© Volume 2, Issue 10, October 2026 | JATIR
e-ISSN: 2839-5786, p-ISSN: 3139-8464

Smart Electric Vehicle Charging Slot Booking and Route Optimization Using Artificial Intelligence

[1]Shivani Shitole, [2]Umesh Jadhav, [3]Samarth Raykar, [4]Prasad Shelar, [5]Abhijit Padole
[1]*Faculty Guide Department of Information Technology P.G. Moze College of Engineering (PGMCOE), Wagholi, Pune, Maharashtra, India*
[2,3,4,5]*Student, Department of Information Technology P.G. Moze College of Engineering (PGMCOE), Wagholi, Pune, Maharashtra, India*

***Abstract*—The rapid growth of electric vehicles (EVs) has increased the need for charging services that are reliable, predictable, and easy to plan around, particularly during long distance travel. Existing charging applications mainly list nearby stations and rarely combine the battery level of the vehicle, the planned route, charger availability, traffic, and expected waiting time in a single decision. This paper presents the design of an intelligent web-based system for charging slot booking and route optimization that applies artificial intelligence (AI) and machine learning (ML). The user enters a destination and the current battery percentage. The system estimates the remaining driving range, identifies candidate charging stations along the route, predicts availability and waiting time from historical booking and demand data, and ranks the stations with a weighted multifactor score. The driver can then reserve a slot in advance and receive map-based navigation to the selected station. The architecture combines a React front end, a Flask application programming interface, a MySQL database, a scikit-learn recommendation engine, and an open map service. A worked scenario with illustrative values shows how the ranking balances distance, waiting time, availability, and traffic. The proposed approach is expected to reduce waiting at stations, lower range anxiety, and distribute demand more evenly across charging infrastructure.**

***Index Terms*—Electric vehicle, charging station recommendation, advance slot booking, route optimization, waiting-time prediction, machine learning.**

I. INTRODUCTION

The transition from fuel-based vehicles to electric vehicles (EVs) is reshaping road transport by offering a cleaner and more sustainable alternative. As the number of EVs grows, the demand for reliable and easily accessible charging infrastructure grows with it, and national initiatives such as

the e-AMRIT portal of NITI Aayog now publish a charging map to support adopters [1]. Charging, however, remains a planning problem for the driver. Unlike refueling a conventional vehicle, charging takes considerably longer, and a station that appears nearby on a map may have all of its chargers occupied when the vehicle arrives.

These difficulties are most visible during long distance journeys. A driver must decide where to stop, whether the remaining battery is sufficient to reach that stop, and how long the wait will be. Uncertainty about range is a documented psychological barrier to EV use [2], and studies of everyday charging behavior show that charging decisions depend strongly on how drivers perceive their remaining range [3]. Most existing charging applications display station locations and, in some cases, connector status, but they do not weigh the current battery percentage, the estimated driving range, the destination, the predicted waiting time, and traffic conditions together. They also seldom allow the driver to reserve a charger in advance. As a result, several vehicles may arrive at the same station during peak hours while other stations remain lightly used.

This paper presents the design of a smart charging slot booking and route optimization system that applies artificial intelligence (AI) and machine learning (ML). The system accepts a destination and the current battery percentage, estimates the remaining range, identifies suitable stations along the planned route, predicts availability and waiting time from historical booking and demand data, recommends a station, reserves a slot, and provides map-based navigation. The main contributions are as follows. First, a battery aware recommendation approach combines range, route, availability, waiting time, and traffic in one weighted score. Second, a waiting time prediction component is trained on historical booking and demand data. Third, an advance slot booking module spreads demand across stations. Fourth, a modular web architecture is built on widely available open technologies.

The remainder of the paper is organized as follows. Section II reviews related work. Section III states the problem and objectives. Section IV describes the system architecture, and Section V explains the proposed methodology. Section VI presents the implementation environment and database design. Section VII gives an illustrative scenario and the expected outcomes, Section VIII discusses advantages, applications, and limitations, and Section IX concludes the paper and outlines future scope.

II. RELATED WORK

Research on EV charging infrastructure has concentrated mainly on where stations should be built. Kchaou-Boujelben [4] reviewed the models and solution approaches for the charging station location problem, and Majhi et al. [5] presented a systematic review of the same problem. Ahmad et al. [6] examined optimal station placement together with its impact on the distribution network, covering objective functions, constraints, and optimization techniques. These studies serve planners and grid operators. They do not address the daily decision of an individual driver who must choose among stations that already exist.

A second line of work treats routing for electric vehicles. Schneider et al. [7] formulated the electric vehicle routing problem with time windows and recharging stations and solved it with a hybrid metaheuristic, which shows that recharging decisions need to be integrated into route planning rather than added afterward. Such formulations are designed mainly for fleet or logistics operators with known schedules, and they do not model uncertain waiting times or customer reservations. For the road network itself, classical shortest path algorithms [8], [9] and community-maintained map data [10] provide the practical basis for route computation in open applications.

A third line of work applies data driven methods to demand and waiting time. Queueing theory offers analytical baselines for waiting at facilities with several servers [11], and supervised learning methods such as random forests [12] and gradient boosting [13], [14] are widely used for tabular prediction tasks. Open-source libraries such as scikit-learn [15] make these methods readily available for student and industrial projects alike.

The works reviewed here therefore offer strong results on station placement, vehicle routing, and prediction, but they do not bring range estimation, availability prediction, advance booking, and navigation together in a single application aimed at the driver. Table 1 summarizes the related work and its relation to the proposed system.

Table 1 Summary of related work

Reference	Focus	Relevance to this work
[4]	Review of models and solution approaches for the charging station location problem	Background on infrastructure planning. Driver side decisions are not addressed.
[5]	Systematic review of the charging infrastructure location problem	Additional background on infrastructure location from a planner perspective.
[6]	Review of optimal station location and its effect on the distribution network	Highlights grid constraints that motivate spreading demand across stations.
[2], [3]	Range experience and charging behavior of EV users	Motivates battery aware planning and the use of behavior data.
[7]	Electric vehicle routing with time windows and recharging stations	Shows that recharging must be integrated into routing. Reservations are not modeled.
[14]	Scalable gradient boosted tree learning	Candidate model family for waiting time prediction.

III. PROBLEM STATEMENT AND OBJECTIVES

During long distance travel, EV users have difficulty locating suitable charging stations, judging whether chargers will be free, estimating the waiting time, and planning stops that match the battery

level and the destination. Existing applications mainly list stations and may not combine the battery percentage, remaining range, availability, expected waiting time, traffic conditions, and route requirements. Without advance booking, several vehicles can reach the same station in the same peak period, which creates queues and leaves other stations underused.

The aim of this work is to design an AI powered charging recommendation and slot booking system that helps drivers find suitable stations, predict availability and waiting time, reserve a slot in advance, and follow an optimized route that respects the battery level. The specific objectives are (i) to estimate the remaining driving range from the battery percentage; (ii) to identify and recommend suitable stations on the basis of battery level, destination, route, and availability; (iii) to optimize the route with map navigation and shortest path concepts; (iv) to predict waiting time from historical booking and demand data; (v) to provide advance slot booking with secure user authentication and booking management; (vi) to improve recommendations over time from booking patterns, traffic information, and user feedback; and (vii) to improve the utilization of charging infrastructure by distributing users across suitable stations.

IV. SYSTEM ARCHITECTURE

Layered Architecture

The architecture follows a layered web design, shown in Fig. 1. The driver interacts with a React web application that collects the destination and battery percentage and displays recommendations, bookings, and navigation. Requests pass to a Flask representational state transfer (REST) application programming interface (API) written in Python, which coordinates authentication, trip handling, and booking. This layer calls the AI and ML recommendation engine to compute range, predict waiting time, and rank candidate stations. The engine exchanges data with a map and route service, which may be Google Maps or OpenStreetMap, and with external data sources such as traffic, charger status, and historical booking records. A MySQL database stores users, vehicles, stations, slots, bookings, and trips. Slot booking and notification services confirm reservations and send alerts, and the selected station is passed to the navigation component.

Functional Modules

The system is organized into nine modules. The user authentication module handles registration, login, and secure access. The vehicle and battery module stores vehicle information and processes the battery percentage to estimate the remaining range. The destination and route module accepts the destination and identifies the route through the map service. The charging station module identifies stations relevant to the route and the travel requirements. The AI recommendation module analyzes battery level, destination, charger availability, estimated waiting time, and route efficiency to recommend a station. The waiting time prediction module uses historical booking and demand data. The slot booking module provides advance reservation and booking management. The route optimization and navigation module guides the driver to the selected station. Finally, the charging status module presents charging related status and booking information to the user.

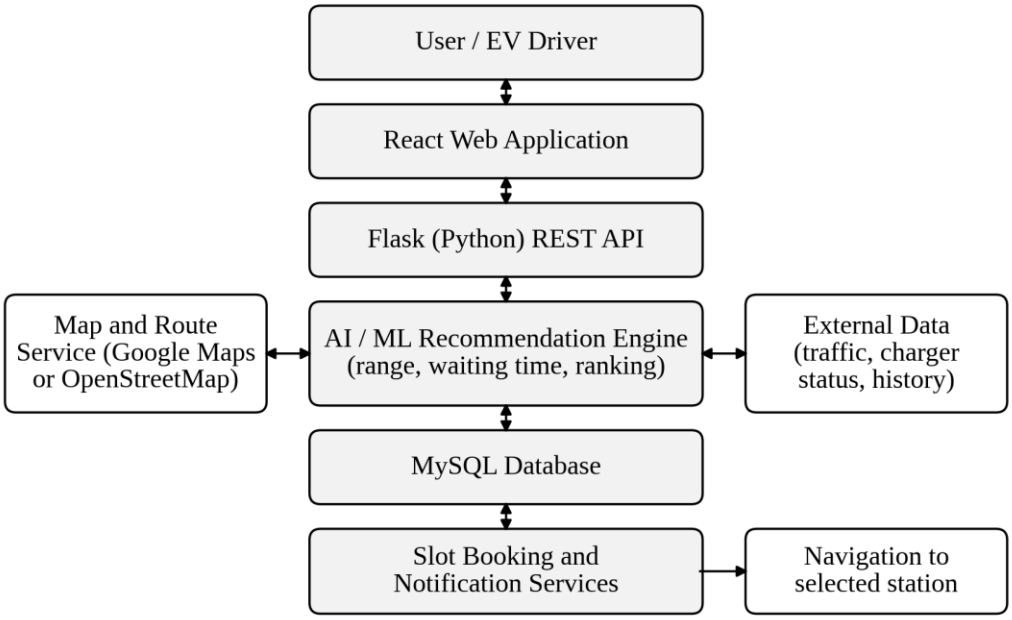


Fig. 1. Layered architecture of the proposed system.

V. PROPOSED METHODOLOGY

Overview of the Workflow

Figure 2 shows the end-to-end workflow. After login, the driver enters a destination and the current battery percentage. The system estimates the remaining range, finds candidate stations along the route, predicts availability and waiting time, recommends a station, reserves the chosen slot, and starts navigation. Booking outcomes and feedback are stored and later used to improve the predictions.

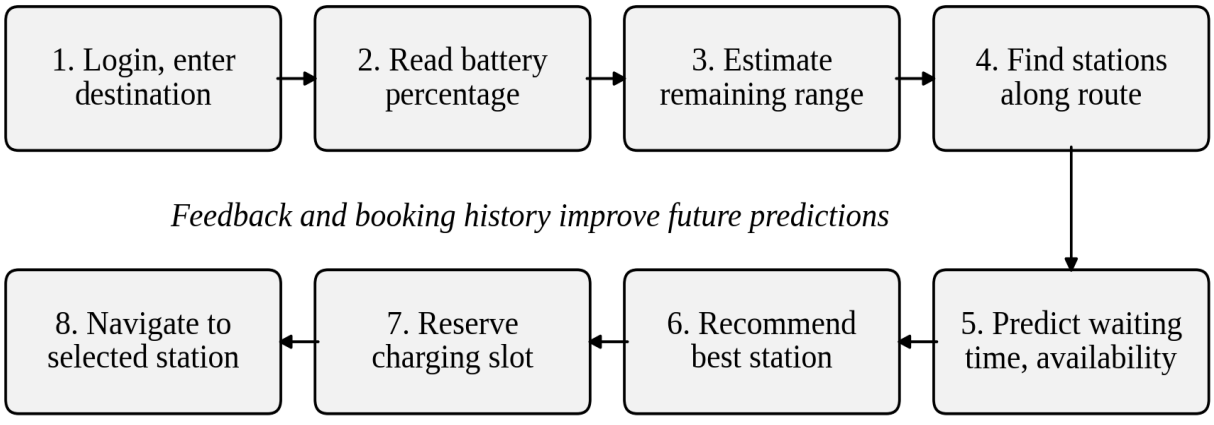


Fig. 2. Workflow of the proposed system.

Range Estimation

Let S be the current state of charge (SoC) of the battery in percent, E the usable battery capacity in kilowatt-hours, and ε the average energy consumption of the vehicle in kilowatt-hours per kilometer. The remaining driving range R in kilometers is estimated as

$$R = (S / 100) \times (E / \varepsilon) \quad (1)$$

To avoid arriving at a station with an almost empty battery, a reserve fraction ρ is held back, which gives the safe range

$$R_{safe} = (1 - \rho)\, R \quad (2)$$

In practice ε depends on speed, load, temperature, and traffic, so it is stored for each vehicle and can be updated from past trips. A station i located at route distance d_i from the current position is considered reachable only if $d_i \leq R_{safe}$. The SoC on arrival at that station is

$$S_{arr,i} = S - 100 \times (d_i\, \varepsilon / E) \quad (3)$$

Candidate Station Identification

The destination and the current position are sent to the map service, which returns the planned route as a sequence of road segments. Stations stored in the database are filtered to those located along the route or within a limited detour from it. When a hosted map service is used, route computation is delegated to it. When OpenStreetMap data [10] are processed locally, route and detour distances are computed with shortest path search, for which Dijkstra's algorithm [8] provides the exact baseline and A* search [9] provides a faster heuristic variant on large road graphs. Stations that fail the reachability condition in Equation 2 are discarded, and the remaining stations form the candidate set. If the destination lies beyond R_{safe}, the procedure is repeated from the selected station until the destination becomes reachable.

Availability and Waiting-Time Prediction

For each candidate station, the system estimates the arrival time from the route distance and the traffic adjusted speed. A simple baseline follows from queueing theory [11]. If q_i vehicles are expected to be ahead of the driver at arrival, each session lasts t_s on average, and the station has n_i chargers, the waiting time is approximately

$$W_i = q_i\, t_s / n_i \quad (4)$$

Here q_i is obtained from the booking table and the live status of the station. Because real demand varies with time of day, day of week, location, and traffic, the baseline is refined by a supervised regression model trained on historical booking and demand records. Candidate model families are random forests [12] and gradient boosted trees [13], [14], which handle mixed numerical and categorical features and need little feature scaling. The input features include the hour, weekday, station identifier, number of chargers, bookings already held for nearby time slots, recent occupancy, and traffic level. The target is the observed waiting time, and a classifier trained on the same features estimates the probability a_i that a charger is free on arrival. The models are implemented with scikit-learn [15] and are evaluated with the mean absolute error and the root

mean square error for waiting time and with standard classification metrics for availability. When a new station has little history, the queueing baseline is used until enough bookings accumulate.

Station Recommendation

The recommendation component ranks the candidate stations with a weighted score that balances the factors named in the problem statement. Four normalized terms are defined for station i. The distance term $u_i = d_i / R_{safe}$ lies between 0 and 1 for reachable stations. The waiting term $v_i = \min(W_i / W_{max}, 1)$ caps the predicted wait at an upper limit W_{max}. The traffic term $g_i = \min(T_i / T_{max}, 1)$ uses the expected traffic delay T_i in minutes on the way to the station and an upper limit T_{max}. The availability term is the predicted probability a_i. The composite score is

$$C_i = w_1(1 - u_i) + w_2(1 - v_i) + w_3 a_i + w_4(1 - g_i) \quad (5)$$

where the weights w_1 to w_4 are nonnegative and sum to one. A higher C_i is better, and the recommended station is the reachable station with the largest score. The weights can first be set from expert judgment, for example by giving waiting time and availability more influence than distance, and they can later be tuned from booking outcomes and user feedback. The top ranked stations are shown to the driver with their predicted wait, distance, and availability so that the final choice remains with the user.

Charging Time and Slot Reservation

Once a station is selected, the energy to be added to reach a target SoC S_{tgt} is

$$E_{req} = ((S_{tgt} - S_{arr,i}) / 100) \times E \quad (6)$$

and the charging duration is approximately

$$t_c = E_{req} / (\eta P) \quad (7)$$

where P is the rated power of the charger in kilowatts and η is the charging efficiency. The reservation covers the predicted arrival time and the charging duration. Each reservation is written to the booking table inside a database transaction that first checks that the slot is still marked available, so that two drivers cannot hold the same slot. A booking record links the user, station, slot, and trip, and the slot status changes from available to reserved. Confirmation and reminder notifications are sent to the user, and an unused reservation can be released after a grace period so that the charger is not left idle.

Navigation and Continuous Improvement

After booking, the map service provides turn by turn navigation to the selected station and, after charging, onward to the destination. Completed trips, actual waiting times, charger usage, traffic observations, and user ratings are written back to the database. These records are used to retrain the prediction models at regular intervals and to adjust the score weights, so that recommendations improve as the system accumulates data.

VI. IMPLEMENTATION ENVIRONMENT

Hardware and Software Requirements

The prototype is planned on commodity hardware and open-source tools, as listed in Table 2. Python supports the ML components and the Flask backend, React.js builds the user interface, MySQL stores the data, and Google Maps or OpenStreetMap supplies routing. Docker and TensorFlow are optional, with Docker intended for later deployment and TensorFlow for deeper models if they are needed.

Table 2 Hardware and software requirements

Category	Item	Purpose
Hardware	Intel Core i5 processor or higher	Development and testing
Hardware	8 GB memory minimum, 16 GB recommended	Model training and local servers
Hardware	256 GB solid-state drive or higher	Storage of code and data
Hardware	Internet connection, smartphone optional	Map services and mobile testing
Software	Windows 10 or 11, Linux, macOS	Operating system
Software	Python	Core programming and ML processing
Software	JavaScript, HTML5, CSS3, React.js	Web application development
Software	Flask or Django	Backend framework
Software	Scikit-learn, Pandas, NumPy, TensorFlow (optional)	Machine learning and data processing
Software	Google Maps API or OpenStreetMap API	Maps, routing, and navigation
Software	MySQL	Database management and storage
Software	Docker (optional), GitHub	Containerization and version control

Database Design

The relational schema contains six tables, summarized in Table 3. The Charging_Slots table holds the time and status of each slot, and the Bookings table references users, stations, and slots. This allows the booking history to serve also as the training data for the prediction models.

Table 3 Database tables and their main fields

Table	Main fields	Purpose
Users	user_id, name, email, password	Stores user information
Vehicles	vehicle_id, user_id, vehicle details, battery information	Stores vehicle and battery data
Charging_Stations	station_id, station name, location, availability	Stores station information
Charging_Slots	slot_id, station_id, slot time, status	Stores available and reserved slots
Bookings	booking_id, user_id, station_id, slot_id, booking details	Stores charging reservations
Trips	trip_id, user_id, source, destination, route details	Stores travel and route information

VII. ILLUSTRATIVE SCENARIO AND EXPECTED OUTCOMES

Because the system is presented at the design stage, this section uses a worked scenario with assumed values, not measured results, to show how the equations behave. A vehicle has a usable battery capacity $E = 40$ kWh and an average consumption $\varepsilon = 0.15$ kWh/km. The current charge is $S = 40\%$, the reserve fraction is $\rho = 0.10$, and the destination is 180 km away. Equation 1 gives $R = 106.7$ km, and Equation 2 gives a safe range of 96.0 km, so the destination cannot be reached without a charging stop.

Three candidate stations, labeled A, B, and C, lie along the route. Their assumed distances, predicted waiting times, availability probabilities, and traffic delays are given in Table 4. With $W_{max} = 60$ min, $T_{max} = 30$ min, and weights $w_1 = 0.30$, $w_2 = 0.30$, $w_3 = 0.25$, and $w_4 = 0.15$, Equation 5 produces the scores in the last column.

Table 4 Assumed values and scores for the illustrative scenario

Station	Distance (km)	Predicted wait (min)	Availability probability	Traffic delay (min)	Arrival charge (%)	Score
Station A	45	12	0.60	10	23.1	0.649
Station B	70	4	0.85	5	13.8	0.699
Station C	90	25	0.40	15	6.3	0.369

Station A is the nearest, but its predicted wait is three times that of Station B and its availability is lower, so Station B obtains the highest score although it is 25 km farther. Station C is within the

safe range but the vehicle would arrive with the lowest charge and face the longest predicted wait, so it ranks last. For the selected Station B, Equation 3 gives an arrival charge of 13.8%. Charging to S_{tgt} = 80% requires 26.5 kWh by Equation 6. With a 50 kW charger and $\eta = 0.90$, Equation 7 gives a charging time of about 35 min. The remaining 110 km needs about 16.5 kWh, so the vehicle reaches the destination with roughly 38.8% charge in this scenario.

The expected outcomes of the implemented system are that drivers can select a station that matches their battery and route, check availability and predicted waiting time before leaving, reserve a slot in advance, and follow optimized navigation to the station. These features should reduce waiting and unnecessary detours, lower range anxiety, and spread demand more evenly across stations. Quantitative validation is planned as future work with the prototype. The proposed metrics are the mean waiting time per charging session, the proportion of arrivals that find no free charger, the waiting time prediction error, the detour distance, and the balance of utilization across stations. Each can be compared with a baseline that simply sends the driver to the nearest station.

VIII. DISCUSSION

Advantages and Applications

The main advantages of the proposed system are reduced waiting through advance reservation, battery aware recommendations, route optimization with integrated navigation, and the joint consideration of charger availability and predicted waiting time. Together these improve the charging planning experience of the driver. The concept applies to long distance EV travel, where it helps plan charging stops, and to urban charging, where it supports finding and reserving slots at suitable stations. It can support charging station management through organized use of chargers and slots, it can be extended to EV fleet operations that manage the changing needs of many vehicles, and it can form part of wider smart mobility services.

Limitations

The quality of the recommendations depends on the availability and accuracy of station, traffic, booking, and battery related data. Map and navigation features depend on the selected map service. Waiting time prediction requires sufficient historical booking and demand data, which a new deployment will not have at the start, and real time charger availability depends on integration with charging station data sources.

IX. CONCLUSION AND FUTURE SCOPE

This paper presented the design of a smart EV charging slot booking and route optimization system that estimates range from the battery level, identifies reachable stations along the route, predicts availability and waiting time, ranks the stations with a weighted score, and allows slots to be reserved in advance before map-based navigation. By recommending stations intelligently, booking slots ahead of arrival, and optimizing the route, the system is intended to help EV users travel efficiently, reduce waiting, improve convenience, and support the growing EV ecosystem. The

worked scenario showed how the ranking trades distance against waiting time, availability, and traffic.

Future work includes building and testing the prototype against the evaluation metrics above. Further extensions are integration with real time charging station APIs for live charger status, additional traffic and navigation data sources, advanced ML models for waiting time and demand prediction, personalized recommendations from travel and charging history, mobile applications for Android and iOS, integration with vehicle telemetry for automatic battery status updates, and dynamic pricing with charging cost comparison.

ACKNOWLEDGMENT

The authors thank the faculty of the Department of Information Technology, PGMCOE, Wagholi, for their guidance and support during this project.

REFERENCES

[1]. NITI Aayog, "e-AMRIT electric vehicle portal: Charging map," Government of India. [Online]. Available: https://e-amrit.niti.gov.in/charging-map

[2]. T. Franke, I. Neumann, F. Bühler, P. Cocron, and J. F. Krems, "Experiencing range in an electric vehicle: Understanding psychological barriers," Appl. Psychol., vol. 61, no. 3, pp. 368–391, 2012.

[3]. T. Franke and J. F. Krems, "Understanding charging behaviour of electric vehicle users," Transp. Res. F, Traffic Psychol. Behav., vol. 21, pp. 75–89, 2013.

[4]. M. Kchaou-Boujelben, "Charging station location problem: A comprehensive review on models and solution approaches," Transp. Res. C, Emerg. Technol., vol. 132, Art. no. 103376, 2021.

[5]. R. C. Majhi, P. Ranjitkar, M. Sheng, G. A. Covic, and D. J. Wilson, "A systematic review of charging infrastructure location problem for electric vehicles," Transp. Rev., 2021, doi: 10.1080/01441647.2020.1854365.

[6]. F. Ahmad, A. Iqbal, I. Ashraf, M. Marzband, and I. Khan, "Optimal location of electric vehicle charging station and its impact on distribution network: A review," Energy Rep., vol. 8, pp. 2314–2333, 2022.

[7]. M. Schneider, A. Stenger, and D. Goeke, "The electric vehicle-routing problem with time windows and recharging stations," Transp. Sci., vol. 48, no. 4, pp. 500–520, 2014.

[8]. E. W. Dijkstra, "A note on two problems in connexion with graphs," Numer. Math., vol. 1, no. 1, pp. 269–271, 1959.

[9]. P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," IEEE Trans. Syst. Sci. Cybern., vol. 4, no. 2, pp. 100–107, Jul. 1968.

[10]. M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," IEEE Pervasive Comput., vol. 7, no. 4, pp. 12–18, Oct. 2008.

[11]. L. Kleinrock, Queueing Systems, Volume 1: Theory. New York, NY, USA: Wiley, 1975.

[12]. L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[13]. J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” Ann. Statist., vol. 29, no. 5, pp. 1189–1232, 2001.

[14]. T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, San Francisco, CA, USA, 2016, pp. 785–794.

[15]. F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.