 doi:10.20944/preprints202203.0094.v1

Article

Smart Scheduling of Electric Vehicles Based on Reinforcement Learning

Andrei Viziteu [1], Daniel Furtună [1], Andrei Robu [1], Stelian Senocico [1], Petru Cioată [1], Remus Marian Baltariu [1], Maria Simona Răboacă [2,3,*] and Constantin Filote [2].

[1] Research and Development Department, ASSIST Software, Str. Tipografiei Nr.1, 720043 Suceava, Romania; andrei.viziteu@assist.ro, daniel.furtuna@assist.ro, andrei.robu@assist.ro, stelian.senocico@assist.ro, petru.cioata@assist.ro, remus.baltariu@assist.ro
[2] Faculty of Electrical Engineering and Computer Science, Stefan cel Mare University of Suceava, Str. Universitatii Nr.13, 720229 Suceava, Romania; filote.constantin@gmail.com, simona.raboaca@icsi.ro
[3] National Research and Development Institute for Cryogenic and Isotopic Technologies—ICSI Rm. Valcea, 240050 Ramnicu Valcea, Romania; simona.raboaca@icsi.ro

Abstract: As the policies and regulations currently in place concentrate on environmental protection and greenhouse gas reduction, we are steadily witnessing a shift in the transportation industry towards electromobility. There are, though, several issues that need to be addressed to encourage the adoption of EVs at a larger scale. To this end, we propose a solution capable of addressing multiple EV charging scheduling issues, such as congestion management, scheduling a charging station in advance, and allowing EV drivers to plan optimized long trips using their EVs. The smart charging scheduling system we propose considers a variety of factors such as battery charge level, trip distance, nearby charging stations, other appointments, and average speed. Given the scarcity of data sets required to train the Reinforcement Learning algorithms, the novelty of the recommended solution lies in the scenario simulator, which generates the labelled datasets needed to train the algorithm. Based on the generated scenarios, we created and trained a neural network that uses a history of previous situations to identify the optimal charging station and time interval for recharging. The results are promising and for future work we are planning to train the DQN model using real-world data.

Keywords: Smart scheduling, Smart Reservations, Reinforcement Learning, Electric vehicle charging, Electric Vehicle Charging Management platform, DQN Reinforcement Learning algorithm

1. Introduction

The last few decades have been marked by rapid technological advances that have resulted in significant positive changes in daily life, but as well in an increase of pollution levels. To address the issue of pollution associated with the transportation industry, extensive technological research and development has been carried out in order to pave the way for electromobility worldwide.

As pointed out in Electric Vehicles for Smarter Cities: The Future of Energy and Mobility issued by the World Economic Forum in January 2018 [1], to prevent congestion and pollution in the urban areas that have already been and will continue to be reshaped by demographic shifts, it is required to implement radical sustainable and secure mobility and energy solutions. The authors underline the fact the charging infrastructure deployment should be based on an anticipation of the long-term mobility transformation. Moreover, they consider that ensuring a reduction of range anxiety and developing smart charging technologies are key elements in the EV market approach, as they would contribute to the adoption of electro-mobility.

Wang et al [2] emphasize the need of a scheduling approach to bridge the gap between EV charging needs and charging station supplies, as well as to deliver a favorable user experience that will stimulate EV adoption. Franke et all [3] have analyzed the psychological barriers in adopting electric mobility, interfering with purchase intentions, focusing on EV range.

 doi:10.20944/preprints202203.0094.v1

In this context, we propose a solution that comes to address precisely the aspects mentioned above, reducing range anxiety among EV owners by ensuring a smart scheduling of electric vehicles charging using reinforcement learning. To this end, we have structured the present paper into 9 chapters. Following the introduction, the second chapter constitutes a brief overview of the current state of the art concerning the use of Reinforcement Learning techniques in the EV charging area. After having presented the work methodology, we went on to provide a detailed description of data collection processes and simulations, provided by the scenario generator, in section 5. We have dived deeper into the issue in chapter 6, where we illustrated the algorithm workflow and then presented the case study of Smart EVC, explaining the charging station recommendation system workflow. Subsequently, we analysed the results of the experiments performed based on the scenarios generated by the simulator.

2. *State of the art*

To address congestion management, Rigas et all [8] analyze the issue from both the perspective of the EV users and of the charging points, suggesting that congestion at the charging stations could be avoided by directing EVs towards various charging points and by distributing charging points' location along the routes. The authors studied artificial intelligence techniques for establishing energy-efficient routing, as well as for selecting charge points, analysing, at the same time, the possibility to integrate EVs into the smart grid.

Tuchnitz et al. [17] propose a flexible and scalable charging strategy for EVs, using a Reinforcement Learning algorithm to establish a smart system for the coordination of an EV fleet charging process. Unlike optimization-based strategies, the proposed system does not require variables such as arrival and departure time and electricity consumption in advance.

Ruzmetov et al. [18] identifies the lack of certainty of EV drivers to have an available charging point once they reach the charging station on their route as one of the major drawbacks in adopting electromobility. The authors introduce a platform meant to ensure a constant cooperation between the different involved entities: the energy suppliers, the charging stations and the EVs and EV users, proposing an optimization of the EVs' scheduling and allocation to the charging stations. The destination set by the drivers, as well as the battery level are taken into account when proposing a charging station ensuring that this does not divert them from the route.

Qin and Zhang [19] conducted a theoretical study that enabled them to create a distributed scheduling protocol meant to reduce the waiting time consisting of both the queuing time and the actual charging time, during a trip, along a highway. The reservations made by EV drivers for their next charging and the reservation adjustments are based on the minimum waiting time communicated by the charging stations, which periodically update this information, so as to enable drivers to make the optimal selection, in terms of waiting time.

Weerdt et al. [20], too, propose a solution to address the issue of congestions at the charging stations and long queuing times, in the form of an Intention Aware Routing System (IARS), which enables vehicles to reduce their travel time by taking into account the intentions of other vehicles. A central system is fed with probabilistic information about the intentions of vehicles in terms of estimated time of arrival at the charging station and therefore, it can predict overcrowding and associated waiting times.

Furthermore, to address the needs of EV owners, while at the same time avoiding charging station congestions and power grid overload, Lui et al. [21] have conducted extensive research and proposed a reinforcement learning-based method for scheduling EV charging operations. To this end, the authors created a framework to enable communication between the charging stations and the EVs, and then determined a dynamic travel time model, as well as the EV charging navigation model. To further optimize the charging stations located in the area, they used reinforcement learning to enhance the charging scheduling model.

Shahriar et al [22] provide a comprehensive overview of the application of various machine learning technologies to analyse and predict EV charging behaviour, identifying the lack of publicly available datasets necessary for the training of ML models as one of the major drawbacks in the field.

The authors also underline that available data is rather irrelevant, as it is specific to certain geographical areas, with traffic and EV users' behaviour particularities that cannot be applied in other locations.

Mnih et al [23] explore recent breakthroughs in deep neural network training to create a deep Q-network that makes use of end-to-end RL to "learn successful policies directly from high-dimensional inputs". The authors claim that using DQN, they managed to create a first AI agent capable of achieving mastery in a wide variety of difficult tasks. This research, along with its predecessor Playing Atari with Deep Reinforcement Learning [24] could be considered the cornerstone of DQN, demonstrating that in complex RL contexts, a convolutional neural network trained with the Q-learning algorithm is able learn control policies from raw video data.

Among the techniques that enable DQN to overcome unstable learning, Experience Replay has a significant impact, as it stores experiences such as state transitions, actions, and rewards, which constitute required data for Q-learning, and it creates mini-batches for updating the neural networks. Learning speed is increased and the reuse of past transition prevents cataclysmic forgetting. [25]

In their paper entitled Deep Reinforcement Learning Based Optimal Route and Charging Station Selection, Lee et al. [26] propose an algorithm designed to return the best possible route and charging stations, reducing the overall travel time, taking into account the dynamic character of traffic conditions, as well as unpredictable future charging requests. To determine the best policy for the selection of an EV charging station a well-trained Deep Q Network (DQN) has been used. The authors point out that due to dimensionality, it is difficult to use Q-learning with the lookup table method for large-scale problems in real-world scenarios. They used DQN to approximate the optimal action-value function. To contribute to the fuel efficiency of plug-in hybrid EVs, Chen et all [27] propose a stochastic model predictive control strategy for energy management, based on Reinforcement Learning. Furthermore, the authors employ the Q-learning algorithm to set a RL controller used in the optimization process.

In this context, the purpose of this paper is to propose a solution capable of addressing multiple issues related to EV charging scheduling, such as congestion management, scheduling a charging station in advance, and enabling EV drivers to plan optimized long trips using EVs.

The novelty consists of the smart charging scheduling system that takes into account multiple parameters such as battery charge level, trip distance, available charging stations nearby, other appointments and the average speed. Furthermore, given the lack of data sets necessary to train the Machine Learning / Reinforcement Learning algorithms, the novelty of the proposed solution also resides in the scenario simulator that generates labelled datasets required to train the algorithm.

3. *Methodology*

The present article proposes to illustrate the technical details for the implementation of the DQN Reinforcement Learning algorithm (deep Q-networks) for the smart reservation of a charging point for an electric vehicle, as it has been developed within the framework of the Smart EVC project.

4. As illustrated in the diagram below, the methodology employed while conducting the current study has as a starting point the parameters generated by the scenario simulator we had to create to compensate for the lack of relevant publicly available datasets. The datasets are required to train the neural network of the DQN model. The simulator feeds the algorithm with the data associated to a specific situation, based on which the algorithm makes a decision. The scenario generator runs the simulation according to this decision and subsequently, the algorithm is assigned a reward associated with the scenario, informing the algorithm of the correctness of its decision. Given the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 March 2022 doi:10.20944/preprints202203.0094.v1

rewards it has received in each situation, the network is then trained to make better decisions in the future.

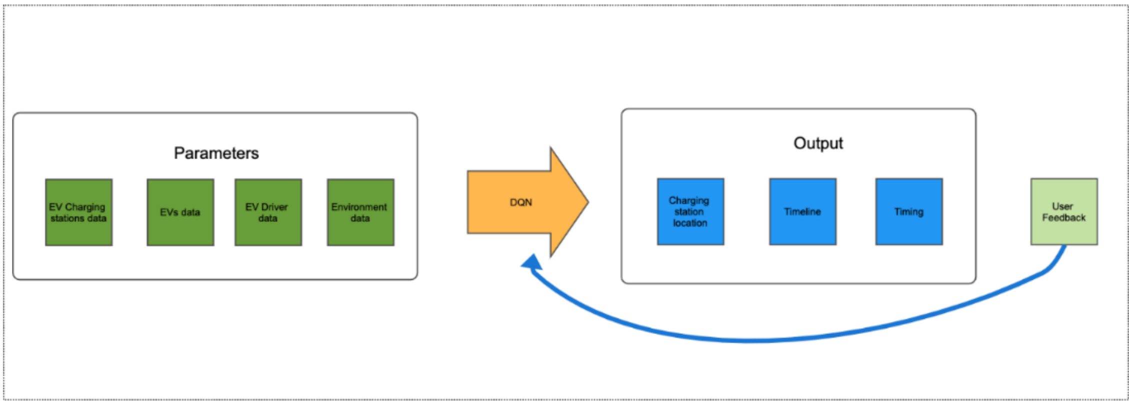

Figure 1 – Study Methodology

The smart scheduling of charging points for electric vehicles represents an issue that involves random variables. Given that it is difficult to create a dataset with the optimal solution for every potential situation, it cannot be addressed through Supervised Learning. For this reason, we have focused on a different class of Machine Learning algorithms, namely Reinforcement Learning.

The main goal of Reinforcement Learning algorithms is to identify the optimal solution to a problem, to maximize the reward function, in the long term. The RL agents learn by continuously interacting with the environment and observing the rewards in all circumstances. As we deal with an infinite number of possible situations, it is not possible to create a table to store the best solution associated with every situation. This challenge led us to the DQN algorithm, an RL algorithm using a function to decide the optimal solution in any situation. The DQN algorithm uses a neural network to decide upon the optimal solution.

5. *Data Collection and Simulations*

The DQN algorithm was written in Python based on the article entitled Human-level Control Trough Deep Reinforcement Learning, published in 2015 [23].

The neural network was written using the Keras module from the TensorFlow library. To begin with, we opted for a simple neural architecture, which can be extended later, following the experiments. Because training this type of network requires extensive time resources, the neural network model is automatically saved and then loaded during the next iteration.

The Neural Network architecture of the DQN model consists of the following layers: a 7-neurons input layer, a hidden layer with 16 neurons and the output layer with one neuron. The model is compiled using the Adam optimizer and the Huber loss. The neural network's weights are saved after each 100 iterations and the last checkpoint is loaded when we restart the app.

For training purposes, we needed a history of the situations encountered thus far, as well as rewards associated with each decision made by the algorithm. To keep track of all scenarios, we created the ReplayMemory class that allows us to store new scenarios.

The Replay Memory class stores two lists: the first one stores the scenarios without the reward and the second one saves the scenarios with their associated reward, assigned after the simulation of that scenario. When it receives the reward for a specific scenario, it is then moved from the first list to the second one, to avoid storing the same data in two locations. The training is performed only on scenarios from the second list, as we can't train the DQN model on data without results.

The Deepmind paper [23] suggests a maximum replay length of 1000000. We reached the conclusion that this causes memory issues, and therefore we used a maximum length of 10000. When the Replay Memory exceeds the 10000 scenarios limit, the oldest items in the list are removed.

When the DQN algorithm makes a decision, the simulator's generated situations are stored. The simulator then runs based on the algorithm's previous decisions and assigns a reward. This reward

 doi:10.20944/preprints202203.0094.v1

is automatically associated with the respective scenario, yielding a data set that can be used to train the neural network.

The DQN model has 2 types of decision-making processes:

- Exploration - In this phase the decision-making process is mostly random, so that the model can explore more and discover what results are yielded by different decisions. This enables the agent to improve its knowledge about each action, which is usually beneficial in the long-term. In the beginning, the agent uses this random decision-making process and slowly it starts to gradually exploit its knowledge. By the time it reaches the 10000th iteration, this exploration process probability reaches its lowest point, where it stays for the rest of the training time.

- Exploitation - In this phase the agent is exploiting its knowledge and opts for the greedy approach to get the most reward.

The neural network of the DQN algorithm is trained after every n decisions, n representing a hyperparameter of the DQN class. The consulted paper recommends starting a training session after every four algorithm decisions, which is the value used in the current implementation. The training of the neural network is performed based on 32 situations randomly selected from the replay memory.

Below, we have provided the steps involved in training the DQN model's neural network:

1. Create a list of 32 random indices from 0 to the length of the Replay Memory.
2. Get a list of samples from the Replay Memory based on the indices list created at step 1.
3. Split the sample in state and reward lists.
4. Train the neural network for 1 epoch using the state and reward lists as input.

To enable communication with the scenario generator, a Python server has been implemented, using the FastApi library. The Endpoint recommending the charging station receives as body parameters information related to the current situation and returns the station's ID and the time slot of the recommended reservation.

The "recommend_charging_station" endpoint is the main piece that connects the scenario generator with the DQN model. This endpoint expects as input the total distance of the trip, the battery capacity, current level of the battery and a list of charging stations, each charging station with its own id, charging power and reservations. As for its output, it returns the id of the recommended charging station and the start and end points of the interval recommended for reservation.

When a request is made, the received data is processed so that it can be fed to the neural network. The processed data is then used to make a prediction for the charging station best suited to the current scenario. Before returning the data, it is processed again to create the interval of time recommended.

The scenario generator simulator is written in JavaScript using the PixiJS library. An essential component of this simulator is the vehicle's controller, this class being the main element that generates new scenarios and assesses the rewards assigned following a decision.

The Car controller class has multiple properties such as the car's speed, the current battery level, the battery capacity, and the consumption per KM. It also has a method that decides its next destination and it controls the direction it should take at each frame.

There are two other important classes employed by the scenario simulator. The first one defines the id, position and charging power of a charging station and the second one keeps track of all reservations with their charging station id and interval.

For the purposes of the current study, we have created an administration interface that enables us to create charging stations randomly positioned, as well as vehicles that start their journey from a random place and have a random destination. To this end, the simulator has created three interconnected cities, to allow for the simulation of the scenarios. The simulator communicates with the Python server through the Axios library.

The simulator's interface is presented in Figure 1, below:

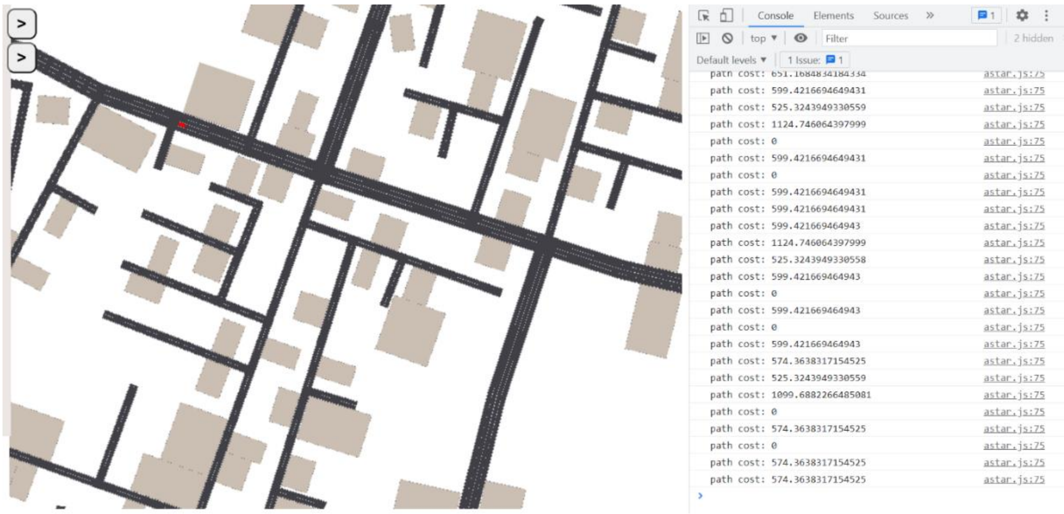

Figure 2 – Simulator Interface

6. *Reinforcement Learning Model*

We have provided below a diagram summarizing the DQN algorithm workflow, along the scenario simulator.

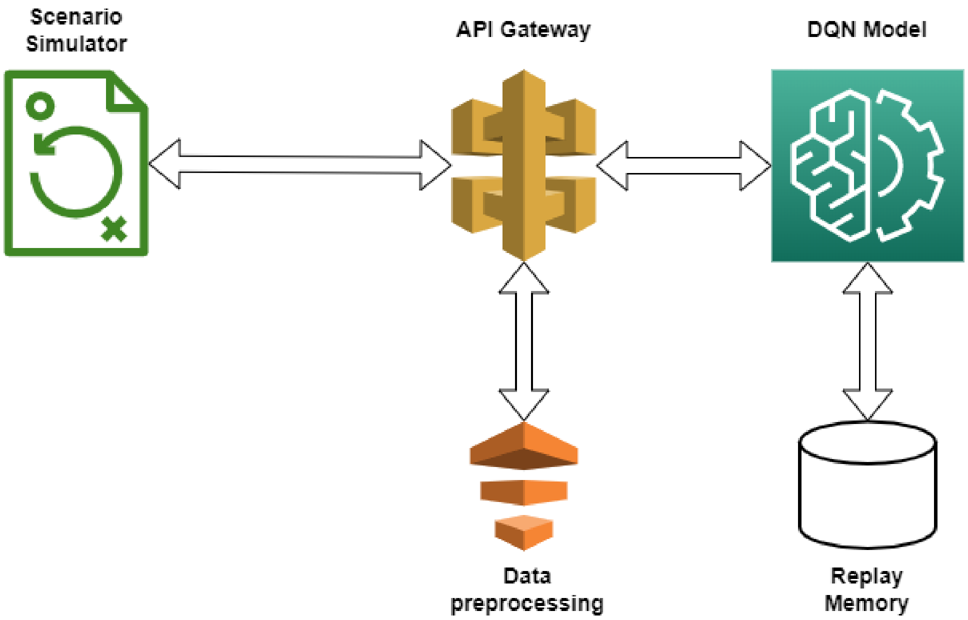

Figure 3 – DQN Algorithm Workflow

1. A new situation is generated by the simulator.
2. The simulator calls the charging station recommendation endpoint, feeding the information associated with the generated scenario.
3. The FastApi server calls an input data processing method, to map the data in the format required by the DQN algorithm.
4. The FastApi server calls the training method of the DQN algorithm with the processed data.
5. The DQN algorithm makes a decision according to the current learning level or, with a certain probability, makes a random decision.
6. The DQN algorithm stores the current scenario in ReplayMemory.
7. If there are enough entries in the ReplayMemory, a training dataset is generated, which is then used to train the neural network of the DQN algorithm.
8. The DQN algorithm returns the decision to the FastApi server.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 March 2022 doi:10.20944/preprints202203.0094.v1

9. The FastApi server maps the data in a format required by the simulator and sends it as a response to the request initiated by the scenario generator.
10. The scenario with the charging point reservation is simulated using data from the FastApi server.
11. The value of the reward is established based on the simulation (a number between 0 and 100).
12. The reward is sent to the FastApi server, along with the scenario's ID.
13. Through the FastApi server, the reward reaches the ReplayMemory and the scenario in question is updated.

7. *Case Study*

The research activities described in the present paper have been carried out within the framework of the Smart EVC project, which aimed to create an intelligent charging station management platform based on Blockchain and Artificial Intelligence allowing for user – charging station interactions. Among other things, the mobile app is designed to enable users to plan a trip, a feature that is especially useful for longer routes.

The "Plan a Trip" feature has the role of scheduling the reservation of charging stations along a route. The user can reserve the charging station recommended by the ML model or can choose other ones along the way.

When the user creates a new trip, the app prompts them to enter some relevant information (Fig. 4.b), which is critical for the DQN model's accuracy and is related to the battery status and the vehicle type. This should be enough to assess the maximum battery capacity and the power consumption.

The history of the user's trips (Fig. 4.a) provides data on previous trips that can be used as a training set to further refine the current model or training a new model from scratch.

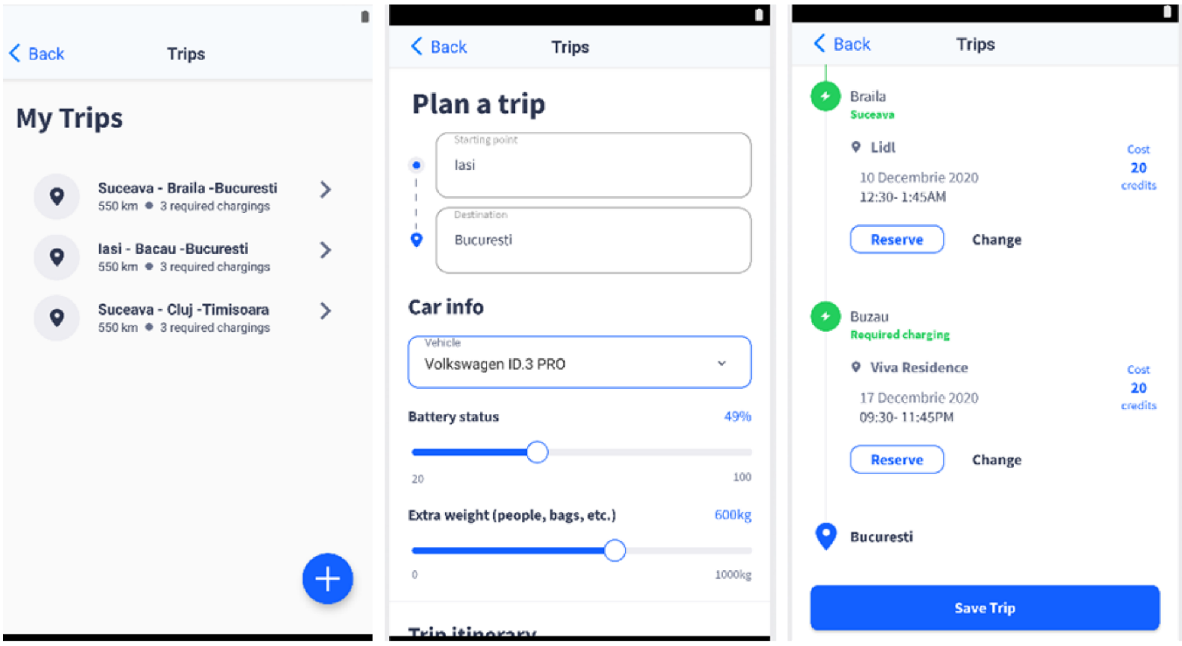

Figure 4- Smart EVC – Plan a trip section

As illustrated in the architecture in Figure 5, the charging stations selection system works as follows:

- The user sets up the starting and destination points for the trip in the mobile app, along with information about the vehicle and the battery status.
- The mobile app will call a Back-End endpoint with the data the user entered in the app.

doi:10.20944/preprints202203.0094.v1

- The Back-End will get from database all the charging stations relevant for the data provided by the user and call the ML recommendation endpoint with the data received from database.
- The DQN model will predict the charging stations and the specific reservation time best suited for the data it gets from Back-End and sends them back as a response.
- The Back-End will send the response to the mobile app with the charging stations recommended by the DQN model.
- The mobile app will display the recommended charging stations to the user, with the option to create a reservation for the specified charging stations (Fig. 4.c)

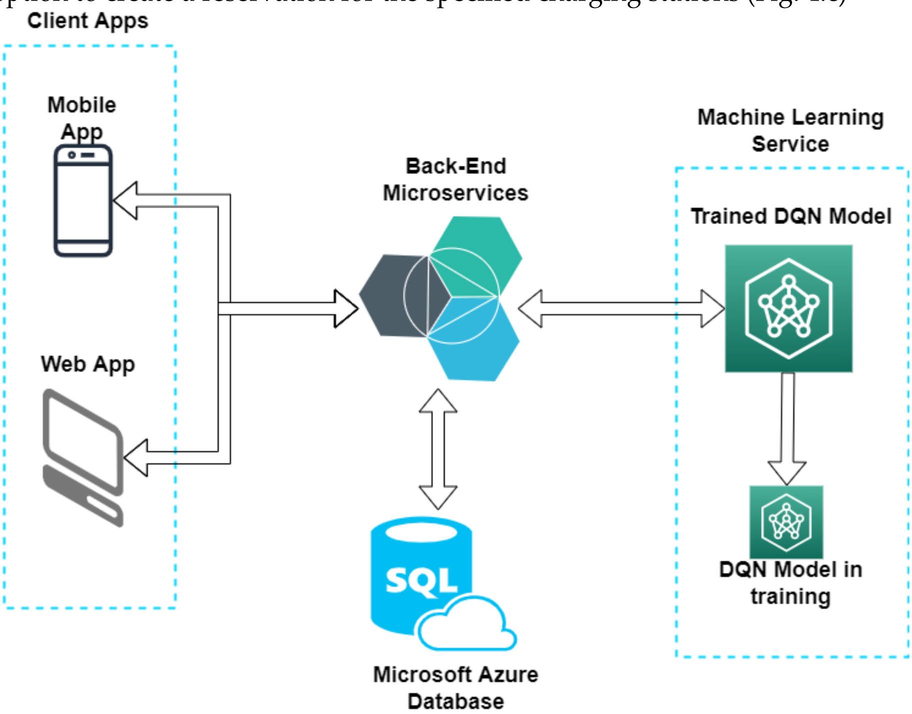

Figure 5 – Architecture of the trained module

8. *Results and discussions*

So far, we have performed several experiments based on the scenarios generated by the simulator.

In these experiments we used a DQN with a simple neural network (a single hidden layer of 16 neurons). For this model we used the Adam optimizer with a learning rate of 1e-3 and the Huber loss for a more stable training. The first 10000 scenarios were epsilon-greedy, meaning that the model took more random decisions to explore the environment. After the first 10000 scenarios, the probability of taking a random decision is set at 0.1, so most of the time we exploit what the model learned so far.

Every scenario is saved in the Replay Memory and after the simulation is run for that specific scenario, the Replay Memory is updated with a specific reward. The scenarios saved in the Replay Memory are then used to further train the DQN model. The training is performed after every 4 scenarios with minibatches of size 32. The scenarios used for training are drawn at random from the Replay Memory.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 March 2022 doi:10.20944/preprints202203.0094.v1

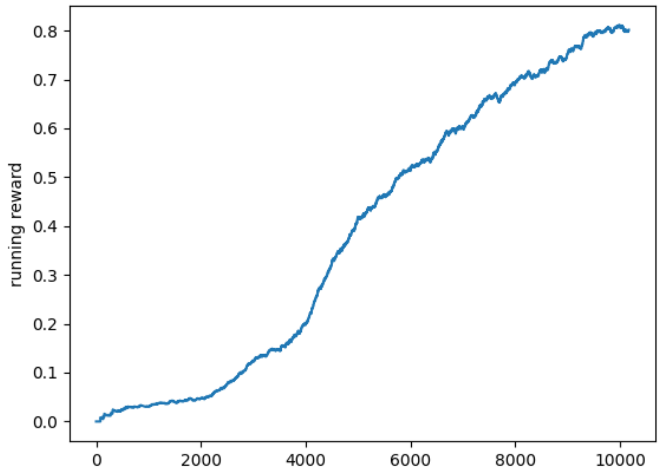

Figure 6 - Plot of the mean running reward

We chose as the evaluation metric the mean running reward over the last 1000 scenarios. Fig. 6 shows a plot of the mean running reward in the exploration period. We can see that the model's accuracy registers a steady growth in this period.

As a testing set, we used 2000 scenarios generated by the scenario simulator, where the DQN model only predicted the action to take, based on the previous training. In over 80% of the situations, the model predicted the correct charging station and the appropriate time slot. The plot below shows the mean running reward for the testing set.

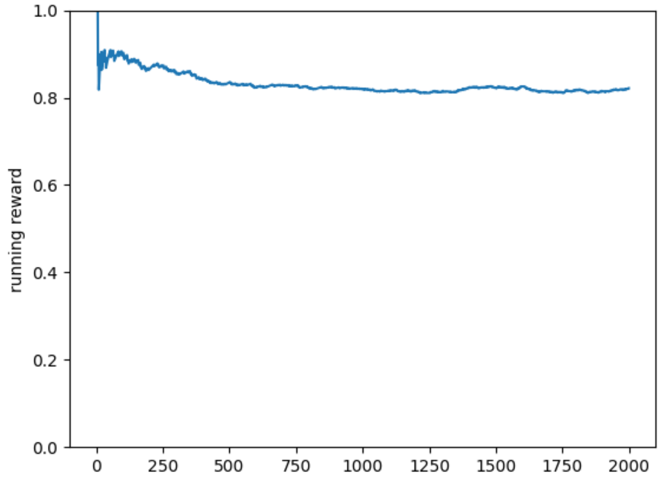

Figure 7 - Plot of the mean reward on the testing set

9. *Conclusions and limitations*

We have developed and trained a neural network that uses a history of the situations encountered thus far to identify the optimal charging station and time interval for recharging. Rewards are assigned to each decision made by the algorithm. In lack of available datasets, a simulator that generates training data has been implemented, creating new scenarios.

Because the training data is generated by a simulator, it rarely resembles real-world data. If the DQN algorithm encounters situations that differ significantly from the scenarios in the training data sets, it may make incorrect decisions. The considerable training time of the neural network constitutes yet another constraint, as it needs to go through multiple situations and iterations to reach an optimal solution. Furthermore, the training time is extended because every scenario must be simulated before the algorithm is assigned a reward.

10. Future research directions

Going further, we are planning to train the DQN model using real-world data. This data can be retrieved from Google Maps through its Directions API. We can set up some fake charging stations to use as reference, but these charging stations will use some real-world restrictions, such as being located within the boundaries of a city and (arguably) dispersed across the entire map.

With this setup we can simulate scenarios with real-world distances, traffic, and car parameters. The model will then be trained using these scenarios and further refined when we start testing on real vehicles and charging stations. Training the DQN model this way will save us from having to train it from scratch with real data. We consider this a big win as training an algorithm from scratch implies a multitude of time-consuming scenarios.

Acknowledgements:

This work was supported by a grant of the Romanian Ministry of Education and Research, CCCDI - UEFISCDI, project number PN-III-P2-2.1-PTE-2019-0642, within PNCDI III.

References

1. World Economic Forum, "Electric Vehicles for Smarter Cities: TheFuture of Energy and Mobility,"World Econ. Forum, no. January, p.32, 2018. https://www3.weforum.org/docs/WEF_2018_%20Electric_For_Smarter_Cities.pdf
2. Wang, R.; Chen, Z.; Xing, Q.; Zhang, Z.; Zhang, T. A Modified Rainbow-Based Deep Reinforcement Learning Method for Optimal Scheduling of Charging Station. Sustainability 2022, 14, 1884. https://doi.org/10.3390/su14031884, https://www.mdpi.com/2071-1050/14/3/1884/pdf
3. Franke, Thomas & Kreißig, Isabel & Schmalfuß, Franziska & Cocron, Peter & Krems, Josef. (2012). Experiencing Range in an Electric Vehicle: Understanding Psychological Barriers. Applied Psychology. 61. 368-391. 10.1111/j.1464-0597.2011.00474.x. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1071.6268&rep=rep1&type=pdf
4. Ajay Subramanian, Sharad Chitlangia, Veeky Baths, Reinforcement learning and its connections with neuroscience and psychology, Neural Networks, Volume 145, 2022, Pages 271-287, ISSN 0893-6080, https://doi.org/10.1016/j.neunet.2021.10.003.
5. V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare and J. Pineau, "An Introduction to Deep Reinforcement Learning," Foundations and Trends in Machine Learning, vol. 11, 2018. (https://www.sciencedirect.com/science/article/pii/S0893608021003944)
6. Shibl, Mostafa & Ismail, Loay & Massoud, Ahmed. (2020). Machine Learning-Based Management of Electric Vehicles Charging: Towards Highly-Dispersed Fast Chargers. Energies. 13. 10.3390/en13205429. https://www.researchgate.net/publication/344712843_Machine_Learning-Based_Management_of_Electric_Vehicles_Charging_Towards_Highly-Dispersed_Fast_Chargers
7. Arwa, E. (2020). Reinforcement Learning Approaches to Power Management in Grid-tied Microgrids: A Review. 2020 Clemson University Power Systems Conference (PSC), Clemson, SC, USA, 1–6. https://doi.org/10.1109/PSC50246.2020.9131138.
8. Rigas, E. S., Ramchurn, S. D., & Bassiliades, N. (2015). Managing Electric Vehicles in the Smart Grid Using Artificial Intelligence: A Survey. IEEE Transactions on Intelligent Transportation Systems, 16(4), 1–17. https://doi.org/10.1109/TITS.2014.2376873 https://www.academia.edu/18155827/Managing_Electric_Vehicles_in_the_Smart_Grid_Using_Artificial_Intelligence_A_Survey

9. Valogianni, Konstantina & Ketter, Wolfgang & Collins, John. (2013). Smart charging of electric vehicles using reinforcement learning. AAAI Workshop - Technical Report. 41-48. https://www.researchgate.net/publication/286726772_Smart_charging_of_electric_vehicles_using_reinforcement_learning
10. Qi, X., Wu, G., Boriboonsomsin, K., Barth, M. J., & Gonder, J. Data-Driven Reinforcement Learning–Based Real-Time Energy Management System for Plug-In Hybrid Electric Vehicles. Transportation Research Record: Journal of the Transportation Research Board, 2572(1), 1–8. https://doi.org/10.3141/2572-01
11. H. Li, Z. Wan and H. He, "Constrained EV Charging Scheduling Based on Safe Deep Reinforcement Learning," in IEEE Transactions on Smart Grid, vol. 11, no. 3, pp. 2427-2439, May 2020, doi: 10.1109/TSG.2019.2955437. https://ieeexplore.ieee.org/document/8910361
12. Wan, Zhiqiang & Li, Hepeng & Prokhorov, Danil. (2018). Model-Free Real-Time EV Charging Scheduling Based on Deep Reinforcement Learning. IEEE Transactions on Smart Grid. PP. 1-1. 10.1109/TSG.2018.2879572. https://www.researchgate.net/publication/328764282_Model-Free_Real-Time_EV_Charging_Scheduling_Based_on_Deep_Reinforcement_Learning
13. Abdullah, Heba M., Adel Gastli and Lazhar Ben-Brahim. “Reinforcement Learning Based EV Charging Management Systems–A Review.” IEEE Access 9 (2021): 41506-41531. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9371688
14. Lee, Jaehyun, Eunjung Lee, and Jinho Kim. 2020. "Electric Vehicle Charging and Discharging Algorithm Based on Reinforcement Learning with Data-Driven Approach in Dynamic Pricing Scheme" Energies 13, no. 8: 1950. https://doi.org/10.3390/en13081950 https://www.mdpi.com/1996-1073/13/8/1950/htm
15. Kang Wang, Haixin Wang, Junyou Yang, Jiawei Feng, Yunlu Li, Shiyu Zhang, Martin Onyeka Okoye, Electric vehicle clusters scheduling strategy considering real-time electricity prices based on deep reinforcement learning, Energy Reports, Volume 8, Supplement 4, 2022, Pages 695-703, ISSN 2352-4847, https://doi.org/10.1016/j.egyr.2022.01.233 (https://www.sciencedirect.com/science/article/pii/S2352484722002335)
16. Valogianni, Konstantina & Ketter, Wolfgang & Collins, John. (2013). Smart charging of electric vehicles using reinforcement learning. AAAI Workshop - Technical Report. 41-48. https://www.researchgate.net/publication/286726772_Smart_charging_of_electric_vehicles_using_reinforcement_learning
17. Felix Tuchnitz, Niklas Ebell, Jonas Schlund, Marco Pruckner, Development and Evaluation of a Smart Charging Strategy for an Electric Vehicle Fleet Based on Reinforcement Learning, Applied Energy, Volume 285, 2021, 116382, ISSN 0306-2619, https://doi.org/10.1016/j.apenergy.2020.116382 https://www.sciencedirect.com/science/article/pii/S0306261920317566
18. Ruzmetov, A., Nait-Sidi-Moh, A., Bakhouya, M., & Gaber, J. (2013). Towards an optimal assignment and scheduling for charging electric vehicles. 2013 International Renewable and Sustainable Energy Conference (IRSEC). https://doi.org/10.1109/IRSEC.2013.6529691
19. Qin, Hua & Zhang, Wensheng. (2011). Charging scheduling with minimal waiting in a network of electric vehicles and charging stations. Proc. 8th ACM VANET. 51-60. 10.1145/2030698.2030706. https://www.researchgate.net/publication/220926703_Charging_scheduling_with_minimal_waiting_in_a_network_of_electric_vehicles_and_charging_stations

20. M. M. De Weerdt, E. H. Gerding, S. Stein, V. Robu, and N. R.Jennings, "Intention-aware routing to minimise delays at electric vehicle charging stations," in Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, ser. IJCAI'13. AAAI Press,2013, pp. 83–89. https://doi.org/10.1145/2516911.2516923, https://core.ac.uk/download/pdf/77010611.pdf
21. Yongguang Liu, Wei Chen, and Zhu Huang, "Reinforcement Learning-Based Multiple Constraint Electric Vehicle Charging Service Scheduling" (2021) https://doi.org/10.1155/2021/1401802 https://www.hindawi.com/journals/mpe/2021/1401802/
22. Shahriar, Sakib & Al-Ali, A. & Osman, Ahmed & Dhou, Salam & Nijim, Mais. (2020). Machine Learning Approaches for EV Charging Behavior: A Review. IEEE Access. 8. 168980-168993. 10.1109/ACCESS.2020.3023388. https://www.researchgate.net/publication/344615813_Machine_Learning_Approaches_for_EV_Charging_Behavior_A_Review
23. Mnih, V., Kavukcuoglu, K., Silver, D. et al. Human-level control through deep reinforcement learning. Nature 518, 529–533 (2015), https://doi.org/10.1038/nature14236, https://web.stanford.edu/class/psych209/Readings/MnihEtAlHassibis15NatureControlDeepRL.pdf
24. Mnih, Volodymyr & Kavukcuoglu, Koray & Silver, David & Graves, Alex & Antonoglou, Ioannis & Wierstra, Daan & Riedmiller, Martin. (2013). Playing Atari with Deep Reinforcement Learning. https://arxiv.org/pdf/1312.5602.pdf
25. Takuma Seno, online resource: Welcome to Deep Reinforcement Learning part 1 – DQN, 2017, Retrieved from: https://towardsdatascience.com/welcome-to-deep-reinforcement-learning-part-1-dqn-c3cab4d41b6b
26. Lee, K.-B.; A. Ahmed, M.; Kang, D.-K.; Kim, Y.-C. Deep Reinforcement Learning Based Optimal Route and Charging Station Selection. Energies 2020, 13, 6255. https://doi.org/10.3390/en13236255
27. Zheng Chen, Hengjie Hu, Yitao Wu, Yuanjian Zhang, Guang Li, Yonggang Liu,Stochastic model predictive control for energy management of power-split plug-in hybrid electric vehicles based on reinforcement learning, Energy, Volume 211, 2020, 118931,ISSN 0360-5442, https://doi.org/10.1016/j.energy.2020.118931. (https://www.sciencedirect.com/science/article/pii/S0360544220320387)