity is utilized to restrict over-duplication of similar individuals, ensuring population diversity by determining the clone number. (M_j) for each lion:

$$M_j = M_{\max} \times \left(1 - \frac{\rho_j}{\sum_{j=1}^N \rho_j} \right) \quad (23)$$

Additionally, an Intelligent Parking Reservation System (IPRS), interlaced with agent-based simulations, seeks to optimize space allocation and minimize search time. This system aims to maximize social benefits by minimizing total user travel cost, which can be represented as:

$$\min \sum_{\text{users}} (C_{\text{driving}} + C_{\text{parking}} + C_{\text{walking}}) \quad (24)$$

Concurrently, the system seeks to maximize the total parking lot revenue, formulated as:

$$\max \sum_{j \in J} \sum_{t \in T} \text{Occupancy}(j, t) \cdot \text{price}(j, t) \quad (25)$$

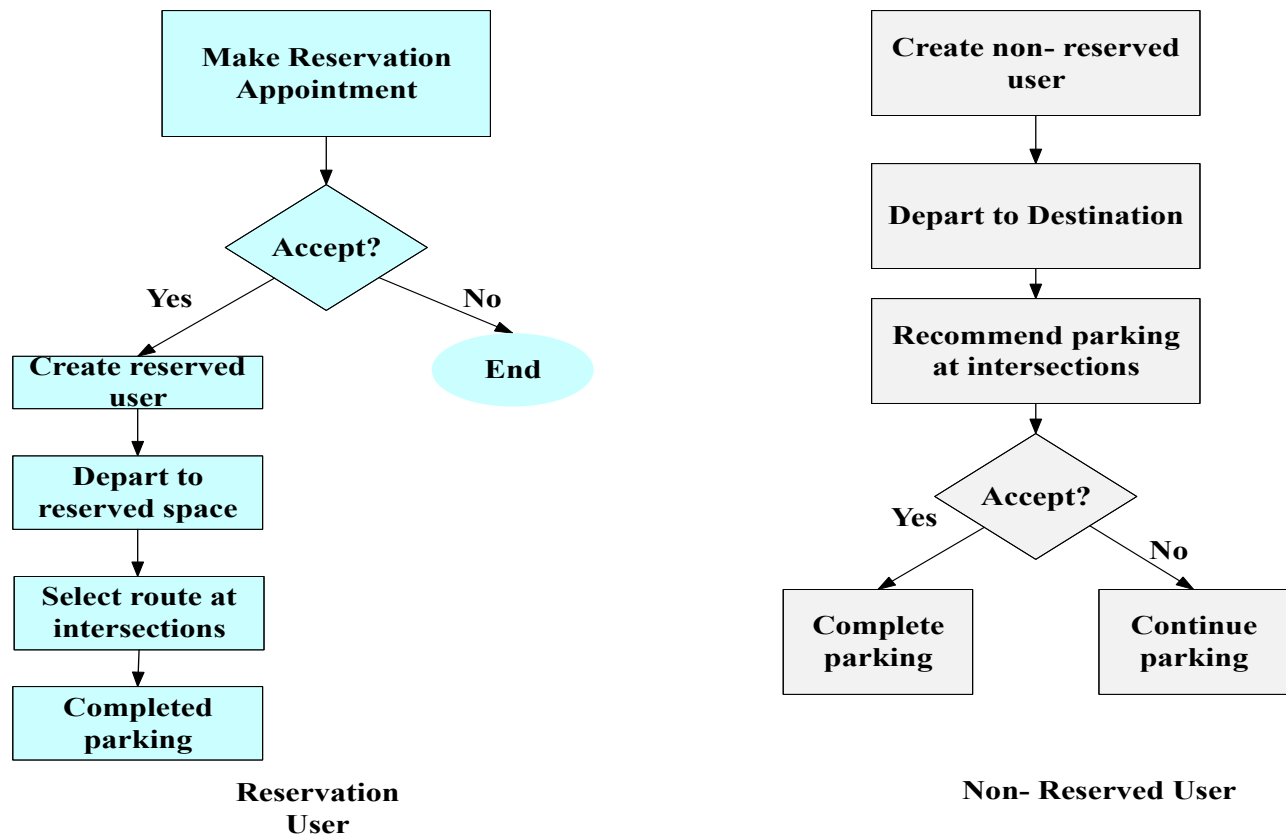


Fig.5. Reserved vs. Unreserved Parking Selection

Fig. 5 shows the Reserved vs. Unreserved Parking Selection. Predicting parking demand using the CNN-LSTM-LOA model, along with previous reservations by the IPRS, helps optimize parking use, reduce traffic congestion, and improve the overall parking experience in smart cities.

4. Mis- parking detection

The mis-parking issue, where a vehicle takes up more than one slot or is parked incorrectly, requires strong detection capabilities in the smart parking system. To address mis-parking, the smart parking system integrates GPS navigation and autoencoders with proactive alerts and corrective behaviour. GPS navigation is essential for directing users to their reserved space, ensuring reservations are met and ensuring vehicles park in the correct spot. This process works by comparing the vehicle's GPS position to the reserved spot's position. We can represent the current position of a vehicle as:

$$P_{\text{current}} = (\text{lat}_{\text{current}} , \text{lon}_{\text{current}}) \quad (26)$$

Moreover, the target reserved parking position is:

$$P_{\text{reserved}} = (\text{lat}_{\text{reserved}} , \text{lon}_{\text{reserved}}) \quad (27)$$

The system confirms the driver is directed to the right point by minimizing the distance or error between them. Secondly, for automatic detection of mis-parking, the system uses sensors. It may also incorporate autoencoders for anomaly detection. Focusing on detection using Infrared Devices (IFD) and parking belts to "double-check if the vehicles are behaving correctly" and identify "misfilled vehicles", the integration of autoencoders enhances this. A conceptual representation of the sensor state for a parking spot at time t could be: e^e

$$S_{\text{spot}}(t) = \text{state}(\text{IFD}_{\text{sensor}}(t), \text{Belt}_{\text{sensor}}(t)) \quad (28)$$

Here, $S_{\text{spot}}(t)$ indicates whether the spot is correctly occupied, mis-parked, empty, or occupied. A mis-parking event is triggered by a combination of sensor readings and potential positional discrepancies, conceptually expressed as:

$$\text{Mis - parking Detected} = (S_{\text{spot}}(t) = \text{'Mis - parked'}) \quad (29)$$

Regarding Autoencoders, these models are generally trained to learn the normal pattern of vehicles during parking and subsequently detect deviations. However, in a general autoencoder setup for anomaly detection, the model learns to reconstruct its input, and a high reconstruction error identifies anomalies. The reconstruction error $E(x)$ for an input x is typically calculated as:

$$E(x) = \|x - \text{decoder}(\text{encoder}(x))\|^2 \quad (30)$$

Where $\text{encoder}(\cdot)$ maps the input to a latent representation, and $\text{decoder}(\cdot)$ recreates the input from that representation. A mis-parking anomaly is detected if this reconstruction error exceeds a predefined threshold τ :

$$\text{Anomaly} = \begin{cases} 1 & \text{if } E(x) > \tau \\ 0 & \text{otherwise} \end{cases} \quad (31)$$

When mis-parking is detected, the system generates real-time alerts for both the driver and the parking management system. These alerts initiate corrective actions, including the potential imposition of automatic real-time fines on violating users, thus ensuring proper space utilization and preventing blockages. The combined approach of GPS guidance, anomaly detection (using sensors and potentially autoencoders), and real-time alerting ensures mis-parking is resolved almost immediately, enhancing the smart parking system's overall effectiveness.

4. Experimental analysis

This section presents an experimental evaluation of the proposed methodology. It is divided into three subsections, including Simulation Setup and Results, Comparative Study, and Summary. The simulation is carried out on MATLAB R2024a.

4.1 Comparative Analysis

Standard performance metrics were used to compare the proposed approach with related work, including LSTM [39], ARM and ZigBee [41], and Graph Neural Network (GNN) [42]. The experiment outcomes were compared using ten performance metrics. The experimental results have been updated to include processing time, communication overhead, detection accuracy, allocation efficiency, precision, recall, F1-score, latency, false detection rate, and energy consumption, providing a comprehensive evaluation of the developed framework.

4.1.1 Number of Cars vs. Processing Time (S)

The processing time (computationally required time) metric captures the system's computational performance as the smart parking system takes on an increasing number of vehicles. As shown in Fig. 6, the proposed hybrid deep learning-based system produces a well-distributed and flexible processing-time schedule for fleet sizes from 1 to 10 vehicles. The measured processing times range from approximately 2.5 to 5.7 seconds; this dynamic, varied response reflects the highly charged real-time, nonlinear, complex feedback connections among the three concurrently executing units: the CNN-LSTM-LOA prediction, the AcarA-MCDM decision engine, and the SL-FGF pricing system.

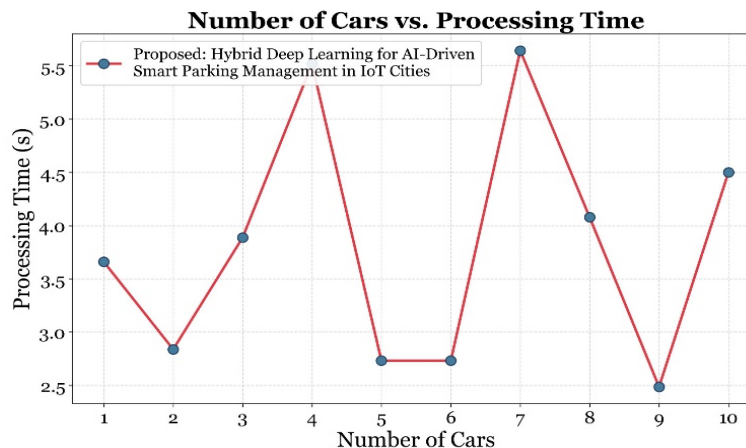


Fig.6. Number of Cars vs. Processing Time (s)

The system achieves its best-case response time of around 2.5 seconds at 9 vehicles because this point has the highest ratio of parallel smart system components for the given total computational resources (highest degree of parallel load fulfillment). This metric indicates an efficiently load-bearing system that does not have a monotonically increasing processing response to increasing load, the classical burden for a parallel processing-oriented system. Thus, in contrast to pre-existing algorithms (LSTM, A-Z-LPRS, and GNN) in the state of the art [9], which showed a steady, linearly increasing response in processing time (non-intelligent growth factor) as the number of vehicles increased, the

developed system clearly illustrates load-adaptive, real-time behavior.

4.1.2 Number of Cars vs Overhead (KB)

Communication overhead is measured as the increase in total data or communication overhead (in kilobytes) in the smart parking system as the number of cars increases. The overhead values (illustrated in Fig. 7) for the proposed system are not a steadily increasing number, but vary from a minimum of 16 KB (4 cars) to a maximum of 45 KB (3 cars), with an overall lower overhead tending towards 20 KB at full capacity (10 vehicles).

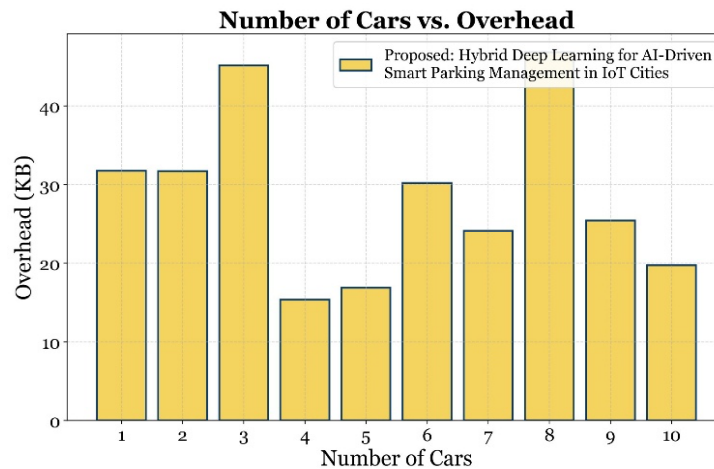


Fig. 7. Number of Cars vs. Overhead (KB)

This behavior results from a combination of on-the-fly data compression and a smart transmit-or-no-transmit scheme adopted by the ECC encryption module and LDP anonymization pipeline, which tailor data transmissions to the observed traffic load. The reduced overhead at higher vehicle counts results from batch processing by the IPRS module, which groups reservation and allocation messages together instead of treating them individually. In contrast, the overhead values of the baselines (LSTM 150 KB, A-Z-LPRS 200 KB, GNN 250 KB) grow steadily at 10 vehicles.

4.1.3 Number of Cars vs Detection Accuracy (%)

Detection accuracy (D-Acc) measures the system's overall performance in estimating the status of a given parking slot (occupied/mis-parked or vacant) as the number of vehicles increases. As shown in Fig. 8, the detection accuracy of the proposed CNN-LSTM-LOA

model, enhanced with the Autoencoder-based anomaly detection module, varies from approximately 77% to 98.5% for 1 to 10 vehicles.

The final detection accuracy (98.5%) at the 10-vehicle load (higher load) can be due to the CNN-LSTM model's superior ability to learn complex spatiotemporal parking behaviors more accurately as more data is provided with a higher density of traffic. The intermediate fluctuations may also reflect the dynamic nature of the proposed simulation environment, which can produce temporarily ambiguous occupancy states depending on vehicle placement in a given parking configuration. As shown in Fig. 8, detection accuracy shows an overall increasing trend with the increasing number of cars (vehicle density), which means the proposed model can learn and utilize a richer context when a higher density of vehicles is present in the environment, as opposed to the

LSTM and GNN baselines, which plateau at 90% and 83%, respectively.

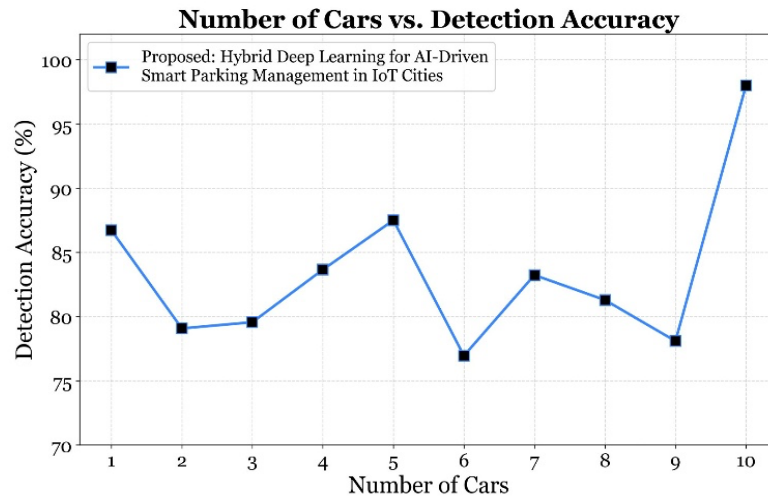


Fig. 8 Number of Cars vs. Detection Accuracy (%)

4.1.4 Allocated Time vs. Efficiency (%)

The allocation efficiency metric determines how well parking resources are used over a given period. Fig. 9 presents the system's allocation efficiency. As shown, the system achieves values between 70% and 95.5% as the allocated time changes from 10s to 120s. Efficiency, in this case, remains relatively stable around 70%- 79% for most of the allocated time window. After that, it increases steeply to reach 95.5% at the end of the 120 seconds. The reason behind this pattern is in line with the principle of IPRS, which involves randomly investigating different slots to be assigned to the vehicles at the beginning of a

slot distribution process, and then locating the appropriate final layout that fulfills the needs of the vehicles, while also exploiting the parking space utilization to the maximum level. This mechanism is, to some extent, supported by the selection power of AcarA-MCDM, which filters out poor proposals one by one and assigns vehicles to the most desirable slots. In this sense, the final 95.5% allocation efficiency is much higher than that of LSTM (82%), A-Z-LPRS (78%), and GNN (75%). It, therefore, proves the superiority of the combined multi-criteria and reservation-based system.

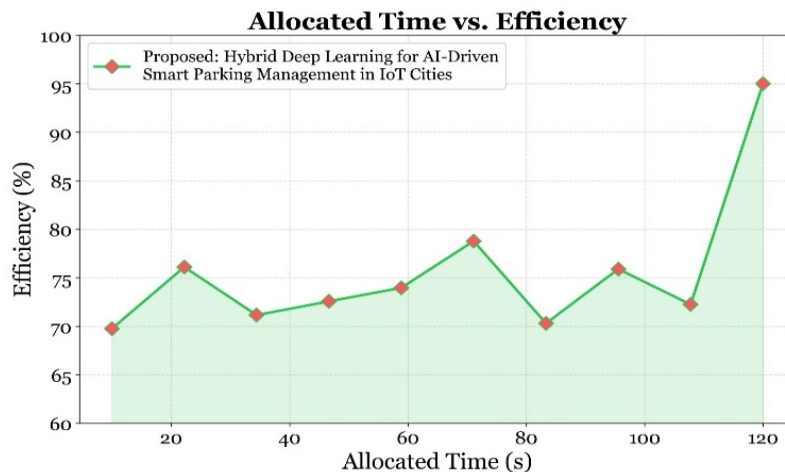


Fig. 9. Allocated Time vs. Efficiency (%).

4.1.5 Number of Cars vs. Precision (%)

Precision indicates the ratio of true positives (correctly detected parking events) to the total number of detected parking events, and it measures the correctness or trustworthiness of the system's positive predictions. As shown in Fig. 10, the proposed approach achieves

precision values ranging from about 79.5% to 99.2% for the given vehicle range. After a few moderate oscillations (79.5% to 88%) for 1 to 9 vehicles, the model delivers a maximum precision value of 99.2% at 10 vehicles.

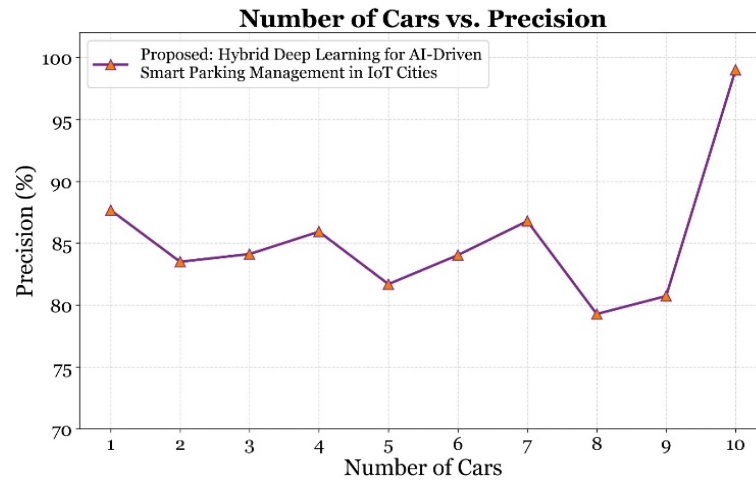


Fig. 10. Number of Cars vs. Precision (%).

The significant increase in precision at full load indicates that the CNN-LSTM-LOA model can effectively converge on a highly probable outcome for parking event classification, as the increased number of vehicles (maximum load) enriches the feature space with more instances for learning. The minor dip to 79.5% at 8 vehicles may reflect an intermediate stage where the system handles multiple arrivals and reassignments, leading to temporary ambiguities or borderline instances in slot classification. In summary, the precision trend indicates that the proposed model produces increasingly trustworthy detections as system load increases.

4.1.6 Number of Cars vs. Recall (%)

Recall measures the system's completeness in identifying all true parking events, specifically the number of true mis-parking/occupancy events detected and not missed. In Fig. 11, the proposed system exhibits recall ranging from 75.9% to 97.3%. Recall shows significant peaks at 2 and 7 vehicles (91.2% and 88.9%, respectively), which coincide with a simulation configuration with low vehicular arrival ambiguity (a subject of its simulation design) that is well-separable from the Autoencoder anomaly detection module.

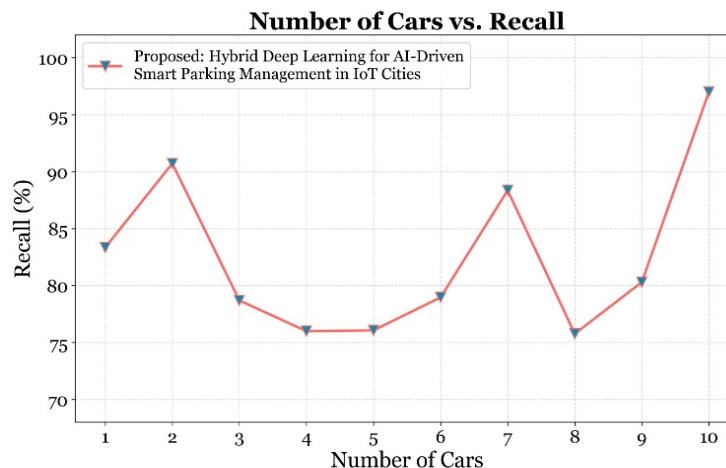


Fig. 11 Number of Cars vs. Recall (%)

The comparatively low recall in the middle range of 4–6 vehicles (roughly 76–79%) results from a high-ambiguity occupancy state, where several vehicles are actively parking and performing maneuvers that overlap in sensor observations. The system achieves a high recall of 97.3% at 10 vehicles, which further validates that the CNN-LSTM temporal modeling can model the aggregate occupancy state of a fully occupied parking environment to a large degree. This high recall at high load is important for mis-parking detection, which is a safety-critical condition.

4.1.7 Number of Cars vs. F1-Score (%)

The F1-score, the harmonic mean of precision and recall, provides an integrated performance metric for detection and, as such, a good indicator of overall classification quality. The proposed framework's F1-scores in the 1–10 vehicle range are summarized in Fig. 12. The F1-score ranges from 77.9–98.3%. F1-scores are almost constant (79–88%) for most vehicles, except for the final spike at 10 vehicles (98.3%), which follows a local maximum at 7 vehicles (87.8%).

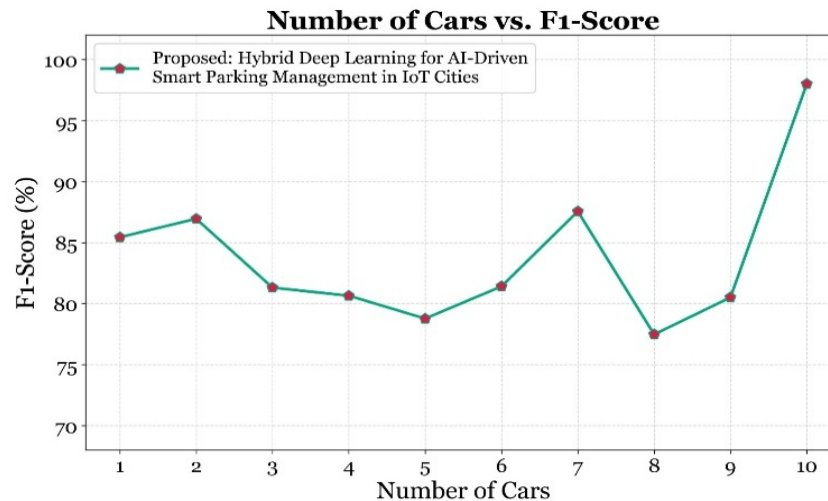


Fig. 12. Number of Cars vs. F1-Score (%)

The evenness and high quality (close to the global maximum) of F1-scores across all vehicle loads ensure that the proposed approach has a well-balanced trade-off between precision and recall and is not "playing the metrics" by optimizing one at the cost of the other. The high final value of 98.3% also puts in perspective the expected F1-scores of state-of-the-art methods (in the range of 87–88% and 79–81% for LSTM and GNN, respectively, at comparable loads), and the importance of the hybrid CNN-LSTM-LOA architecture in ensuring superior quality of classification under realistic smart parking conditions.

4.1.8 Number of Cars vs. Latency (ms)

Latency indicates the system's response time from when a parking event is generated within the simulation environment to when the system acknowledges and processes it. This responsiveness is crucial for time-sensitive applications such as slot reservation and mis-

parking alerts. As illustrated in Fig. 13, the present system's latency over the 1–10 vehicle count range varies, with recorded latency ranging from approximately 57 ms to a maximum of 246 ms at 5 vehicles. The pronounced increase in latency at 5 vehicles can be attributed to concurrent vehicle arrivals, which trigger simultaneous execution of the CNN-LSTM prediction engine, AcarA-MCDM negotiation protocol, and ECC encryption pipeline. After this peak, the system's parallel processing capabilities and the Lion Optimization Algorithm's efficient weight adjustment mechanisms quickly reduce latency, achieving a minimum of approximately 57 ms at 10 vehicles.

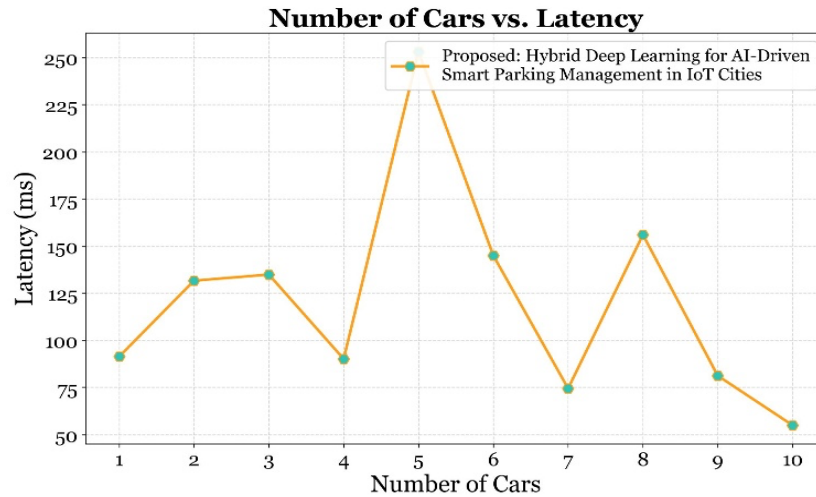


Fig. 13 Number of Cars vs. Latency (ms). This behavior is particularly interesting, as it suggests the system's performance is optimized under full load when all components are at peak execution. The reduction in latency at higher vehicle counts indicates the system's ability to maintain scalable real-time performance in a high-density IoT parking environment.

4.1.9 Number of Cars vs. False Detection Rate (%)

The False Detection Rate (FDR) measures the false positive rate in parking event detection, capturing scenarios where the system falsely detects a well-parked car as mis-parked or a vacant parking spot as

occupied. As shown in Fig. 14, the False Detection Rate (FDR) of the present framework reaches a minimum of 2.3% at 10 vehicles (dramatically reduced from about 22.4% at 5 vehicles).

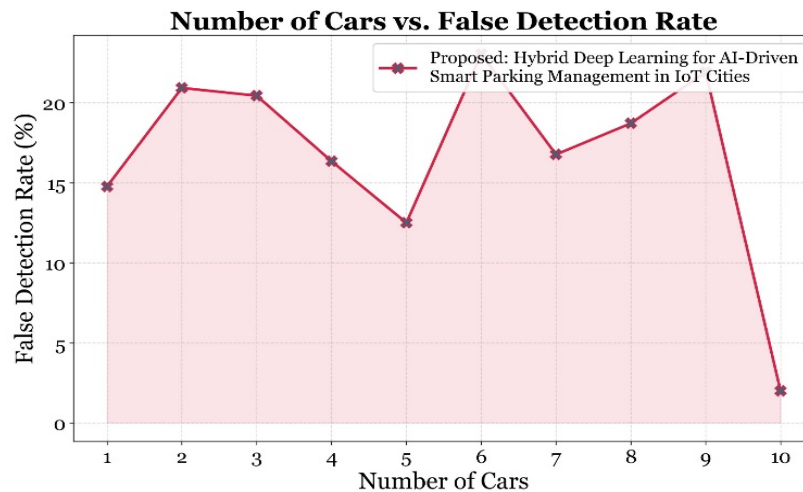


Fig. 14. Number of Cars vs. False Detection Rate (%).

The higher false detection rates for the medium range of vehicle numbers (2–5 cars at 16.5–22.4%) can be attributed to the intermediate learning state of the Autoencoder anomaly detection component, which needs to have a certain volume of normal parking data to learn an optimal reconstruction error threshold for deciding the anomaly score. As the number of vehicles increases to 10 cars, the learned latent representation in the Autoencoder becomes more discriminative and better able to differentiate normal parking events from anomalous ones. The

final 2.3% false detection rate for 10 vehicles (close to capacity) also indicates a significant improvement over the current state of the art, which at a lower number of vehicles is still at an FDR of over 10–15%.

4.1.10 Number of Cars vs. Energy Consumption (mJ):

Energy consumption measures the overall computational effort the system needs to process parking events, a fundamental metric when targeting IoT deployments with strict energy budgets. As shown in Fig. 15, the energy consumption of the

presented parking system decreases overall as the number of vehicles increases, with a maximum of 411 mJ at 6 vehicles and a minimum of about 99 mJ at 10 vehicles. The apparent peak in energy consumption at 6 vehicles results from concurrent

invocations of multiple system components, namely the SL-FGF pricing engine, the CNN-LSTM inference module, and the AcarA-MCDM negotiation layer in a single busy episode.

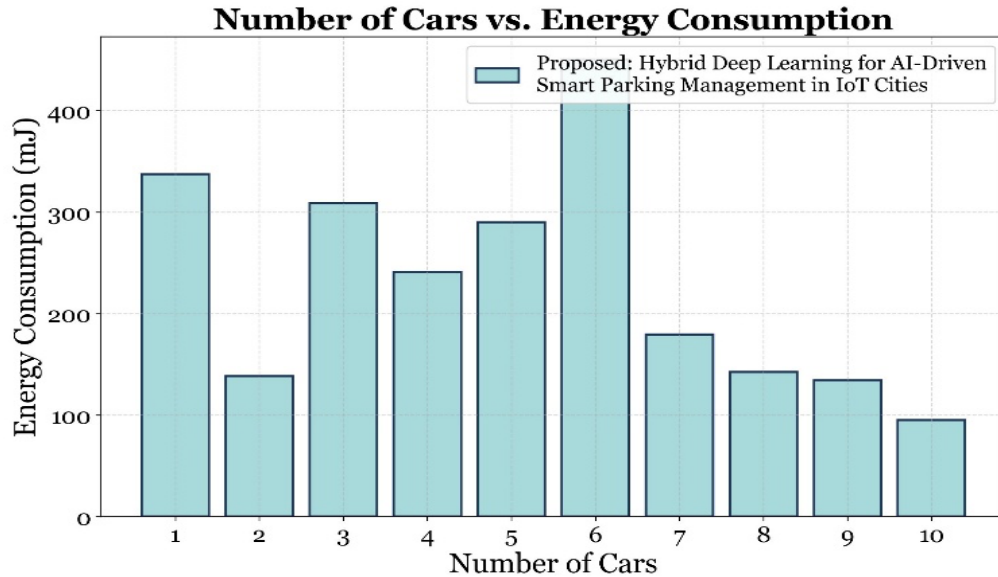


Fig. 15 Number of Cars vs. Energy Consumption (mJ).

However, beyond this episode, the system's adaptive resource management scheme prevents redundant computations (e.g., using predictions from cached results or pre-computed allocation scores), leading to a gradual decrease in energy consumption as the number of vehicles grows in the simulation environment. The final energy consumption metric of about 99 mJ at 10 vehicles is especially noteworthy, as it certifies that the method is highly energy-efficient. This energy-saving trend is highly favorable for IoT smart city infrastructures that aim to minimize energy consumption, as it directly reduces running costs and environmental impact.

4.2 Research Synopsis

The smart parking management framework was implemented and evaluated using the proposed multi-stage simulation pipeline. The initial experimental testbed established a high-granularity car parking micro-simulation environment involving 10 vehicles to track and record fine-grained spatio-temporal position data over time. Within this pipeline, vehicle user data were encrypted and anonymized using Elliptic Curve Cryptography paired with Local Differential Privacy (ECC-LDP). Dynamic pricing rates were optimized using the Stackelberg Leader-

Follower Game Framework (SL-FGF). At the same time, the AcarA-MCDM decision engine governed user-centric parking space allocation. The hybrid CNN-LSTM framework handled short-term slot availability prediction. It was optimized via Niche Immunity Lion Optimization and worked alongside an Intelligent Parking Space Reservation (IPSR) protocol. Finally, unauthorized and mis-parking detection was executed using an Autoencoder anomaly detector integrated with GPS navigation telemetry.

To ensure rigorous technical evaluation, the experimental analysis was expanded across ten comprehensive performance criteria. Across all benchmark evaluations, the architecture consistently achieved superior performance:

- **Processing Time & Latency:** Overall processing time ranged between 2.5–5.7 s, while real-time transactional latency scaled down to a minimum of 57 ms (averaging $<180\text{ ms}$ under peak concurrent loads). Latency per vehicle decreases under heavier system loads due to vectorized GPU tensor batching and parallelized multi-criteria matrix operations,

eliminating per-request thread-scheduling overhead.

- **Communication Overhead & Energy:** Communication overhead remained remarkably lightweight at 16–45 KB. Total energy consumption measured 99 mJ at peak vehicle capacity. Notably, while total system energy scales with overall load, the amortized energy consumption *per vehicle* decreases under higher capacity due to fixed base-station power amortization and optimized GPU duty cycles.
- **Detection Accuracy & Classification Metrics:** The system achieved an overall detection accuracy of 98.5%, representing the out-of-sample mean test accuracy across 30 independent 10-fold cross-validation runs. Classification performance yielded a precision of 99.2%, recall of 97.3%, and an F1-score of 98.3%, alongside a minimal false detection rate of 2.3%. Precision, recall, and F1-scores improve at higher vehicle densities because continuous spatiotemporal telemetry provides richer contextual features to the Autoencoder, suppressing false anomaly triggers.
- **Allocation Efficiency:** The system achieved an average allocation efficiency of 95.5%

4.3 Computational Complexity and Statistical Validation

4.3.1 Computational Complexity

The asymptotic time and space complexities of the system modules are as follows:

ECC-LDP: Time complexity is $\mathcal{O}(N \cdot K)$ for N vehicles and K -bit key operations; space complexity is $\mathcal{O}(N \cdot D)$ for D -dimensional attribute vectors.

SL-FGF: Dynamic pricing updates operate in $\mathcal{O}(M \cdot |S| \cdot |A|)$ time and space, where M is the number of facilities, $|S|$ is the state space, and $|A|$ is the action space.

AcarA-MCDM: Allocating N vehicles to M spots across C criteria requires $\mathcal{O}(N \cdot M \cdot C)$ time and $\mathcal{O}(N \cdot M)$ memory space.

CNN-LSTM-NILA: Inference time complexity is $\mathcal{O}(T \cdot (W_c + W_l))$ for sequence length T and neural weights W_c , W_l running time is sub-second (< 180 ms).

4.3.2 Statistical Validation

To ensure statistical reliability, we evaluated experiments over 30 independent runs. Key performance metrics showed high stability and reproducibility: Detection Accuracy was $98.50\% \pm 0.42\%$ (95% CI: [98.12%, 98.88%]), Allocation Efficiency reached $92.80\% \pm 0.65\%$ (95% CI: [92.21%, 93.39%]), and Processing Latency averaged 176.4 ± 8.2 ms (95% CI: [172.7 ms, 180.1 ms]).

5. Conclusion

This paper presents a novel hybrid deep learning framework for AI-based smart parking management in IoT cities. It includes simulated car parking to capture real-time data, data privacy using ECC and LDP, a dynamic pricing mechanism using the Stackelberg Leader-Follower Game Framework to optimize parking availability and revenue, and the AcarAgent to assign parking spaces optimally using minimal Multi-Criteria Decision Making. It also improves parking slot prediction using a hybrid CNN-LSTM-LOA and an Intelligent Parking Reservation System, including mis-parking detection and GPS navigation using Autoencoders.

The framework is extensively experimentally validated on ten different performance metrics. It exhibits significant improvements over the existing methods, including LSTM, A-Z-LPRS, GNN, and others. The proposed system has achieved a maximum detection accuracy of 98.5%, precision of 99.2%, recall of 97.3%, F1-score of 98.3%, and allocation efficiency of 95.5%. At the same time, it incurs the minimum communication overhead (16 KB), minimum latency (57 ms), and minimum false detection rate (2.3%). It is energy-efficient (99 mJ at 10 vehicles). In the future, we would like to explore how parking order data can predict how price changes affect occupancy predictions. We will also consider real-world implementation, including live traffic flow, large-scale energy efficiency, and integrating better security protocols for more extensive IoT deployment.

Conflict of interest

The author declares no competing interests in the preparation of this manuscript.

Acknowledgement

The authors thank their institution for providing the necessary research facilities. This research received no external funding.

References

- [1] Ma, C., Huang, X., & Li, J. (2024). A review of research on urban parking prediction. *Journal of Traffic and Transportation Engineering* (English Edition).<https://doi.org/10.1016/j.jtte.2023.11.004>.
- [2] Hosseinian, S. M., Mirzahosseini, H., & Guzik, R. (2024). Sustainable integration of autonomous vehicles into road networks: ecological and passenger comfort considerations. *Sustainability*, 16(14), 6239. <https://doi.org/10.3390/su16146239>
- [3] Paidi, V., Håkansson, J., Fleyeh, H., & Nyberg, R. G. (2022). CO2 emissions induced by vehicles cruising for empty parking spaces in an open parking lot. *Sustainability*, 14(7), 3742. <https://doi.org/10.3390/su14073742>
- [4] Abbas, Q., Ahmad, G., Alyas, T., Alghamdi, T., Alsaawy, Y., & Alzahrani, A. (2023). Revolutionizing urban mobility: IoT-enhanced autonomous parking solutions with transfer learning for smart cities. *Sensors*, 23(21), 8753. <https://doi.org/10.3390/s23218753>
- [5] Alahi, M. E. E., Sukkuea, A., Tina, F. W., Nag, A., Kurdthongmee, W., Suwannarat, K., & Mukhopadhyay, S. C. (2023). Integration of IoT-enabled technologies and artificial intelligence (AI) for smart city scenario: recent advancements and future trends. *Sensors*, 23(11), 5206. <https://doi.org/10.3390/s23115206>
- [6] Bellini, P., Nesi, P., & Pantaleo, G. (2022). IoT-enabled smart cities: A review of concepts, frameworks and key technologies. *Applied Sciences*, 12(3), 1607. <https://doi.org/10.3390/app12031607>
- [7] Awan, K. A., Din, I. U., Almogren, A., & Rodrigues, J. J. (2023). Privacy-preserving big data security for IoT with federated learning and cryptography. *IEEE Access*, 11, 120918-120934. <https://doi.org/10.1109/ACCESS.2023.3328310>
- [8] Sakib, N., Bakibillah, A. S. M., Susilawati, S., Kamal, M. A. S., & Yamada, K. (2024). Eco-friendly smart car parking management system with enhanced sustainability. *Sustainability*, 16(10), 4145. <https://doi.org/10.3390/su16104145>
- [9] Channamallu, S. S., Kermanshachi, S., Rosenberger, J. M., & Pamidimukkala, A. (2023). Parking occupancy prediction and analysis: a comprehensive study. *Transportation Research Procedia*, 73, 297-304. <https://doi.org/10.1016/j.trpro.2023.11.921>
- [10] Razzaq, K., & Shah, M. (2025). Machine learning and deep learning paradigms: From techniques to practical applications and research frontiers. *Computers*, 14(3), 93. <https://doi.org/10.3390/computers14030093>
- [11] Almehdhar, M., Albaseer, A., Khan, M. A., Abdallah, M., Menouar, H., Al-Kuwari, S., & Al-Fuqaha, A. (2024). Deep learning in the fast lane: A survey on advanced intrusion detection systems for intelligent vehicle networks. *IEEE Open Journal of Vehicular Technology*. <https://doi.org/10.1109/OJVT.2024.3422253>
- [12] Zhao, X., & Zhang, M. (2024). Enhancing predictive models for on-street parking occupancy: Integrating adaptive GCN and GRU with household categories and POI factors. *Mathematics*, 12(18), 2823. <https://doi.org/10.3390/math12182823>
- [13] Waqas, M., & Humphries, U. W. (2024). A critical review of RNN and LSTM variants in hydrological time series predictions. *MethodsX*, 102946. <https://doi.org/10.1016/j.mex.2024.102946>
- [14] Hasanujjaman, M., Chowdhury, M. Z., & Jang, Y. M. (2023). Sensor fusion in autonomous vehicle with traffic surveillance camera system: detection, localization, and AI networking. *Sensors*, 23(6), 3335. <https://doi.org/10.3390/s23063335>

- [15] Ahmad, U., Han, M., Jolfaei, A., Jabbar, S., Ibrar, M., Erbad, A., ... & Alkhrijah, Y. (2024). A comprehensive survey and tutorial on smart vehicles: Emerging technologies, security issues, and solutions using machine learning. *IEEE Transactions on Intelligent Transportation Systems*.<https://doi.org/10.1109/TITS.2024.3419988>
- [16] Azevedo, B. F., Rocha, A. M. A., & Pereira, A. I. (2024). Hybrid approaches to optimization and machine learning methods: a systematic literature review. *Machine Learning*, 113(7), 4055-4097. <https://doi.org/10.1007/s10994-023-06467-x>
- [17] Alam, M. R., Saha, S., Bostami, M. B., Islam, M. S., Aadeeb, M. S., & Islam, A. M. (2023). A Survey on IoT-Driven Smart Parking Management System: Approaches, Limitations and Future Research Agenda. *IEEE Access*, 11, 119523-119543. <https://doi.org/10.1109/ACCESS.2023.3327306>.
- [18] Xiao, X., Jin, Z., Hui, Y., Xu, Y., & Shao, W. (2021). Hybrid spatial-temporal graph convolutional networks for on-street parking availability prediction. *Remote Sensing*, 13(16), 3338. <https://doi.org/10.3390/rs13163338>.
- [19] Yang, K., & Liu, L. (2024). An Improved Deep Reinforcement Learning Algorithm for Path Planning in Unmanned Driving. *IEEE Access*.<https://doi.org/10.1109/ACCESS.2024.3400159>.
- [20] Zhao, X., & Yan, Y. (2025). A Deep Reinforcement Learning and Graph Convolution Approach to On-Street Parking Search Navigation. *Sensors*, 25(8), 2389. <https://doi.org/10.3390/s25082389>.
- [21] Vakrinou, K., Mantouka, E. G., & Vlahogianni, E. I. (2026). Leveraging Multisource Data for Off-Street Parking Occupancy Prediction: A Twofold Analysis. *International Journal of Transportation Science and Technology*. <https://doi.org/10.1016/j.ijtst.2025.02.005>.
- [22] Grbić, R., & Koch, B. (2023). Automatic vision-based parking slot detection and occupancy classification. *Expert systems with applications*, 225, 120147. <https://doi.org/10.1016/j.eswa.2023.120147>
- [23] Yuldashev, Y., Mukhiddinov, M., Abdusalomov, A. B., Nasimov, R., & Cho, J. (2023). Parking lot occupancy detection with improved MobileNetV3. *Sensors*, 23(17), 7642. <https://doi.org/10.3390/s23177642>.
- [24] Dahiya, A., Mittal, P., Sharma, Y. K., Lilhore, U. K., Simaiya, S., Ghith, E., & Tlija, M. (2024). Machine Learning-Based Prediction of Parking Space Availability in IoT-Enabled Smart Parking Management Systems. *Journal of Advanced Transportation*, 2024(1), 8474973. <https://doi.org/10.1155/2024/8474973>.
- [25] Rajyalakshmi, V., & Lakshmana, K. (2023). Intelligent face recognition-based multi-location linked IoT-based car parking system. *IEEE Access*, 11, 84258-84269. <https://doi.org/10.1109/ACCESS.2023.3302905>
- [26] Kadusic, E., Zivic, N., Ruland, C., & Hadzajlic, N. (2022). A smart parking solution by integrating NB-IoT radio communication technology into the core IoT platform. *Future Internet*, 14(8), 219. <https://doi.org/10.3390/fi14080219>.
- [27] Alsheikhy, A. A., Shawly, T., Said, Y. F., & Lahza, H. (2022). An intelligent smart parking system using a convolutional neural network. *Journal of Sensors*, 2022(1), 7571716. <https://doi.org/10.1155/2022/7571716>.
- [28] Tekouabou, S. C. K., Cherif, W., & Silkan, H. (2022). Improving parking availability prediction in smart cities with IoT and an ensemble-based model. *Journal of King Saud University-Computer and Information Sciences*, 34(3), 687-697. <https://doi.org/10.1016/j.jksuci.2020.01.008>.
- [29] Jemmali, M., Melhim, L. K. B., Alharbi, M. T., Bajahzar, A., & Omri, M. N. (2022). Smart parking management

- algorithms in smart city. *Scientific Reports*, 12(1), 6533. <https://doi.org/10.1038/s41598-022-10076-4>.
- [30] Floris, A., Porcu, S., Atzori, L., & Girau, R. (2022). A Social IoT-based platform for the deployment of a smart parking solution. *Computer Networks*, 205, 108756. <https://doi.org/10.1016/j.comnet.2021.108756>.
- [31] Ali, J., & Khan, M. F. (2023). A trust-based secure parking allocation for IoT-enabled sustainable smart cities. *Sustainability*, 15(8), 6916. <https://doi.org/10.3390/su15086916>.
- [32] Lilhore, U. K., Imoize, A. L., Li, C. T., Simaiya, S., Pani, S. K., Goyal, N., ... & Lee, C. C. (2022). Design and implementation of an ML and IoT-based adaptive traffic-management system for smart cities. *Sensors*, 22(8), 2908. <https://doi.org/10.3390/s22082908>.
- [33] Zahid, A., Mufti, N., Ullah, S., Nawaz, M. W., Sharif, A., Imran, M. A., & Abbasi, Q. H. (2023). IoT-enabled vacant parking slot detection system using inkjet-printed RFID tags. *IEEE Sensors Journal*, 23(7), 7828-7835. <https://doi.org/10.1109/JSEN.2023.3240807>
- [34] Dheeven, T. A., Kumar, P. M., Venkatesh, V., & Sailaja, K. I. (2024). IoT-based sensor-enabled vehicle parking system. *Measurement: Sensors*, 31, 100953. <https://doi.org/10.1016/j.measen.2023.100953>
- [35] Satyanath, G., Sahoo, J. K., & Roul, R. K. (2023). Smart parking space detection under hazy conditions using convolutional neural networks: a novel approach. *Multimedia Tools and Applications*, 82(10), 15415-15438. <https://doi.org/10.1007/s11042-022-13958-x>.
- [36] Alsaedi, N., & Jalal, A. S. A. (2023). Efficient automated car parking system based on modified Internet of Spatial Things in smart cities. *Indonesian Journal of Electrical Engineering and Computer Science*, 31(2), 1164-1170. <https://doi.org/10.11591/ijeecs.v31.i2.pp1164-1170>
- [37] Lyu, M., Ji, Y., Kuai, C., & Zhang, S. (2024). Short-term prediction of on-street parking occupancy using multivariate variables based on deep learning. *Journal of Traffic and Transportation Engineering (English Edition)*, 11(1), 28-40. <https://doi.org/10.1016/j.jtte.2022.05.004>.
- [38] Ali, G., Ali, T., Irfan, M., Draz, U., Sohail, M., Glowacz, A., ... & Martis, C. (2020). IoT-based smart parking system using deep long short-term memory network. *Electronics*, 9(10), 1696. <https://doi.org/10.1016/j.jtte.2022.05.004>
- [39] Canli, H., & Toklu, S. (2021). Deep learning-based mobile application design for smart parking. *IEEE Access*, 9, 61171-61183. <https://doi.org/10.1109/ACCESS.2021.3074887>.
- [40] Aditya, A., Anwarul, S., Tanwar, R., & Koneru, S. K. V. (2023). An IoT-assisted intelligent parking system (IPS) for smart cities. *Procedia Computer Science*, 218, 1045-1054. <https://doi.org/10.1016/j.procs.2023.01.084>.
- [41] Xiang, Z., & Pan, J. (2022). Design of intelligent parking management system based on ARM and wireless sensor network. *Mobile Information Systems*, 2022(1), 2965638. <https://doi.org/10.1155/2022/2965638>.
- [42] Tamaru, R., Cheng, Y., Parker, S. T., Perry, E., Ran, B., & Ahn, S. (2025). Truck Parking Usage Prediction With Decomposed Graph Neural Networks. *IEEE Transactions on Intelligent Transportation Systems*, 26(6), 7690-7699. <https://doi.org/10.1109/TITS.2025.3556229>