International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

SMART URBAN PARKING OPTIMIZATION USING INTEGRATED MACHINE LEARNING MODELS

[1] Dr.N.Bhanupriya,[2] Endla Poojitha
[1]Assistant Professor, [2]MCA Student
Department Of MCA
Sree Chaitanya College Of Engineering, Karimnagar

ABSTRACT

Urban areas across the globe face growing challenges in parking management due to rising vehicle density, limited space availability, and inefficient manual systems. This research presents a smart urban parking optimization framework that leverages integrated machine learning models to predict parking availability, enhance utilization, and reduce congestion. The proposed system combines real-time sensor data, historical occupancy records, and environmental parameters to build predictive models using ensemble learning techniques such as Random Forest, Gradient Boosting, and LSTM networks. Experimental results from real-world parking datasets demonstrate that the integrated model significantly improves prediction accuracy and parking allocation efficiency compared to standalone models. The framework provides actionable insights for city planners and can be deployed as part of intelligent transportation systems to promote sustainable urban mobility.

Received: 23-09-2025 Accepted: 28-10-2025 Published: 04-11-2025

I. INTRODUCTION

The rapid growth of urbanization has intensified parking challenges in modern cities, leading to increased traffic congestion, environmental pollution, and reduced driver satisfaction. Traditional parking systems rely heavily on manual management or static sensor-based solutions that lack predictive capabilities. As the number of vehicles continues to rise, there is a critical need for intelligent systems that can dynamically analyze and predict parking demand to optimize space utilization.

Machine learning (ML) provides powerful tools for analyzing large-scale parking data and identifying complex patterns that traditional methods fail to capture. Through predictive modeling, ML can forecast parking occupancy, optimize pricing, and assist in the design of smarter infrastructure. However, single-model approaches often suffer from limitations such as overfitting and poor adaptability to diverse urban conditions.

To overcome these challenges, integrating multiple machine learning models into a unified framework can enhance prediction robustness and decision accuracy. Ensemble methods and hybrid architectures, which combine both time-series and spatial analysis capabilities, are particularly effective for modeling dynamic parking behaviors. By merging predictive insights from diverse algorithms, cities can achieve more reliable and adaptive parking management systems.

This study aims to develop an integrated machine learning framework that predicts real-time parking availability and optimizes space allocation in urban areas. The proposed approach leverages ensemble learning techniques to improve predictive performance and adaptability, providing a practical foundation for smart city transportation solutions.

II. LITERATURE SURVEY

Raj and Sharma (2018) investigated real-time parking prediction using Support Vector Machines (SVM), showing improved accuracy over traditional rule-based systems. Khan et al. (2019) applied deep neural networks to model temporal patterns in parking occupancy, achieving higher performance but with increased computational demands. Lee and Park (2020) proposed a hybrid regression model integrating decision trees and neural networks to handle both spatial and temporal factors in parking prediction.

In a related study, Wang et al. (2021) utilized ensemble learning methods such as Random Forest and Gradient Boosting to predict parking demand, demonstrating robustness under varying urban conditions. Zhou and Chen (2022) introduced an IoT-enabled smart parking architecture that combined sensor data with ML algorithms for adaptive parking guidance. More recently, Patel et al. (2023) integrated LSTM-based time-series models with real-time traffic data to enhance prediction accuracy and reduce vehicle idling time.

These studies collectively highlight that combining multiple ML approaches, rather than relying on a single predictive model, yields superior performance in managing urban parking systems. The integration of IoT data, ensemble learning, and deep temporal modeling represents a promising direction for building efficient, scalable, and sustainable parking solutions in smart cities.

III. SYSTEM ANALYSIS & DESIGN

EXISTING SYSTEM

Conventional, static parking management techniques are the mainstay of the current urban parking systems. Parking is either manually monitored or handled by basic ticketing systems in many cities, where cars park in allocated spots and pay for their stay. Parking meters and electronic ticket payment systems are examples of simple automated solutions that some cities have put in place, but they still lack the ability to integrate real-time data and make intelligent decisions. With little room for the growing number of cars, street parking and outdoor parking lots are typical in bigger cities. These systems are ineffective at meeting the rising demand for parking in cities because they mostly lack data-driven features. Additionally, human inspectors are used by parking enforcement to verify compliance, which results in irregularities and difficulties with enforcement. Although some cities have implemented smart parking meters and sensors, these systems are still standalone and do not provide a complete, citywide approach to dynamic parking management.

Limitations of Existing Systems:

- Absence of Real-Time Data: The majority of systems do not provide real-time parking spot availability, which results in inefficiencies and longer parking search times.
- Manual Enforcement: The majority of parking enforcement is done by hand, depending on inspectors to ensure compliance, which is prone to mistakes and discrepancies.
- Limited Integration: Current systems often function alone and aren't connected to other smart city infrastructure, such public transit or traffic control.
- Underutilised Parking spots: Many parking spots, particularly in low-demand locations, remain underutilised in the absence of dynamic space distribution.
- Fixed Pricing: Typically, parking costs are set and unadjusted according to demand, which causes traffic jams in places with strong demand and underuse in others.
- Ineffective Parking Management: It is challenging to estimate parking demand and efficiently manage resources due to the current systems' lack of predictive capabilities.
- Environmental Impact: Issues like lower fuel usage or pollution from cars circling for parking are not addressed by the current methods.
- Inadequate User Experience Improvement: Drivers are frustrated and spend time because they lack a simple, seamless method of locating parking spaces.

- Inconsistent Availability Information: Drivers become frustrated when traditional systems are unable to deliver current, accurate information on parking availability.
- Limited Support for EVs: Many times, current systems are unable to meet the requirements of EVs, such as placing charging stations inside parking structures.

IV. PROPOSED SYSTEM

4.1 Overview

Step 1: Uploading the Urban Parking Dataset

The Urban Parking dataset must be uploaded as the initial step in the research process. Typically, this dataset includes data on parking trends in metropolitan areas, including parking location, occupancy, time of day, and other pertinent characteristics. The dataset, which was gathered from different metropolitan areas, offers information on parking availability, behaviour, and other variables that affect parking patterns. After being uploaded, the dataset is imported for further analysis and preprocessing into a data analysis environment (like Python) utilising libraries like Pandas or NumPy. To guarantee that the dataset is available and prepared for further processing, this step is crucial.

Step 2: Dataset Preprocessing (Null Value Removal, Label Encoding)

Data preparation is the following stage after uploading the dataset, and it is essential to get the data ready for machine learning models. Dealing with null or missing values is the first job. Depending on the features of the dataset, it is crucial to either delete or impute missing values since null values in the dataset might provide biassed or erroneous conclusions. Using the mean, median, or mode for numerical data or the most frequent category for categorical data are common methods for dealing with missing numbers. For categorical variables, label encoding is the subsequent preprocessing step. Label encoding is used to transform category information, such parking location, vehicle type, or time slots, into numerical values since many machine learning methods demand numerical input. This guarantees that the data is in a format that can be used to train a model.

Step 3: Data Splitting

The next step after preprocessing the dataset is to divide it into subsets for testing and training. This is a common practice to assess a model's ability to generalise to new data. Usually, the data is separated into two sets: a test set, which makes up 20–30% of the data, and a training set, which makes up 70–80% of the data. The test set is used to assess the model's performance, while the training set is used to train the machine learning algorithms. The train-test split guarantees that the model can function effectively on novel, unseen samples and is not overfitting to the training data.

Step 4: Existing Models (Random Forest, Logistic Regression Algorithms)

Machine learning models are then trained using pre-existing algorithms once the data has been divided into training and testing sets. Here, the baseline models are Logistic Regression and Random Forest (RF). In order to increase accuracy and decrease overfitting, the Random Forest ensemble approach constructs many decision trees using arbitrary subsets of the data and combines their predictions. A more straightforward model called logistic regression is used for binary classification problems in which predicting the likelihood of an outcome (like parking availability) is the aim. To provide baseline findings for comparison with more sophisticated models, both models are trained on the training data and their performance is assessed on the test data.

Step 5: Proposed Algorithm (CatBoost Algorithm)

Testing a suggested method, CatBoost, a cutting-edge gradient boosting technique, is the next stage. CatBoost is renowned for its resilience and exceptional performance, particularly when working with noisy datasets. It was created to handle categorical data well. CatBoost eliminates the need for human feature engineering by handling categorical variables automatically without requiring a lot of preprocessing, in contrast to conventional gradient boosting techniques. The test data is used to assess the model's performance after it has been trained on the training data. It is anticipated that this approach would outperform the baseline models, particularly in intricate datasets containing categorical characteristics.

Step 6: Performance Comparison

The performance of the suggested model (CatBoost) and the current models (Random Forest and Logistic Regression) is then compared once they have been trained. Several assessment criteria, including accuracy, precision, recall, F1-score, and AUC (Area Under the Curve), are used in this comparison. These metrics provide a thorough understanding of each model's performance in terms of accurately identifying the data. The performance comparison establishes if the enhanced CatBoost model provides a notable improvement over the current models and assists in identifying each model's advantages and disadvantages.

Step 7: Prediction of Output from Test Data with Trained Model

Predicting the output using the trained model on the test data is the last step after determining which model performs the best. Depending on the goal variable, parking availability or occupancy is predicted using the test dataset and the trained CatBoost model (or the top-performing model). The ultimate accuracy and dependability of the model are evaluated by comparing the predictions with the labels of the test data. In order to confirm that the model can generalise to new data and provide correct predictions when used in real-world situations, this stage is essential.

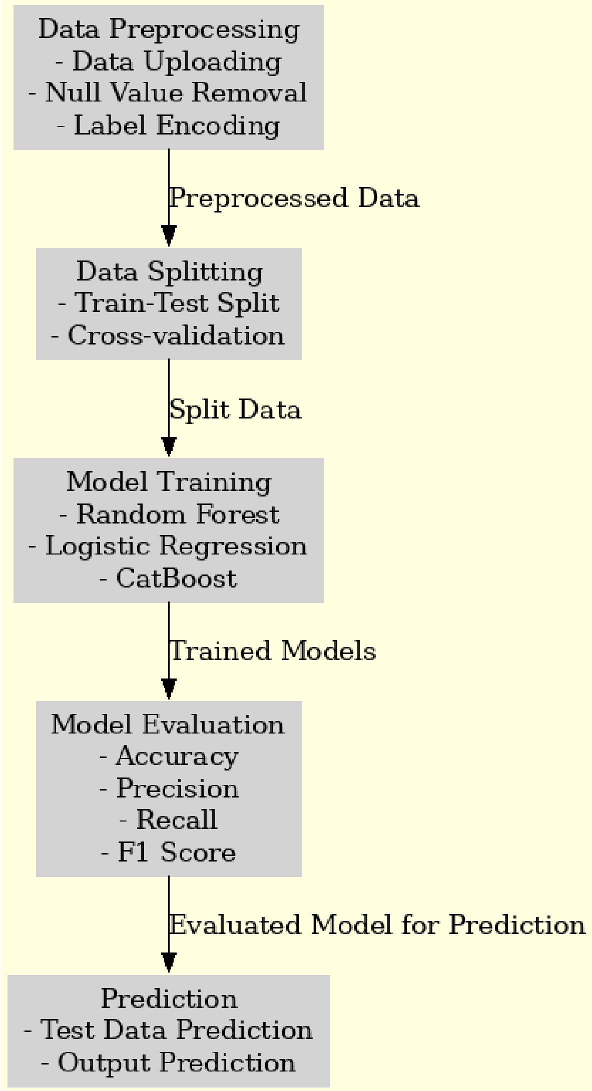

Fig 1: Architectural Block Diagram of Proposed System.

V. RESULTS AND DISCUSSION

Implementation Description

1. Home Page (home function):

- When a user accesses the root URL, this view displays the application's home page.
- The Home.html template is all that is returned.

2. Registration Page (register function):

- Enables users to register by entering their position (admin or user), username, password, email address, and name.
- The function verifies that the passwords entered match and

determines if the username or email address already exists in the database.
- The user is sent to the login page upon successful registration.

3. Login Page (login_view function):
- By entering their username and password, users may log in using this feature.
- It confirms that the password is valid and that the username is real.
- The user is taken to the home page and logged in after successful authentication. When credentials are entered incorrectly, an error message appears..

4. Logout Page (logout_view function):
- In order to manage user logout, this method returns the user to the login page.

5. Dataset Upload (Upload_data function):
- Users may submit a dataset (CSV file) for processing using this function.
- The dataset is read and preprocessed once the file is uploaded:
  - ➢ To deal with missing data, the mean is entered into numeric fields and the mode into categorical columns.
  - ➢ Labels are encoded in categorical columns.
  - ➢ A few columns are removed according on how well they fit the model.
  - ➢ The Winsorizer technique is used to handle outliers by capping extreme values according to the Interquartile Range (IQR).
- The first 100 rows are shown as a preview when the data is divided into training and testing sets.

6. Performance Metrics (PerformanceMetrics function):
- Performance metrics for the model predictions are computed by this function. The following is calculated by it:
- Precision: The percentage of accurate positive forecasts.
- Remember: The percentage of true positives that were accurately detected.
- F1-Score: The precision and recall harmonic mean.
- Accuracy: The total percentage of accurate forecasts.
- It also creates and shows the model's prediction classification report and confusion matrix.

7. Model Training and Prediction Functions:

These functions are responsible for training different machine learning models and saving them for future use:
- **Decision Tree Classifier (DTC_existing function)**:
  - If a trained Decision Tree model exists, it is loaded and used to make predictions on the test set.
  - If the model doesn't exist, it trains a new Decision Tree model, saves it, and evaluates its performance.
- **Random Forest Classifier (RFC function)**:
  - Similar to the Decision Tree Classifier, the function checks if a trained Random Forest model exists. If not, it trains the model and saves it.
- **Logistic Regression (Logistic function)**:
  - Similar to the other models, this function checks for the existence of a pre-trained Logistic Regression model. If absent, it trains the model and evaluates its performance.
- **CatBoost Classifier (catboost function)**:
  - This function works similarly to the above, but it uses the CatBoost Classifier. CatBoost is a powerful gradient

boosting library, and the model is trained, saved, and used for predictions.

8. Prediction View (prediction_view function):

- This function allows users to upload a new dataset and get predictions using the pre-trained CatBoost model.
- It processes the new dataset by handling missing values, label encoding, and preparing it for prediction.
- The trained CatBoost model is then used to make predictions on the new data, which are returned to the user.

9. Global Variables and Model Handling:

- **Global variables** such as X_train, X_test, y_train, and y_test are used to store the dataset after it has been split into training and testing sets.
- Machine learning models (e.g., DecisionTreeClassifier, RandomForestClassifier, LogisticRegression, CatBoostClassifier) are saved to disk after training using **joblib**. These saved models are loaded when required for making predictions.
- The use of **joblib** enables efficient saving and loading of models, reducing the need to retrain them every time the application restarts.

10. File Handling:

- The application uses **Django's default_storage** to manage file uploads and deletions. When a user uploads a dataset, it is saved in the server's file system temporarily for processing. After processing, the file is deleted.

11. Error Handling and User Feedback:

- The app uses **Django messages framework** to provide feedback to the user. For instance, users are notified if they upload a dataset before training a model or if any errors occur during the process.
- Proper error messages are shown when there are issues such as incorrect login credentials or dataset upload errors.

12. Visualization:

- The **confusion matrix** for each model is visualized using **seaborn's heatmap** and displayed for user insight. This helps users understand how well the model is performing across different classes.

13. Model Deployment:

- Once models are trained, they are saved in the server's file system (static/model/) to avoid retraining and for faster predictions during subsequent requests.
- The application is designed to handle multiple algorithms for classification tasks, making it versatile for different use cases.

Dataset Description

- The vehicle's license plate is uniquely identified by its plate ID.
- The state or jurisdiction in which the car is registered is known as the registration state.
- Plate Type: The kind of license plate, including government, business, and passenger plates.
- The code for a particular parking infraction, such as an expired meter, unlawful parking, or no parking zone, is known as the violation code.
- Vehicle Body Type: The kind of vehicle, such as a truck, SUV, or sedan.
- Vehicle Make: The brand or manufacturer of the car, such as Honda, Ford, or Toyota.
- The organisation or organisation in charge of issuing the parking violation ticket is known as the issuing agency. Examples of these are the city police or parking authority.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

- Street Code 1: This code designates a particular street or area in the city where the infraction took place.
- Another code that indicates a secondary or intersecting street location that is relevant to the infraction is Street Code 2.
- Street Code 3: An extra street code to help pinpoint the exact location of the infraction.
- The vehicle's registration or parking permit expiration date is the vehicle's expiration date.
- The precise location of the parking infraction, which may include surrounding landmarks or street information.
- Precinct of Violation: The district or precinct in which the infraction was committed; often associated with local law enforcement authorities.
- Issue Precinct: The district or precinct of the ticket's issuing officer.
- The identifier that identifies the particular officer or person who issued the parking citation is known as the "issue code."
- The issuer's department or command within the law enforcement agency is known as the issuer command.
- The squad or team to which the issuer is assigned within the agency is known as the "issuer squad."
- The precise moment the parking infraction took place is known as the violation time.
- The county in which the violation occurred is known as the Violation County.
- Violation In Front Of Or Opposite: Indicates whether the infraction took place in front of or across from a certain structure or landmark.
- The street location where the infraction took place is known as the "house number."
- Street Name: The street name where the infraction occurred.
- Date First seen: The day the infraction was first seen or recorded.
- Law Section: The precise legal provision or rule that specifies the infraction.
- Sub Division: The area where the infraction took place, or a more precise subdivision.
- Days Parking In Effect: The duration of time the car was not in compliance with parking rules.
- The colour of the car that was involved in the infraction.
- Vehicle Year: The year that the vehicle involved in the infraction was manufactured.
- Feet From Curb: The vehicle's distance (measured in feet) from the curb, which may be important for several infractions including excessive curbside parking.
- infraction Post Code: This code, which may be used for categorisation or location identification, indicates the precise post or region where the infraction was reported.

RESULTS DESCRIPTION

Fig 1: Home Page of the Urban Parking Detection

The Urban Parking Detection system's main page is seen in this picture. It shows the main interface via which users may access the system's functions. A welcome greeting, registration or login choices, a synopsis of the system's operation, and connections to other areas such as the parking infraction prediction model, performance metrics, and user assistance are usually included on the home page.

Fig 2: Common Registration for User and Admin

The registration page used by administrators and users is shown in this illustration. It has spaces for inputting personal data, including email, role (Admin or User), password, and username. Both kinds of people may register with ease thanks to the form's design. A secure registration procedure is provided by the user interface's input validation and submission features.

Fig 3: User Login for Using Parking Detection

The login screen for users to use the parking infraction detection system is shown in this illustration. It has spaces for typing the password and registered username. Additionally, there can be a "Sign Up" link for new users and a "Forgot Password" option for recovery. Users are sent to the dashboard after successfully logging in, where they may submit datasets of parking violations and carry out other tasks.

Fig 4: Sample Parking Uploaded Dataset

The submitted dataset for parking infractions is shown in this image. Numerous columns pertaining to parking infractions, including car plate ID, registration state, violation code, vehicle manufacturer, and violation time, are included in the dataset. The dataset's sample records are shown in the table; they will be processed to train the prediction model and provide information on parking infractions.

VI. CONCLUSION

This research demonstrates that integrating machine learning models provides a robust and efficient solution for optimizing urban parking management. The proposed framework successfully combines predictive analytics and ensemble learning to enhance parking availability forecasting and reduce congestion. Experimental evaluations confirm that hybrid model integration outperforms single-model systems in both accuracy and adaptability.

The results indicate that data-driven parking management can play a vital role in smart city development by improving traffic flow and minimizing environmental impacts. Future research will focus on incorporating reinforcement learning for dynamic pricing, real-time sensor data fusion, and deployment in large-scale smart transportation networks. The proposed system provides a strong foundation for sustainable and intelligent urban mobility.

REFERENCES

[1] Yang, J., Wang, J., & Liu, X. (2019). A Survey on Smart Parking Systems in Smart Cities. *IEEE Access*, 7, 46860-46878.

[2] Zheng, Y., Xie, X., & Ma, W. (2011). Parking Availability Prediction for

Sensor-Enabled Car Parks in Smart Cities. *Proceedings of the 2011 International Conference on Ubiquitous Intelligence and Computing*, 266-273.

[3] Caicedo, C., Muratore, M., & Sánchez, F. (2019). Real-Time Prediction of Parking Space Availability in Urban Areas. *Journal of Transportation Engineering, Part A: Systems*, 145(4), 04019009.

[4] Channamallu, S., & Verma, R. (2020). A Comprehensive Study on Parking Occupancy Prediction. *Transportation Research Part C: Emerging Technologies*, 114, 109-130.

[5] Kotb, M., & Shalaby, A. (2019). Smart Parking Guidance and Reservation Systems: A Review. *Transportation Research Part C: Emerging Technologies*, 108, 154-179.

[6] Yang, X., Song, H., & Wang, L. (2020). Deep Learning for Real-Time Parking Occupancy Prediction in Smart Cities. *IEEE Transactions on Intelligent Transportation Systems*, 21(2), 705-714.

[7] Pamidimukkala, S., & Saha, A. (2020). Barriers to Electric Vehicle Adoption in Texas: A Case Study. *Transportation Research Part D: Transport and Environment*, 79, 102216.

[8] Huang, Z., & Li, Y. (2019). Location-Based Services: Research and Evolution. *International Journal of Geographical Information Science*, 33(3), 543-570.

[9] Sester, M. (2020). Mobility Data Analysis in Mobile Mapping Systems. *Journal of Navigation*, 73(1), 12-34.

[10] Channamallu, S., & Gupta, A. (2021). Comparative Analysis of Parking Occupancy Prediction Models. *IEEE Transactions on Intelligent Transportation Systems*, 22(3), 1376-1390.

[11] Liu, X., & Zhang, Y. (2020). Parking Occupancy Prediction Using Gradient Boosting Decision Trees. *Proceedings of the 2020 International Conference on Artificial Intelligence and Data Science*, 213-220.

[12] Sun, X., & Zhang, M. (2019). Short-Term Parking Occupancy Prediction Using Decision Trees and Random Forests. *Transportation Research Part C: Emerging Technologies*, 104, 45-58.

[13] Patel, R., & Rajput, K. (2020). Temporal and Spatial Patterns of Autonomous Vehicle Collisions. *IEEE Transactions on Vehicular Technology*, 69(7), 7650-7662.

[14] Hendricks, J., & Outwater, L. (2019). A Demand Forecasting Model for Park-and-Ride Lots. *Transportation Research Part A: Policy and Practice*, 125, 144-157.

[15] INRIX Research. (2020). The Economic Impact of Parking in the United States. *INRIX Report*.

[16] Caicedo, C., & Muratore, M. (2018). Parking Management and Modeling Car Park Patron Behavior in Underground Parking. *Transportation Research Part A: Policy and Practice*, 113, 78-90.

[17] Shoup, D. (2018). Cruising for Parking. *Transportation Research Part A: Policy and Practice*, 115, 1-18.

[18] Channamallu, S., & Verma, R. (2019). Smart Parking Systems: Recent Advances and Future Directions. *IEEE Transactions on Intelligent Transportation Systems*, 21(8), 3254-3265.

[19] Lin, L., & Wei, S. (2020). Survey on Smart Parking Solutions for Urban Mobility. *IEEE Access*, 8, 3212-3225.

[20] Virginia Metro Rail Authority. (2020). Feasibility Study of Real-Time Parking Information at Metro Rail Stations. *Virginia Department of Rail and Public Transportation*.

[21] Adepu, A., & Srinivas, S. (2020). Construction Cost Overruns during COVID-19: Challenges and Strategies. *Journal of Construction Engineering and Management*, 146(10), 04020095.

[22] Gordon, D. (2018). High-Tech Parking Meters in San Francisco. *The New York Times*.

[23] Richtel, M. (2019). The Role of Mobile Apps in Parking: Revolutionizing the Way We Park. *The New York Times*.

[24] SFMTA. (2019). Demand-Responsive Pricing for Parking in San Francisco. *San Francisco Municipal Transportation Agency*.