

# Deep Learning-Driven Smart Parking Reservation System for Urban Traffic Management

Mr. N. Sekar<sup>1</sup> Dr. S. Nithya<sup>2</sup>

<sup>1</sup>Research Scholar, Department of Computer Science Park's College (Autonomous) Tiruppur, Tamil Nadu, India

<sup>2</sup>Assistant Professor, Department of Computer Science, A.V.P College of Arts and Science (Autonomous), Tiruppur, Tamil Nadu, India

## Article History:

Received 05-08-2025

Revised 01-09-2025

Accepted 12-10-2025

## Abstract:

Parking scarcity and outdated availability data pose significant challenges in metropolitan areas, especially during peak hours. This study introduces an intelligent, reservation-based parking management system leveraging a deep learning framework that integrates Long Short-Term Memory and Deep Residual Recurrent Neural Networks (LSTM-DR-RNN). The proposed system enables real-time prediction and scheduling of vacant parking spots through a cloud-based mobile application, enhancing user convenience by considering regional parking conditions rather than isolated lots. By incorporating external factors such as weather conditions, calendar dates, and traffic flow, the model achieves improved forecasting accuracy. Evaluated using real-world datasets from multiple cities, the approach demonstrates reduced search time and fuel consumption. Comparative analysis with baseline models highlights the system's effectiveness, achieving a mean absolute error (MAE) of 0.89, mean squared error (MSE) of 2.52, and root mean squared error (RMSE) of 1.57, with an occupancy prediction accuracy of 99.6%. This solution serves as an efficient parking enforcement mechanism, reducing congestion and optimizing urban parking availability.

Keywords—smart parking system, deep learning, LSTM-DR-RNN, parking reservation and parking allocation.

## INTRODUCTION

Cities are evolving into smarter hubs with the growing integration of the Internet of Things (IoT). IoT solutions are rapidly expanding into various urban sectors, including transit systems, airlines, healthcare facilities, retail centers, and more. This widespread adoption is transforming urban areas into highly connected and adaptable environments, enhancing the efficiency and interconnectivity of metropolitan infrastructure. By 2024, the total number of IoT-connected devices is expected to exceed 30 billion, marking a significant milestone in information technology development [1]. Over the last decade, technological advancements have significantly improved daily life. A key example is smart

parking networks, which have become essential to IoT-enabled smart cities, enhancing urban efficiency and convenience.

Urbanization and rising vehicle ownership have led to increasing demand for parking in cities, intensifying traffic congestion and making the search for parking spaces a time-consuming challenge, especially during peak hours. This issue complicates urban transportation management, as it results in higher fuel consumption, increased greenhouse gas emissions, and worsened traffic conditions. Traditional parking systems often have a limited scope, focusing on specific parking areas rather than providing a comprehensive view of citywide parking availability.

Smart Parking Systems (SPS) have emerged as a potential solution to these issues, utilizing machine learning and IoT to provide real-time data on available parking spaces [2-8]. However, many existing systems fail to account for variables like weather conditions and time of day, which affect parking capacity. Additionally, these systems often lack scalability, limiting their ability to manage parking effectively in densely populated cities with high demand. Istanbul, Turkey's largest metropolitan area with a population of 15.5 million, serves as a prime example. As a cultural and commercial hub, it faces severe traffic issues, particularly during rush hours when finding parking is a significant challenge. While current methods provide temporary relief, the increasing number of vehicles calls for more innovative solutions.

In modern urban settings, auto transport tracking has been significantly improved by deep learning techniques. A collaborative approach based on reinforcement learning was proposed in [10] to address parking space distribution, employing intelligent decision-making for more efficient allocation. Furthermore, an ant colony optimization method was applied to optimize the leasing process for shared electronic scooter providers, enhancing urban occupancy efficiency [11]. Finally, in [12], the process of building a robust middleware platform that is cost-effective was discussed, aimed at creating an integrated spots-to-park management system to simplify operations and enhance the client experience across various parking lots.

To predict the availability of parking spots and allow users to make real-time bookings, this article introduces an LSTM-DRRNN model. This approach offers more accurate estimates of parking demand and supply by considering the entire parking environment, unlike traditional models. It enhances parking management efficiency by incorporating user habits, historical patterns, as well as factors like weather and calendar events. Data validation in Istanbul shows that the LSTM-DRRNN model

outperforms existing methods by reducing fuel consumption through shorter search times and improving forecasting accuracy. Key new findings from this study include:

- **Aggregate LSTM-DRRNN Method:** The newly developed LSTM-DRRNN algorithm significantly enhances the accuracy of parking availability forecasts compared to existing methods.
- **Comprehensive Parking Evaluation:** Unlike traditional models, the algorithm evaluates parking accessibility across large areas, not just individual lots, enabling better allocation of parking spaces.
- **User Habits and External Influences:** To optimize resource use and improve customer satisfaction, the algorithm takes into account factors such as traffic flow, historical parking data, customer demands, and environmental influences.
- **Real-Time Appointment Scheduling:** The platform allows drivers to book parking spots in advance, reducing search times and alleviating congestion in areas with limited availability.
- **Validation with Istanbul Data:** The model's effectiveness is demonstrated using real-world data from Istanbul, showing improved forecasting accuracy and reduced fuel consumption by minimizing search times.

The following is the outline of the paper: Section 2 provides a survey of the pertinent scholarship, with an emphasis on seminal works on the subject. Section 3 provides the suggested model and describes the study methods. Section 4 assesses the outcomes in relation to previous studies' conclusions using a number of metrics, such as MSE, RMSE, MAE, MdAE, and MSLE. Section 5 concludes the article by summarizing the main points.

## 1. RELATED WORK

There is ongoing research aimed at optimizing the use of parking spaces, a topic that has been extensively explored in published work. Rafique et al. [13] propose an intelligent parking management platform utilizing deep learning techniques, specifically YOLO v5, to improve vehicle identification. The algorithm achieved an impressive accuracy of 99.5% when tested on the PKLot dataset. Functionality in real time was enhanced by detecting vacant parking spots and extracting vehicle characteristics through a refined YOLO v5 algorithm trained on the MS COCO database.

Dahiya et al. [14] provide an evaluation of machine learning (ML) algorithms for predicting parking availability in IoT-enabled environments. The analysis includes models such as K-nearest neighbors

(KNN), decision tree (DT), support vector machines (SVM), Naïve Bayes (NB) and logistic regression (LR). Performance metrics like clarity, recall, precision, and F1-score were used for assessment. Results indicate that the radio frequency (RF) system demonstrated superior effectiveness and reliability, even under complex conditions.

To optimize utilization, Errousso et al. [15] introduce two integer programming models for parking spot allocation: one tailored for individual vehicles and the other for carriers. The first model assigns a single parking spot per driver, while the second allocates two spots based on demand. For smaller problem instances, the organic basic method is applied to compute solutions, whereas the DNA algorithm and the Tabu Search method are employed for larger scenarios. To address uneven parking demand across different city zones, requests are redistributed. When tested in Casablanca, this approach achieved a 75% satisfaction rate, matching previous systems, while reducing unmet demand to 12.7%, halving space-related expenses, and lowering transportation costs by 40%.

Focusing on achieving population parity in parking space distribution, Jemmali et al. [16] propose an innovative method for equitable allocation of parking lots. To minimize disparities in human distribution across parking areas, seven strategies are developed, all of which consider the availability of lots. Extensive testing of period performance and disparity calculations is conducted on 2,430 examples. The results highlight exceptional efficiency, with the Minimal Residuals (MR) method surpassing other widely recognized techniques in the literature, achieving 96.1% efficacy, a consensus distance of 0.02, and a swift execution time of 0.007 seconds.

Considering drivers' preferences and parking area capacity constraints, Wang and Chen [17] develop a theoretical framework for dynamically distributing parking demand. The proposed solution method's efficacy is verified through a numerical example. The study offers insights into optimizing parking locations and pricing strategies, examining how variable pricing influences parking decisions and ensures balanced lot utilization.

Using a multidisciplinary agent-based framework, Mei et al. [18] explore the impact of parking reservation systems on park-and-ride infrastructures. The research, focused on Suzhou's Guanqian Street Trade District in China, evaluates various parking restriction strategies aimed at reducing greenhouse gas emissions. The findings indicate that pollutant levels decrease when the reservation ratio remains at or below 70%, but emissions increase beyond that threshold, as parking spots are allocated based on demand rather than supply. A natural algorithm is suggested to optimize designated parking space distribution, further reducing emissions.

Yan et al. [19] address the fluctuating distribution of parking resources by framing it as an integrated chance-constrained model aimed at maximizing projected total revenue over a limited period. The model is refined into a robust mixed-integer programming framework using joint chance constraint reduction and sample average estimation, handling complexity from two stochastic variables. A stratified ranked set-generation method is applied to create a high-quality sample set, enhancing the solution technique with effective constraints. Real-world data-based computations validate the method for diverse parking scenarios.

Guo et al. [20] propose a car relocation strategy in parking lots using Reinforcement Learning (RL) to improve space management. The system autonomously makes decisions and optimizes resource usage through a flexible scheduling technique and Markov-based decision-making. Unlike conventional approaches, this method emphasizes long-term strategies to maximize future returns. Experimentally, it has demonstrated high efficiency and resilience, enabled effective route planning and reduced parking time.

To improve parking availability predictions, Elomiya et al. [21] present a hybrid approach that integrates Adaptive Neuro-Fuzzy Inference System (ANFIS) with deep learning (DL) techniques. ANFIS addresses uncertainty using fuzzy set concepts, while DL models excel in pattern recognition and handling time series deviations. The ANFIS-RNN, ANFIS-LSTM, and ANFIS-GRU models—combining ANFIS with RNN, LSTM, and GRU, respectively—were evaluated using real-world parking data. The study highlights the predictive effectiveness of these hybrid algorithms, providing valuable insights for city planners and traffic engineers.

Li et al. [22] develop a distributed parking allocation (DPA) model aimed at maximizing net profit for a platform by optimizing short-term parking demands from slot providers. The model employs the Ant Colony Optimization (ACO) technique and compares its performance with the First-Come, First-Served (FCFS) approach. It considers short-term parking space demands using both specialized and uniform pricing strategies. The evaluation focuses on three key metrics: time disintegration level, net revenue, and utilization rate. The findings show that the ACO algorithm enhances allocation efficiency, supporting the practical viability of combining specialized allocation with uniform pricing.

**Table 1: Research gap among parking reservation system**

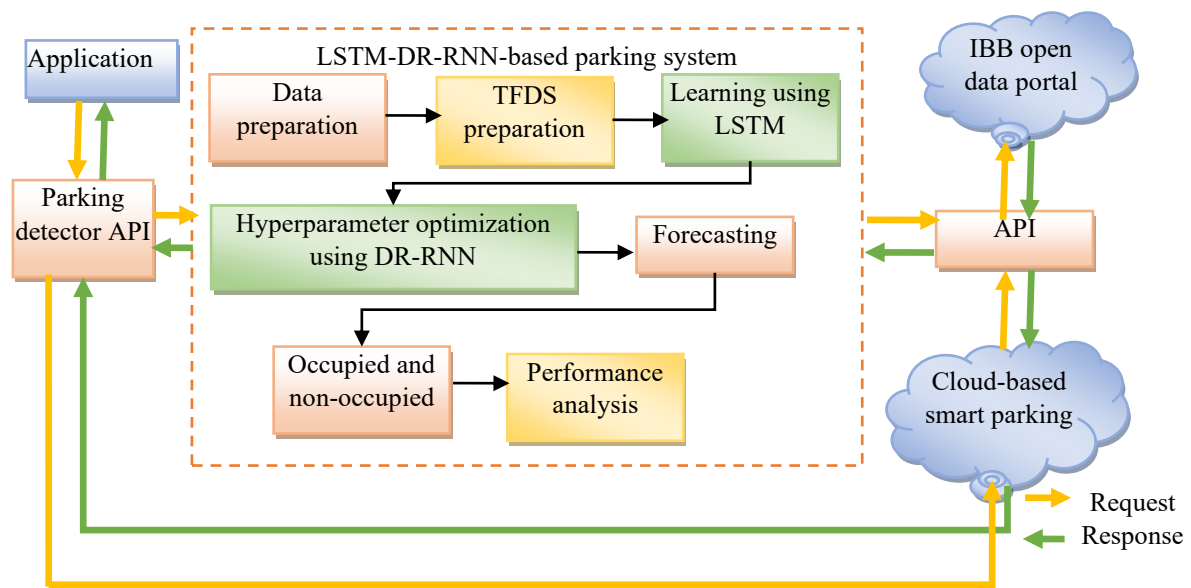
<b>Author name</b>	<b>Methods</b>	<b>Advantages</b>	<b>disadvantages</b>
Rafique et al., [13]	YOLO v5 method	<ul style="list-style-type: none"> <li>Operating with near-instantaneous speed, the machine achieved 96.8% accuracy on a custom database.</li> </ul>	<ul style="list-style-type: none"> <li>However, if the framework relies on a preconditioned model, it may struggle to adapt to parking scenarios not represented in the training data.</li> </ul>
Dahiya et al., [14]	RF method	<ul style="list-style-type: none"> <li>The results indicate that advanced algorithms, particularly randomly distributed forests, enhance the efficiency of parking facility management, reduce delays, and improve customer satisfaction.</li> </ul>	<ul style="list-style-type: none"> <li>However, variations in the initial training data and environmental conditions may diminish the effectiveness of the randomly selected forest model in applications requiring real-time responses.</li> </ul>
Errouss o et al., [15]	Banning Finding and Genetic Computing	<ul style="list-style-type: none"> <li>Experienced an increase of up to 87.3% in parking demand. Customer satisfaction levels improved significantly and Mobility costs were reduced by 40%.</li> </ul>	<ul style="list-style-type: none"> <li>In larger cities, a more comprehensive approach may be required for complex calculations.</li> <li>Due to the demand distribution balance, some parking areas may not be fully utilized.</li> </ul>
Jemmali et al., [16]	MR algorithm	<ul style="list-style-type: none"> <li>Its proven scalability in large operational environments makes it suitable for current applications.</li> </ul>	<ul style="list-style-type: none"> <li>The complexity and limitations of using the seven methods could make maintaining system</li> </ul>

			functionality more challenging.
Wang & Chen [17]	Dynamic Pricing Optimization	<ul style="list-style-type: none"> <li>By considering drivers' preferences in real time, the approach can dynamically assign parking spots, enhancing garage planning and optimizing the use of available spaces.</li> </ul>	<ul style="list-style-type: none"> <li>Drivers' behavior may become unpredictable when approximate methods and fluctuating pricing are used, making it more difficult to accurately predict demand and allocate resources effectively.</li> </ul>
Mei et al., [18]	Genetic Algorithm	<ul style="list-style-type: none"> <li>Finds the best way to reserve parking spots in order to cut down on greenhouse gasses.</li> </ul>	<ul style="list-style-type: none"> <li>If the fundamental demand changes, settled registration percentages could lead to incompetence.</li> </ul>
Yan et al., [19]	sample average approximation (SAA) and stratified ranked set sampling (RSS).	<ul style="list-style-type: none"> <li>The simulation's use of data-driven approaches and indeterminate elements allows it to optimize predicted proceeds.</li> </ul>	<ul style="list-style-type: none"> <li>Computing performance and execution in practice may be hindered by the extent of the model's complexity.</li> </ul>
Guo et al. [20]	Method for Probabilistic Decisions	<ul style="list-style-type: none"> <li>By optimizing strategies that would continue while executing decisions independently, the device</li> </ul>	<ul style="list-style-type: none"> <li>To get the most out of reinforcement teaching, you may need a lot of information and space for conditioning.</li> </ul>

		improves vehicle reliability.	
Elomiya et al., [21]	ANFIS with deep learning (DL)	<ul style="list-style-type: none"> <li>By integrating The ANFIS framework and deep learning approaches, the mixture of techniques dramatically improves the preciseness of vehicle-occupied predictions.</li> </ul>	<ul style="list-style-type: none"> <li>The difficulty of combining Network Functions Information with deep learning methods may hinder the long-term deployment of systems, which might lead to a rise in computing needs.</li> </ul>
Li et al., [22]	ACO	<ul style="list-style-type: none"> <li>The algorithm developed by ACO dramatically improves the distribution of parking places, which boosts revenue and use.</li> </ul>	<ul style="list-style-type: none"> <li>The reliability of demand forecasts and pricing tactics affects the simulation's performance.</li> </ul>

## 2. PROPOSED METHODOLOGY

Figure 1 illustrates the general layout of the proposed automated Smart Parking System (SPS) architecture, which is based on Long Short-Term Memory (LSTM)-DR-RNN. This innovative approach, designed for smartphones, leverages cloud computing and deep learning to address the parking challenge in Istanbul. Through a model built with advanced deep learning algorithms, users can easily access both real-time and predicted parking data via the smartphone app. The process consists of two key steps. First, data from 731 parking facilities in the Turkish capital is continuously collected and stored in a centralized database. This data includes arrival rates, available spots, and location details, which may also include temperature and duration. Second, when queries are made through the smartphone app, a sophisticated learning-based LSTM model combined with DR-RNN processes the requests. This enhanced system uses both historical and real-time data to predict the immediate and future occupancy rates of the recommended parking spaces, providing quick and accurate responses to users' parking inquiries, thereby improving the city's parking situation.



**Figure 1: LSTM-DR-RNN-based parking system overall model**

As shown in Figure 1, the Parking Finder API processes requests from the application interface and provides a list of the most appropriate parking spaces, displaying all car parks within a 2 km radius of the user's destination. The TensorFlow Datasets (TFDS) are then prepared. For each parking space, a request is sent either to the CB-SPS or the LSTM-DR-RNN-Based Service, depending on the specified date. If the request pertains to real-time occupancy rates, the data is retrieved directly from the cloud-based SPS server. For future dates, the request is directed to the LSTM-Based Service, where the LSTM model utilizes deep learning to predict the parking space's occupancy rate and returns the forecasted result. Additional information on data collection from the IBB Open Data Portal is provided in the dataset creation section of the Istanbul SPS.

**i. LSTM-DR-RNN based SPS**

With the rapid advancement of deep learning technologies, traditional smart city parking solutions are quickly becoming outdated, sparking interest in innovative resource management approaches. This research introduces a cutting-edge method that leverages a cloud-based framework powered by deep learning to address parking issues. The proposed solution offers real-time and predictive parking availability, ensuring users can easily find open parking spaces.

A smartphone and web application called "Istanbul SPS" has been developed to help users locate the best available parking spots. The app helps users save time and energy by predicting when parking spaces will become available at various times throughout the day. Piloted in Turkey's largest city, known for its heavy traffic and dense urban areas, the app aims to improve parking efficiency.

The software features an intuitive interface where users simply input their desired date and location. Upon pressing the "search" button, the app uses smart algorithms to identify available parking in the area. Key factors considered include "walking time," which estimates the time to walk from the parking space to the destination, and "occupancy," which indicates parking spot availability. Results are ordered by proximity to ensure the closest options are shown first.

The app utilizes an LSTM system, integrated with a DR-RNN, to predict parking availability in the future. This combination excels at processing multiple input parameters and time-series data. The LSTM-DR-RNN model uses eight variables to predict hourly occupancy rates from a dataset of over 4.5 million records collected from parking facilities in Istanbul. The model boasts an impressive 99.6% accuracy for short- and medium-term predictions.

The LSTM-DR-RNN model is updated weekly to incorporate new data, allowing the system to adjust to changing conditions and fine-tune its forecasting accuracy over time, reducing error rates.

By providing real-time parking data in JSON format, the system supports the city's innovative urban projects. With this solution, both locals and visitors can save time, effort, and money by planning their routes more efficiently. LSTM RNNs, by learning persistent dependencies in sequential data, overcome the limitations of traditional RNNs. Through specialized gates, LSTM models efficiently manage data flow, making them ideal for predicting time-series data like parking availability. An enhanced explanation of the LSTM framework is included for further clarification:

#### a. The Order and Size of Inputs

The period-series input collection of parking capacity data across phases of time  $T$  may be represented as  $A = \{a_1, a_2, \dots, a_T\}$ , where each  $a_t$  may incorporate information like the rate of occupants, duration of the entire day, target location, etc. These patterns will be fed into the LSTM in order to forecast when spaces will be available.

#### b. Neurons and Barriers to Recall

Cell state  $cel_t$  and the concealed state  $hi_t$ , together with the 3 gates that regulate the flow of knowledge, are the fundamental parts of LSTM:

- **Ignore Gate:** Selects data from a particular cell state to exclude.

$$fg_t = \sigma(wt_{fg} \cdot [hi_{t-1}, a_t] + bi_{fg}) \quad (1)$$

The variables  $w_{fg}$  and  $b_{fg}$  represent the forget gate's weights and bias,  $hi_{t-1}$  represents the hidden state from before time  $t$ ,  $a_t$  stands for the most recent input, and  $\sigma$  is the sigmoid activating function.

- **The input entrance:** Identifies the newly stored data in the cell condition.

$$ip_t = \sigma(w_{ip} \cdot [hi_{t-1}, a_t] + b_{ip}) \quad (2)$$

$$\widetilde{Ca}_t = \tanh(w_{ca} \cdot [hi_{t-1}, a_t] + b_{ca}) \quad (3)$$

- **Update the State of the Cell:** A combination of the input parameters and forget gates is used for updating the cell state  $Ca_t$ .

$$Ca_t = f_{gt} \cdot Ca_{t-1} + ip_t \cdot \widetilde{Ca}_t \quad (4)$$

- **A Barrier for Output:** Establishes the output depending on the particular cell state.

$$Op_t = \sigma(w_{op} \cdot [hi_{t-1}, a_t] + b_{op}) \quad (5)$$

$$hi_t = Op_t \cdot \tanh(Ca_t) \quad (6)$$

using the output gate  $Op_t$  and the present time step's adjusted invisible state  $hi_t$ .

### c. Forecasting the number of cars parked

The forecast regarding prospective parking availability is generated by passing the LSTM outcome at the last time phase  $hi_T$  across a dense layer of data.

$$\hat{b} = w_{o} \cdot hi_T + b_{o} \quad (7)$$

In this context,  $w_o$  and  $b_o$  denote the outcome of the layer's biases and weights, respectively, while  $\hat{b}$  stands for the anticipated vacancy rate.

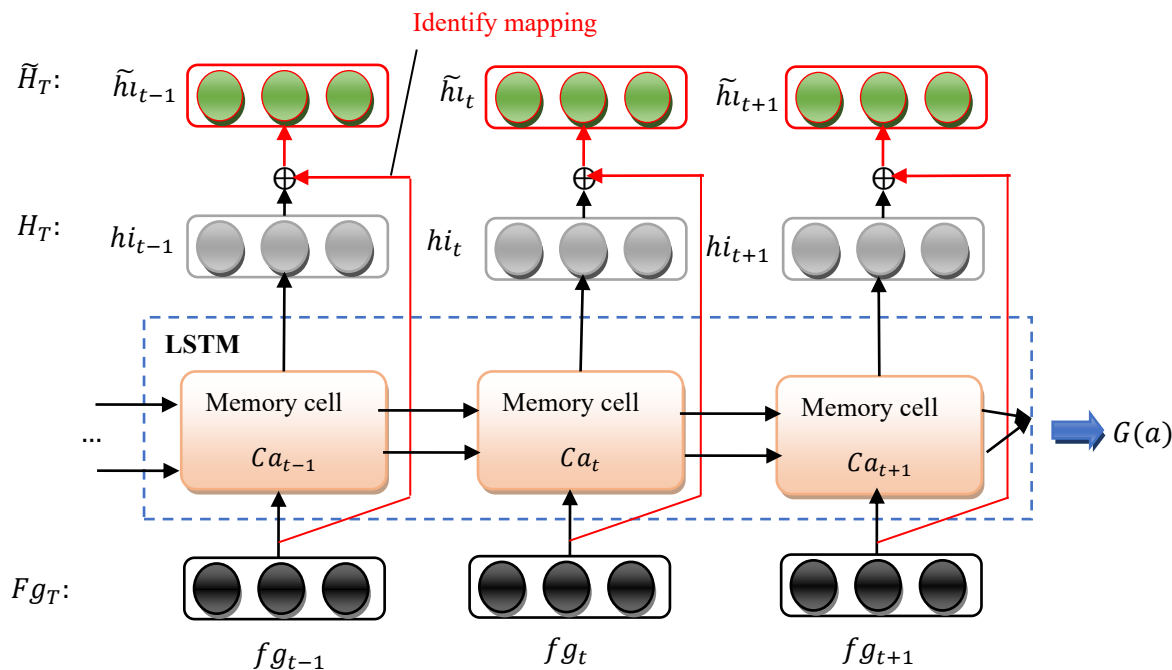
### Optimizing LSTM hyperparameters using DR-RNN

Hyperparameter tuning is applied to the LSTM models with DR-RNN structures to enhance their predictive accuracy. Figure 2 illustrates the configuration of the LSTM integrated with DR.

The design of a Deep Residual RNN utilizes residual connections, which enhance the network's capacity to learn complex patterns over time. A core idea is to improve gradient flow and enable the training of deeper networks by allowing the system to learn the residual mapping instead of the full mapping. Below is the mathematical description of how residual connections are incorporated and how hyperparameters are tuned quantitatively:

The output for a residual connection at layer  $la$  is specified as:

$$hi_t^{(la)} = fg(hi_t^{(la-1)}) + hi_t^{(la-1)} \quad (8)$$



**Figure 2: A look at the LSTM architecture with DR**

In this case, the LSTM applies the changes at layer  $la$ , denoted as  $fg(hi_t^{(la-1)})$ , and  $hi_t^{(la-1)}$  is the result from the layer before it.

Each layer of a multi-LSTM stack with leftover connections applies its relationship to the next:

$$hi_t^{(la)} = LSTM(hi_t^{(la-1)}) + hi_t^{(la)} \quad (9)$$

This method allows the neural network to establish more intricate relationships while also reducing the impact of the problem of decreasing gradients.

It is usual practice to utilize the MSE (i.e, Mean Squared Error) when attempting to forecast parking rates of occupancy:

$$Loss = \frac{1}{Nt} \sum_{i=1}^{Nt} (b_i - \hat{b}_i)^2 \quad (10)$$

In this context,  $Nt$  denotes the quantity of sample training specimens,  $b_i$  stands for the actual occupancy, and  $\hat{b}_i$  denotes the assumed occupancy.

After adding the output of the preceding layer, the hues used for returning propagation via the remaining interconnections are modified:

$$\frac{\partial La}{\partial hi_t^{(la-1)}} = \frac{\partial La}{\partial hi_t^{(la)}} + \frac{\partial fg(hi_t^{(la-1)})}{\partial hi_t^{(la-1)}} \quad (11)$$

There are L layers here. To maximize the LSTM model's forecasting effectiveness for parking capacity prediction, this statistical formulation is used as a foundation for tweaking the model using leftover connections.

### Process flow

The process begins when the customer enters their preferred trip date and destination into the app. If the entered date is more than one day in the past, a warning will be displayed. If the date is current or in the future, the system will search for all parking lots within a specified distance from the destination. The procedure varies depending on the desired end date. If the requested date is today, data is retrieved in real-time for parking lots with at least one available spot. For future dates, the availability of parking spots is predicted using the LSTM-DR-RNN model, which also forecasts the occupancy rates.

The forecasts focus on parking lots that have at least one empty spot. The system then provides the user with a list of parking options, prioritizing those with the highest likelihood of availability, along with the predicted occupancy rate. Additionally, the distance to the destination on foot is considered to help users choose the most convenient option. Figure 3 illustrates the step-by-step process, while Table 3 contains the pseudocode for the LSTM-DR-RNN model.

**Table 3: Pseudocode of LSTM-DR-RNN-based parking availability prediction**

<p><b>Algorithm:</b> LSTM-DR-RNN for Parking Availability Prediction</p> <p><b>Input:</b> Target address TA, Target time t, Current time n, Distance threshold m, Historical parking data HP, IBB Map API, CB-SPS</p> <p><b>Output:</b> Predicted occupancy rates for Parking Locations (PL) near TA at time t</p> <ol style="list-style-type: none"> <li>1. Initialize list of parking locations PL = []</li> <li>2. Retrieve all PLs within distance m of target address A using IBB Map API</li> <li>3. For each PL <math>a_j</math> in the retrieved list:                     <ol style="list-style-type: none"> <li>a. If <math>t &lt; n</math> (target time is in the past):</li> </ol> </li> </ol>
--

i. Return error message "Past dates are not supported"

b. Else:

i. Query total capacity  $X$  and current empty capacity  $Y$  of  $a_j$  using CB-SPS

ii. If  $t == n$  (target time is current time):

1. Get real-time occupancy  $RO = (X - Y) / X$
2. If  $Y > 0$ , add  $a_j$  to PL with RO

iii. If  $t > n$  (target time is future):

1. Prepare input features  $Fe = [HP, X, TA, t]$
2. Pass features  $Fe$  through the LSTM-DR-RNN model:
  - For each LSTM layer  $l$  in  $1 \dots L$ :
    - a. Compute LSTM output  $Hi_l = \text{LSTM}(Fe)$
    - b. Add residual connection  $Hi_l = Hi_l + Fe$
    - c. Update  $Fe = Hi_l$  (pass output to the next layer)
  - Pass the final output through deep recurrent layers to refine the prediction
3. Get the predicted occupancy rate RO from the final output
4. If  $RO < 1$ , add  $a_j$  to PL with predicted occupancy RO

4. Sort list PL based on walking distance to TA and RO

5. Return a sorted list of recommended parking locations PL

End Algorithm

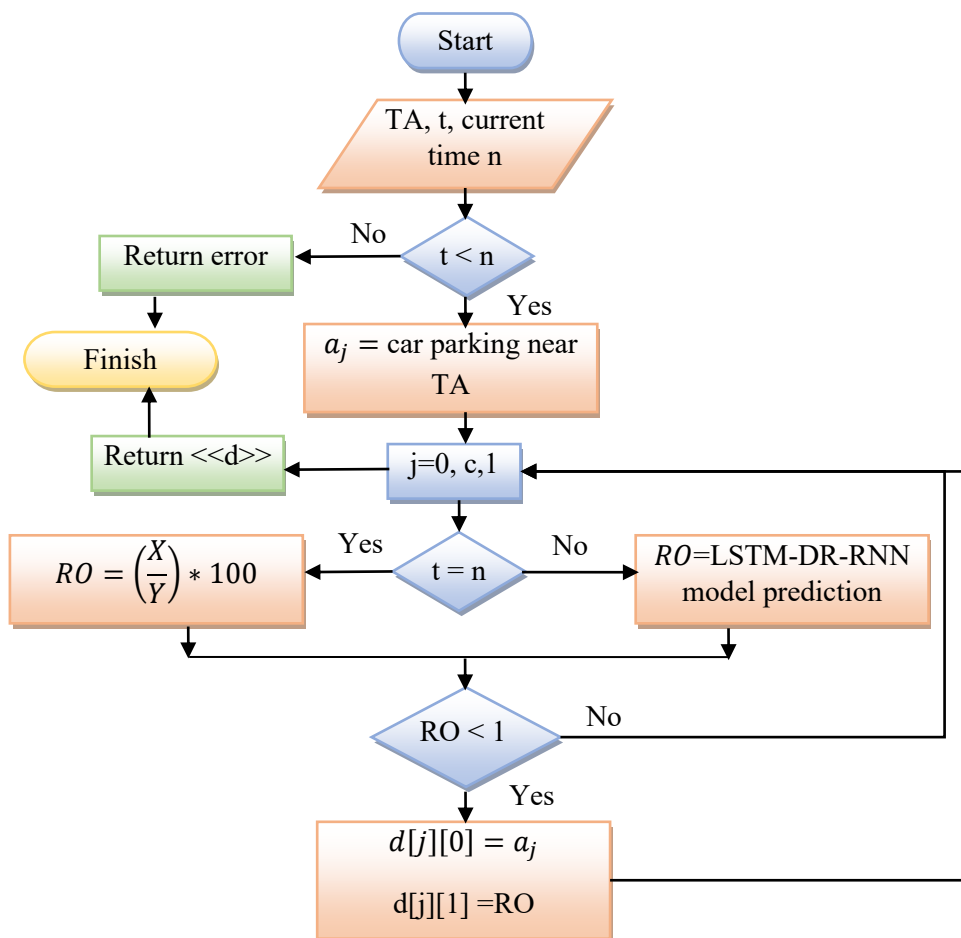


Figure 3: The LSTM-DR-RNN algorithm's flow process for parking lot forecast

### 3. RESULTS AND DISCUSSION

The proposed LSTM-SR-RNN model was evaluated against LSTM [9], ANFIS-DL [21], and ML [14] models to demonstrate its effectiveness and performance. Several metrics, including Accuracy, Root Mean Square Error (RMSE), Mean Squared Error (MSE), and Mean Absolute Error (MAE), were used to assess the performance of each prediction model. These metrics helped identify the best-performing model. For instance, RMSE measures the model's deviation, with smaller values closer to zero indicating better accuracy.

Each model was trained and evaluated across three different scenarios to ensure a comprehensive assessment. These scenarios included LSTM, ANFIS-DL, ML, and LSTM-DR-RNN models. The first scenario involved training on 70% of the dataset and testing on 30%. The second scenario had 80% of the dataset for training and 20% for testing. Finally, the third scenario utilized 90% for training and 10% for testing. The results for these scenarios are presented in Table 4. Additionally, each model was

trained for 100 epochs in all cases: LSTM, SVM, RF, and ARIMA. Performance was measured using Accuracy, MAE, MSE, and RMSE.

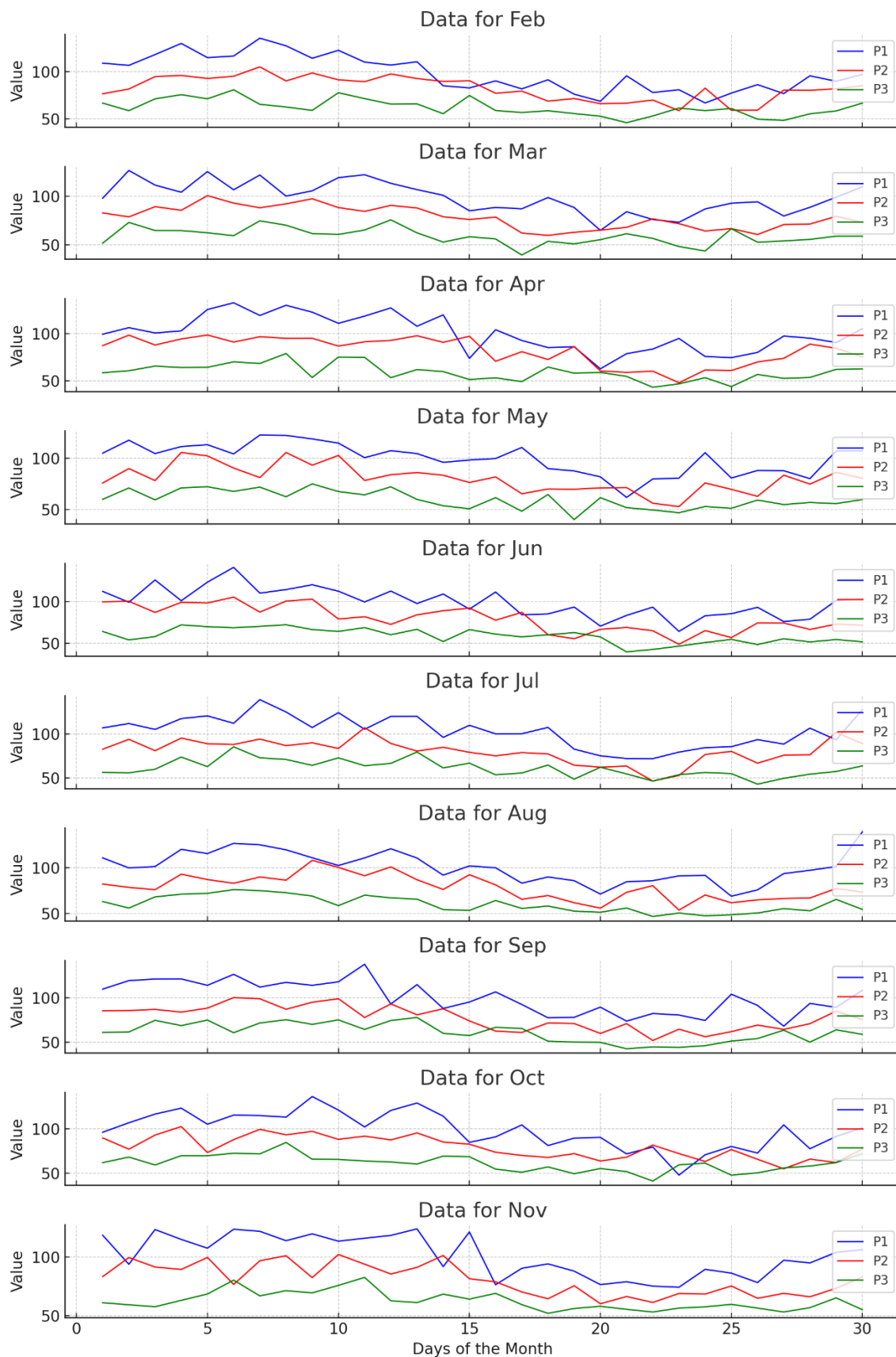
## ii. Realtime dataset

To assess the prediction model's effectiveness, three typical parking lots—designated P1, P2, and P3—were examined. The parking limits of these lots are 157, 132, and 151, respectively, making them small- to medium-sized off-road facilities. These parking lots are located right next to each other, as seen in Figure 4. Cross-referencing between the datasets is made possible by the fact that variations in Net Available Parking (NAP) values result from variations in the geographical region and the public assets surrounding it.



**Figure 4: Geographical Context and Environment of the Examined Parking Lots**

Because of its proximity to a subway station, P1 has the largest demand of all of these, which causes noticeable variations in NAP, as seen in Table 2 and Figure 5. Due to their proximity, P2 and P1 have comparable traits and transportation accessibility, therefore when P1 reroutes traffic, there is no difference. However, because of its functional significance in the area, P3, which is a little further away, comes in second in terms of commercial activity. Each parking lot is normally open from 8:00 to 23:59, with maintenance hours of 0:00 to 07:59. Except in certain situations, vehicle entry is restricted.



**Figure 5: NAP of P1, P2, and P3 in the year 2024**

**Table 2: validation of data and errors**

Parking slot	$N_{p,T}$	$N_{p,C}$	Error
P1	152	152	0
P2	131	131	0
P3	150	150	0

Vehicle arrival and departure timings are precisely captured by entrance and exit cameras, which the management system stores as parking records. The net vehicle influx over a period  $\Delta t$ , from  $t_0$  to  $t_1$ , as follows:

$$q\Delta t = q_{in}\Delta t - q_{out}\Delta t \quad (12)$$

Given the NAP at time  $t_0$ , denoted as  $N_{t_0}$ , the NAP at time  $t_1$  is determined as:

$$N_{t_1} = N_{t_0} - q\Delta t \quad (13)$$

The initial NAP for each parking lot at time  $t_0$ , represented as  $N_{p,t_0}$ , is given by:

$$N_{p,t_0} = Q_p - n_{p,t_0} \quad (14)$$

where  $Q_p$  is the total parking capacity, and  $n_{p,t_0}$  is the number of vehicles present at closing time on February 5, 2024.

On November 2, 2024, a validation point was created to compare the calculated and actual NAP values, represented by the notation  $N_{p,T}$  and  $N_{p,C}$ , respectively, in order to confirm the accuracy of the data. The computed NAP for a specific parking lot  $p$  at a certain time is represented by  $N_{p,C}$ . It is obtained by applying the suggested algorithm, which is based on vehicle entry and exit records. The accuracy of the dataset preprocessing approach is evaluated by comparing it to  $N_{p,T}$ , the real NAP counted manually during validation. When  $N_{p,C}$  roughly resembles  $N_{p,T}$ , the algorithm is processing and estimating parking availability accurately. Data consistency was checked by manual field verification, as indicated in Table X. Sequence durations of 525,600 were obtained by processing the raw data from P1, P2, and P3 using the suggested algorithm to create time-series datasets that represented per-minute NAP values throughout 2024. The statistics are representative of typical parking facilities in the region since they show similar trends, opposites, and variance due to identical geographical and operational factors.

Table 3 lists the primary attributes of the datasets. The true dataset P2 has the lowest dispersion, according to a table analysis, suggesting rather consistent data changes. Additionally, the order of

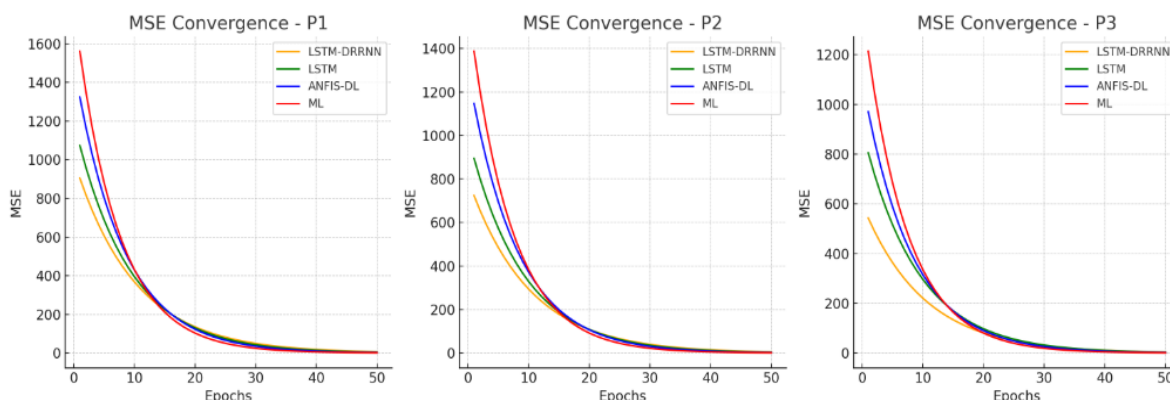
magnitude of the sample data and the ranges and dynamic ranges of P1, P2, and P3 show commonalities, allowing for repeated verification to evaluate the prediction model's stability.

**Table 3: datasets characteristic**

Datasets	Attribute	min	max	Standard deviation (SD)
P1	Real	0	152	58.5
P2	Real	0	132	29.0
P3	Real	0	151	42.2

**Accuracy and Efficiency Comparison among parking models**

Both the TCN and LSTM networks achieved good accuracy on the real and synthetic datasets, as shown in Table 4. Figure 8 shows the MSE convergence curves for LSTM and TCN. Within 50 epochs, both approaches quickly converged, and after 100 epochs, they stabilized. This finding implies that both strategies are effective for datasets with identical features, shifting trends, and different sample sizes. Using three datasets (P1, P2, and P3) for a smart parking system, the graph compares the error convergence of four predictive approaches: LSTM-DRRNN (Proposed Model), LSTM, ANFIS-DL, and ML. The suggested LSTM-DRRNN model outperforms the conventional LSTM, ANFIS-DL, and ML approaches, which exhibit relatively greater error values and slower learning rates, in terms of both speed and accuracy. Zoomed-in insets also show how LSTM-DRRNN quickly lowers MSE in the early training stages, suggesting a better capacity to grasp the intricate patterns of parking availability.



**Figure 6: Convergence curves for smart parking schemes using P1, P2 and P3 datasets**

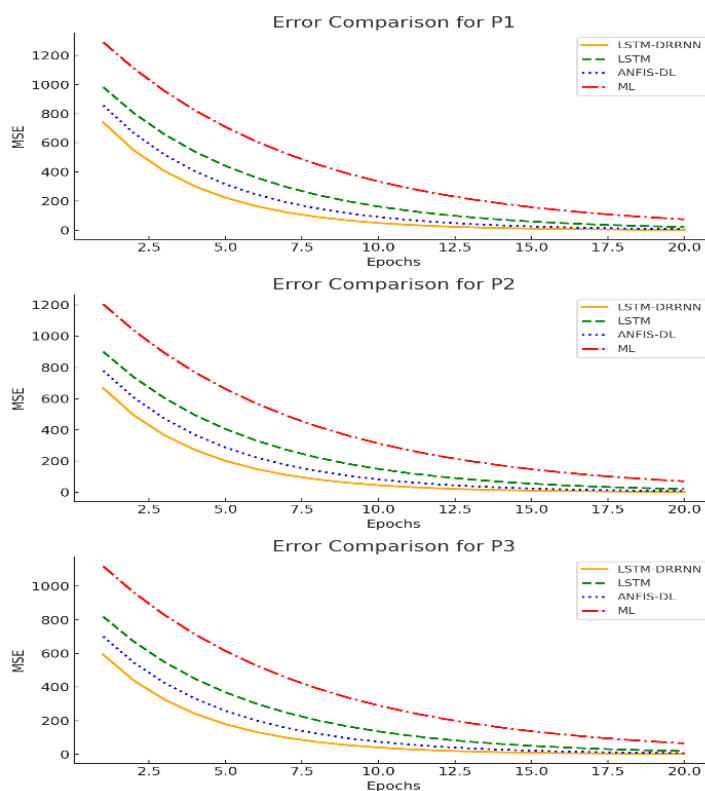
The accuracy of the predictions was improved via LSTM-DRRNN. When comparing the test set MSE of the two networks, Table 4 demonstrates that LSTM-DRRNN outperformed LSTM in terms of

prediction accuracy on P1 and P2. All things considered, the average MSE of LSTM-DRRNN was 23% lower than that of the conventional LSTM, suggesting a significant increase in accuracy brought about by the network upgrade. In fact, LSTM-DRRNN outperformed LSTM by 48% in the P1 dataset.

**Table 4: Prediction accuracies of smart parking schemes**

Models	P1	P2	P3	Mean of P1 &P2
LSTM-DRRNN	0.0145	0.5211	0.0321	0.0336
LSTM	0.0279	0.0435	0.0005	0.0445
ANFIS-DL	0.0578	0.0524	0.0333	0.0436
ML	0.7912	0.3321	0.3423	0.5612

The suggested LSTM-DRRNN's rate of convergence was accelerated. On datasets P1, P2, and P3, LSTM-DRRNN achieved a steady state earlier than LSTM and a lower MSE within the first five epochs, as shown in Figure 6. On the other hand, it took 10–40 runs for the MSE of the traditional LSTM to drop to the same level as that of LSTM-DRRNN.



**Figure 7: Error curves for datasets P1, P2 and P3**

iii. **Istanbul Park dataset**

Its available through the Istanbul Metropolitan Municipality's (IMM) open data site, provides access to a wealth of datasets related to financial markets, emergency response, energy, and more. The underlying datasets are available in various formats such as XML, XLSX, CSV, and through an API, and are licensed under the Apache Data License CC BY 4.0 by the local government of Turkey's capital city. Utilizing Database Server 2014, the dataset includes over 4.5 million recorded data points on parking facilities from 731 different locations across the city. The data collection process spanned an entire year.

To enhance the performance of the models, the Istanbul parking data is processed using feature engineering techniques. This includes extracting and transforming key characteristics. Time-based trends in vehicle usage are captured through chronologies, which help identify temporal patterns such as the time of day, days of the week, and specific events. Geospatial data, like coordinates or neighborhood names, is encoded using location-specific attributes. To allow the algorithms to identify time-dependent patterns, lag functions are created to incorporate historical access data. Additionally, contextual factors, such as weather conditions or traffic flow points, are included when available, to account for variables that influence parking demand. This comprehensive approach significantly improves the model's ability to predict parking availability. The metrics for the Turkish dataset are presented in Table 5.

**Table 5: Parameters of the Istanbul dataset and description**

Parameters	Data type	Description
Location	String	Information about a car park's location indicates the specific area it is situated in. A single location may include one or multiple car parks, all associated with that particular area.
Car park ID	Integer	Includes the car park ID, which is a distinct and unique identifier.
Car park type	String	The type of car park, which can be on-street, indoor, or outdoor.
Car park name	String	Contains carpark name info

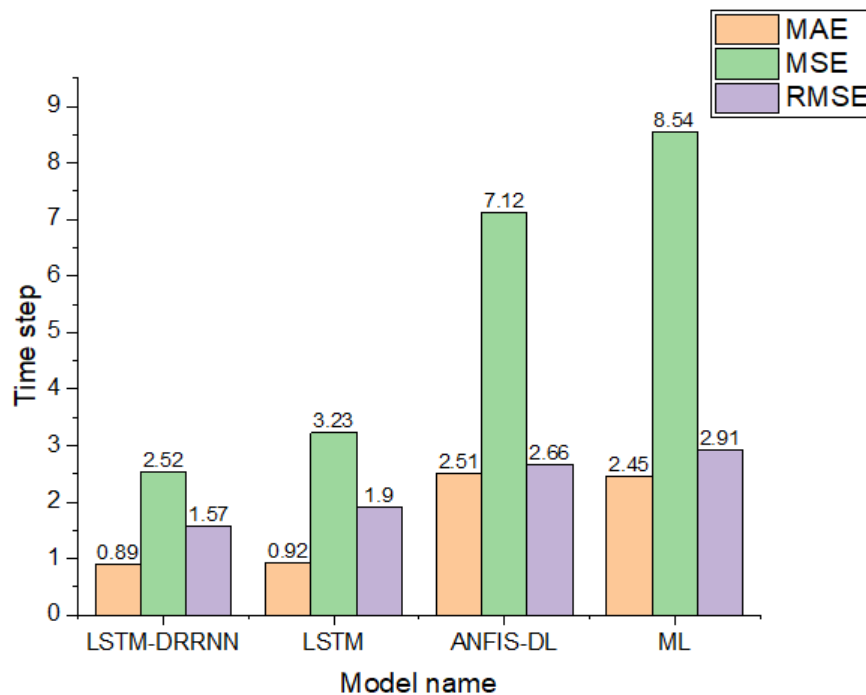
Country	String	A country carpark is located in
Latitude	Double	Latitude car park name info
Longitude	Double	Longitude info of carpark
Capacity	String	The total capacity of a car park
Date of update	Date-time	Date and time of information update
Business hours	String	Business hours of carpark
Address	String	Full address of carpark
Distance	Double	Distance to carpark
Tariff	List<>	Information on tariff

Table 6 indicates that, across all three cases with different training-test splits, the LSTM-DR-RNN model outperformed the other models. It demonstrated superior accuracy in predicting parking spot occupancy, with an MAE of 0.89, MSE of 2.52, and RMSE of 1.57 in its best result. Compared to traditional models, the LSTM-DR-RNN model exhibited significantly reduced error rates.

**Table 6: Error valuation of parking predicted models**

Training and testing (%)	Model name	MAE	MSE	RMSE
70-30	LSTM-DRRNN	0.89	2.52	1.57
	LSTM	0.92	3.23	1.90
	ANFIS-DL	2.51	7.12	2.66
	ML	2.45	8.54	2.91
80-20	LSTM-DRRNN	0.98	3.12	1.77
	LSTM	1.10	3.43	2.21
	ANFIS-DL	2.81	5.09	2.96
	ML	3.51	6.96	4.24
90-10	LSTM-DRRNN	1.12	3.18	1.78

	LSTM	1.31	3.87	2.13
	ANFIS-DL	2.43	5.12	3.02
	ML	2.65	8.43	4.14

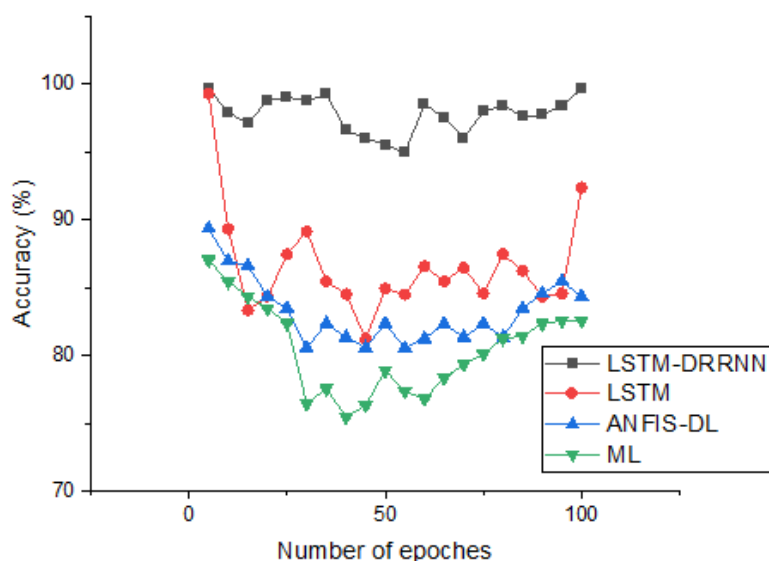


**Figure 8: Valuing parking prediction models with errors**

The evaluation results of each model's accuracy in predicting vehicle parking locations are shown in Figure 8. The proposed LSTM-DR-RNN model achieved lower error rates compared to existing models. Its resistance to overfitting and ability to capture complex temporal relationships make it a strong competitor to LSTM, ANFIS-DL, and traditional ML models. The Deep Residual (DR) connections in the LSTM-DR-RNN enable cleaner gradient flow during training, which helps preserve information over long sequences, making it well-suited for reliable time-series data modeling, where past patterns significantly influence future predictions. The design of LSTM-DR-RNN allows it to avoid the vanishing gradient problem that typically affects regular LSTMs in deep networks, enabling it to learn complex relationships between new and old data points. In comparison to ANFIS-DL and other ML models, LSTM-DR-RNN consistently produces more accurate predictions with lower MAE, MSE, and RMSE values. This is attributed to the residual connections, which help mitigate the impact of data noise, improving overall model performance.

**Table 7: Accuracy comparison of parking predicted models**

Features	Accuracy (%)			
	LSTM-DRRNN	LSTM	ANFIS-DL	ML
Capacity, time, density, day, traffic flow, holiday	99.6	99.3	89.34	87.01
Capacity, time, density, day, traffic flow	97.91	89.32	86.95	85.43
Capacity, time, density, day	97.06	83.33	86.60	84.34
Capacity, time	98.8	84.34	84.34	83.45
Capacity, time, density, day, holiday	99.02	87.43	83.45	82.34
Capacity, time, density, holiday	98.76	89.12	80.56	76.45
Capacity, time, holiday	99.23	85.45	82.34	77.56
Capacity, holiday	96.56	84.5	81.34	75.45
Capacity	96.01	81.2	80.56	76.32
time, density, day, traffic flow, holiday	95.5	84.9	82.34	78.9
density, day, traffic flow, holiday	94.91	84.48	80.54	77.34
day, traffic flow, holiday	98.5	86.56	81.23	76.78
traffic flow, holiday	97.5	85.45	82.32	78.34
holiday	96.01	86.45	81.34	79.34
traffic flow	97.98	84.56	82.34	80.12
density	98.4	87.45	81.34	81.23
day	97.6	86.23	83.45	81.43
time	97.7	84.34	84.56	82.34
time, density, day	98.4	84.56	85.45	82.56



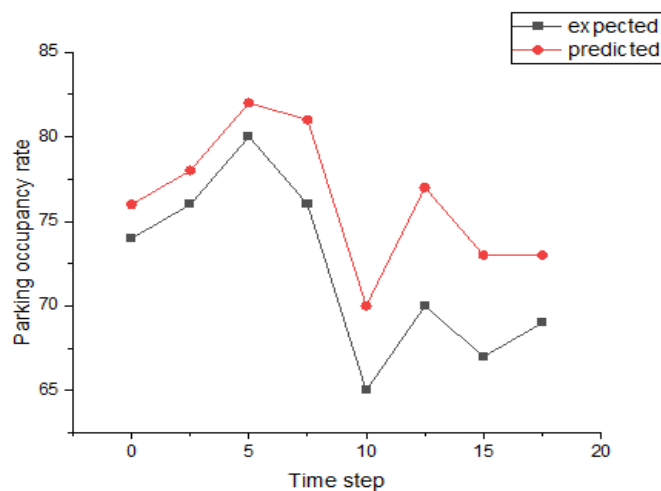
**Figure 9: Accuracy evaluation performance among car parking prediction model**

Capacity, density, time, day, traffic flow, and holiday were the key features that enabled the model to achieve a maximum accuracy rate of 99.6%, as shown in Table 7. Among these, time and day were the most important factors in improving prediction performance. The accuracy was lowest in the single-feature tests, where only density was considered. Despite this, Table 6 demonstrates that prediction accuracy remains within a reasonable range even with a reduced number of attributes, highlighting that this peak accuracy guides the parameter selection.

Figure 9 illustrates the combinations of the LSTM-DR-RNN model used to estimate parking occupancy rates. The LSTM-DR-RNN stands out among these models, outperforming standard LSTM, ANFIS-DL, and other conventional machine learning (ML) models in occupancy prediction. Its advanced design, which integrates LSTM with Deep Residual (DR) layers, results in significantly improved performance. One common issue with deep networks is vanishing gradients, which is addressed by the residual layers, ensuring smoother gradient and information flow during backpropagation. Accurate time-series forecasts, such as parking occupancy, require the model to learn and retain essential patterns over extended periods, as both current and past data influence demand.

The complex, non-linear interactions in parking occupancy data are especially well modeled by the LSTM-DR-RNN. LSTM is able to capture time-dependent relationships, and the ongoing connections are robust. Residual connections help prevent overfitting, which is more common in traditional ML and standard LSTM models, particularly when working with smaller datasets. These connections ensure the model does not overfit to noise in the training data. In contrast, ANFIS-DL and other ML models struggle with capturing both short-term fluctuations and long-term dependencies in parking

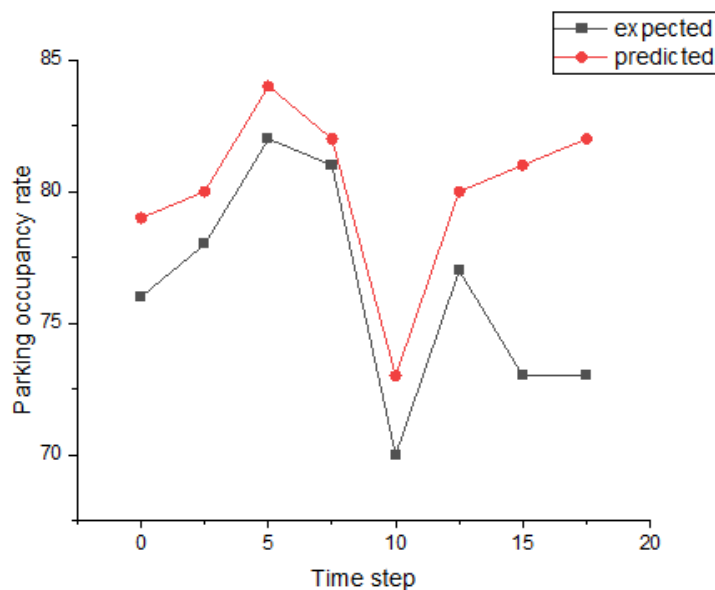
demand due to their limited capacity for long-term memory. While ANFIS-DL handles non-linear patterns adequately, LSTM-DR-RNN offers a much deeper understanding of temporal dynamics. Basic ML models, while effective with simpler connections, require extensive feature engineering to handle complex sequential data. Figure 10 shows that the LSTM-DR-RNN model delivers highly accurate occupancy predictions, making it the best choice for tackling unpredictable, day-to-day parking occupancy challenges.



**Figure 10: ML algorithm-based parking occupancy rate**

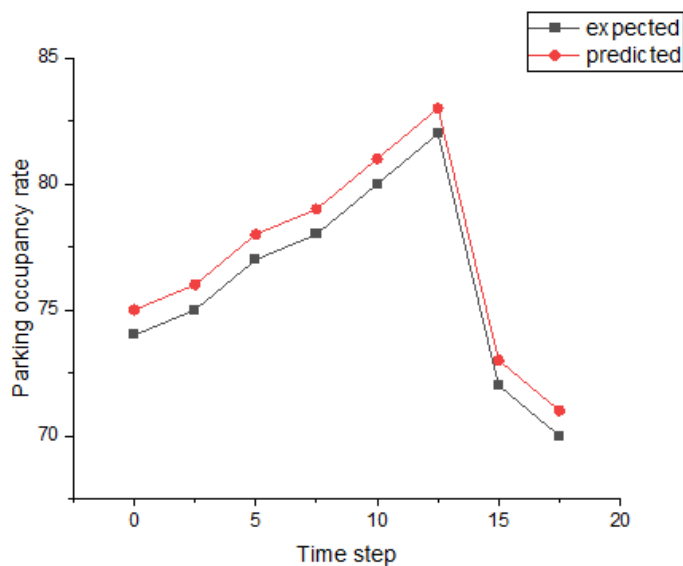
Figure 10 presents the results of the ML model's attempt to forecast parking spot utilization rates, with actual values shown in blue and predicted values in red. The model demonstrates good prediction accuracy when occupancy rates follow a steady, linear trend, effectively capturing regular patterns. However, its forecasts become less precise during abrupt changes, particularly at high and low points or during rapid shifts. This limitation arises because ML models generally perform better with consistent, continuous data but struggle to capture sudden fluctuations and extreme outcomes, leading to larger prediction errors.

Figure 11 illustrates the efficacy of the ANFIS-DL method in predicting parking space utilization rates. The ANFIS-DL model effectively captures consistent linear trends in occupancy rates, performing well with smooth and predictable variations. However, it struggles with abrupt changes, particularly during large swings or fluctuations. While ANFIS-DL is capable in stable scenarios, it falls short when faced with patterns exhibiting significant variation or sudden shifts, which require a deeper understanding of time-based or recurring structures. As a result, the model becomes less accurate at predicting occupancy during peak periods, troughs, and sharp fluctuations, leading to larger discrepancies between expected and actual values.

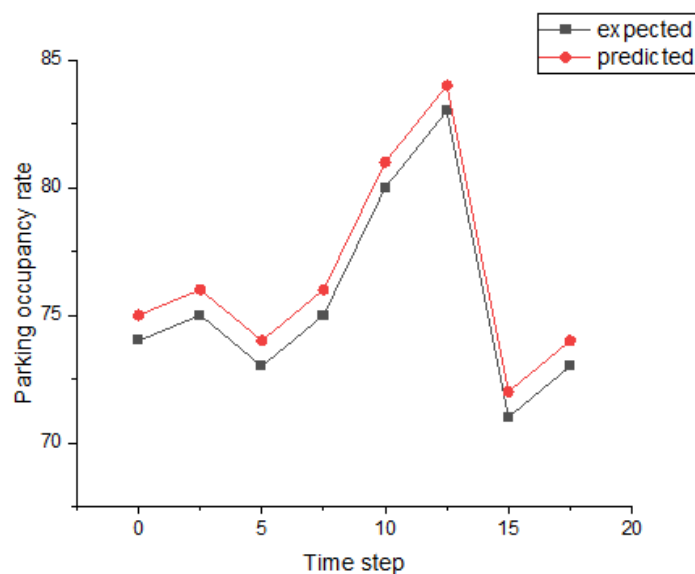


**Figure 11: Rate of parking utilization using the ANFIS-DL methodology**

Figure 12 shows the performance of the LSTM model in predicting parking space occupancy rates. The LSTM model excels at capturing trends and variations in occupancy, performing well in stable scenarios. However, it has limitations when dealing with unexpected changes in occupancy patterns, especially during dramatic spikes or dips. This could be attributed to the LSTM's reliance on long-term interconnections, which, while effective for capturing broader trends, may not respond quickly enough to sudden shifts in real-world conditions.



**Figure 12: Parking rate of occupancy calculated using the LSTM technique**



**Figure 13: Probability of parking availability using the LSTM-DR-RNN method**

Figure 13 shows the accuracy of the parking spot occupancy rate predictions made by the LSTM-DR-RNN model. The comparison of the model's predictions to actual occupancy data demonstrates its ability to track both gradual increases and decreases in parking demand. The model integrates LSTM layers, which excel at understanding temporal trends, with residual connections that help prevent the accumulation of errors. This combination makes the LSTM-DR-RNN model a reliable choice for predicting carpark availability, especially in real-world scenarios where patterns can change rapidly, including during sudden spikes or reductions in demand.

#### 4. CONCLUSION

The integration of Long Short-Term Memory (LSTM) and Deep Residual Recurrent Neural Networks (DR-RNN) into the Reservation-based Smart Parking System (SPS) marks a significant advancement in addressing parking challenges in densely populated cities, particularly during peak times. This hybrid model leverages the strengths of both LSTM and DR-RNN architectures, combining the ability of LSTM to capture temporal dependencies in time-series data with the deep residual connections of DR-RNN that prevent the vanishing gradient problem, enabling the network to better learn complex patterns over extended periods. The LSTM-DR-RNN model operates by predicting parking spot occupancy with remarkable accuracy. The model considers a range of factors such as weather conditions, traffic flow, and event schedules, which impact parking availability. By integrating these features into the prediction system, the model can deliver real-time forecasts that are highly reliable, with an impressive accuracy rate of 99.6% in some cases. The metrics used to evaluate the model's

performance, including Mean Absolute Error (MAE) of 0.89, Mean Squared Error (MSE) of 2.52, and Root Mean Square Error (RMSE) of 1.57, indicate its precision in predicting parking availability. These results show that the LSTM-DR-RNN can efficiently reduce errors in predicting parking occupancy, which is crucial for city-wide parking management. Future work aims to refine the model further by incorporating additional data sources, such as local events and public transportation schedules, to enhance prediction accuracy.

## References

1. Nassereddine, M., & Khang, A. (2024). Applications of Internet of Things (IoT) in smart cities. In *Advanced IoT technologies and applications in the industry 4.0 digital economy* (pp. 109-136). CRC Press.
2. Ke, R., Zhuang, Y., Pu, Z., & Wang, Y. (2020). A Smart, Efficient, and Reliable Parking Surveillance System with Edge Artificial Intelligence on IoT Devices. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 1–13.
3. Khalid, M., Wang, K., Aslam, N., Cao, Y., Ahmad, N., & Khan, M. K. (2021). From smart parking towards autonomous valet parking: A survey, challenges and future Works. *Journal of Network and Computer Applications*, 175.
4. Kokate, R., Kohad, D., Hiwarkar, A., Godbole, P., & Bhasarkar, R. (2024). Real Time Parking System using ML.
5. Awan, F. M., Saleem, Y., Minerva, R., & Crespi, N. (2020). A comparative analysis of machine/deep learning models for parking space availability prediction. *Sensors*, 20(1), 322.
6. Mei, Z., Zhang, W., Zhang, L., & Wang, D. (2020). Real-time multistep prediction of public parking spaces based on Fourier transform-least squares support vector regression. *Journal of Smart Transportation Systems*, 24(1), 68-80.
7. Lyu, M., Ji, Y., Kuai, C., & Zhang, S. (2024). Short-term prediction of on-street parking occupancy using multivariate variable based on deep learning. *Journal of Traffic and Transportation Engineering (English Edition)*, 11(1), 28-40.
8. Ye, X., Wang, J., Wang, T., Yan, X., Ye, Q., & Chen, J. (2020). Short-term prediction of available parking space based on machine learning approaches. *IEEE Access*, 8, 174530-174541.
9. Canli, H., & Toklu, S. (2021). Deep learning-based mobile application design for smart parking. *IEEE Access*, 9, 61171-61183.
10. Xie, J., He, Z., & Zhu, Y. (2023). A DRL based cooperative approach for parking space allocation in an automated valet parking system. *Applied Intelligence*, 53(5), 5368-5387.

11. Nardin, A., & D'Andreagiovanni, F. (2024). A Quantum-Inspired Ant Colony Optimization Algorithm for Parking Lot Rental to Shared E-Scooter Services. *Algorithms*, 17(2), 80.
12. Wang, Y., Liu, D., & Sun, X. (2024). A low-cost, high-performance middleware solution for unified parking management. *Soft Computing*, 28(3), 2291-2308.
13. Rafique, S., Gul, S., Jan, K., & Khan, G. M. (2023). Optimized real-time parking management framework using deep learning. *Expert Systems with Applications*, 220, 119686.
14. 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.
15. Errouso, H., El Ouadi, J., & Benhadou, S. (2022). Dynamic parking space allocation at urban scale: Problem formulation and resolution. *Journal of King Saud University-Computer and Information Sciences*, 34(10), 9576-9590.
16. 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.
17. Wang, Y., & Chen, Q. (2022). Dynamic parking allocation model in a multideestination multiple parking lot system. *IEEE SmartTransportation Systems Magazine*, 14(5), 195-208.
18. Mei, Z., Zhang, H., Tang, W., & Zhang, L. (2023). Assessment and optimization of parking reservation strategy for Park-and-Ride system emissions reduction. *Transportation Research Part D: Transport and Environment*, 124, 103956.
19. Yan, P., Bai, M., Cai, X., Chen, Z., & Xie, H. (2023). Data-driven dynamic optimization for real-time parking reservation considering parking unpunctuality. *IISE Transactions*, 1-14.
20. Guo, F., Xu, H., Xu, P., & Guo, Z. (2024). Design of a reinforcement learning-based smartcar transfer planning system for parking lots. *Mathematical Biosciences and Engineering*, 21(1), 1058-1081.
21. Elomiya, A., Křupka, J., Jovčić, S., & Simic, V. (2024). Enhanced prediction of parking occupancy through fusion of adaptive neuro-fuzzy inference system and deep learning models. *Engineering Applications of Artificial Intelligence*, 129, 107670.
22. Li, Q., Cheng, J., & Chen, L. (2024). Research on shared parking allocation considering the heterogeneity of parking slot providers' temporary parking demand. *Transportation Letters*, 1-13.