9-23-2020

Autonomous PEV Charging Scheduling Using Dyna-Q Reinforcement Learning

Fan Wang

Jie Gao

Mushu Li

Lian Zhao

IEEE Transactions on Vehicular Technology, Vol. 69, No. 11 (23 September 2020): 12609-12620. DOI.

Autonomous PEV Charging Scheduling Using Dyna-Q Reinforcement Learning

Fan Wang
Business Intelligence Department, Bell Canada, ON, Canada
Jie Gao
Department of Electrical and Computer Engineering, Marquette University, Milwaukee, WI, USA
Mushu Li
Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada
Lian Zhao
Department of Electrical, Computer, and Biomedical Engineering, Ryerson University, Toronto, ON, Canada

Abstract:

This paper proposes a demand response method to reduce the long-term charging cost of single plug-in electric vehicles (PEV) while overcoming obstacles such as the stochastic nature of the user's driving behaviour, traffic condition, energy usage, and energy price. The problem is formulated as a Markov Decision Process (MDP) with an unknown transition probability matrix and solved using deep reinforcement learning (RL) techniques. The proposed method does not require any initial data on the PEV driver's behaviour and shows improvement on learning speed when compared to a pure model-free reinforcement learning method. A combination of model-based and model-free learning methods called Dyna-Q reinforcement learning is utilized in our strategy. Every time a real experience is obtained, the model is updated, and the RL agent will learn from both the real experience and “imagined” experiences from the model. Due to the vast amount of state space, a table-lookup method is impractical, and a value approximation method using deep neural networks is employed for estimating the long-term expected reward of all state-action pairs. An average of historical price and a long short-term memory (LSTM) network are used to predict future price. Simulation results demonstrate the effectiveness of this approach and its ability to reach an optimal policy quicker while avoiding state of charge (SOC) depletion during trips when compared to existing PEV charging schemes.

SECTION I. Introduction

Plug-in electric vehicles (PEVs) are gaining popularity due to their lower usage costs, better efficiency, less air and noise pollution compared to traditional gasoline vehicles, which soon makes the PEVs an integral part of the transportation system and smart grid [1]–[2][3]. Due to the increasingly wider adoption of PEVs, scheduling charging/discharging of PEVs has become ever more important in the evolution of smart grid. There are many aspects of scheduling the charging/discharging of PEVs, such as from the perspective of individual drivers or the perspectives of an entire PEV fleet. [4]. Finding an optimal charging/discharging schedule to minimize PEV charging cost is challenging, as there are many unknown parameters such as road conditions, driving behaviour of given drivers, and future energy prices. Furthermore, charging PEVs during peak load time will cause a higher peak demand [5]. The PEV's charging schedule strategy can take advantage of real-time electricity prices by charging during off-peak hours, which alleviates load during peak time and reduces charging cost [6]. On top of reducing costs, the PEV can also discharge energy by utilizing vehicle-to-grid (V2G) technology to earn a profit when energy prices are high [7].

Many methods to reduce the cost of charging PEV fleets have been proposed in literature [8]–[9][10][11][12][13]. An optimal charging scheduling strategy for charging a large PEV fleet to reduce cost and supply the requested power to the grid is developed using partial differential equation in [8]. A fully distributed solution for solving PEV fleet's Cooperative Charging (PEV-CC) problem is proposed in [9]. A multi-agent distributed approach to solving the PEV-CC problem is developed in [10]. A game theoretic approach to finding optimal bidding strategies for electric vehicle aggregators with variable energy sources is formulated as a mathematical programming with equilibrium constraints in [11]. A bidding strategy with the objective of minimizing charging costs for a vehicle fleet is studied in [12], but it does not consider Vehicle-to-Grid mode. In [13], a decentralized real-time PEV power allocation scheme is proposed. All of the above methods take into consideration the PEV's charging problem from a PEV fleet's perspective but not from the perspective of an individual driver.

Numerous optimization methods to schedule single PEVs have been investigated in [14]–[15][16][17][18][19][20]. An optimization approach to reduce cost of charging based on battery degradation is proposed in [14]. Day-ahead charging scheduling using information gap decision theory based approach is introduced in [15]. In [16], an electric vehicle aggregator participating in electricity market, using a two-level optimization problem in the framework of model predictive control is analysed. It is found this method can reduce cost compared to the stochastic method and deterministic predictive method. Another two-stage optimization method for reducing PEV charging cost for the workplace is proposed in [17]. In [18], a real-time charging scheme is proposed. The problem is formulated as a binary optimization problem. A convex relaxation method is developed to reduce the computation complexity of the algorithm. In [19], a task time allocation and reward scheme for advertising PEV charging station information is proposed. PEV charging scheduling schemes can take advantage of the Hybrid Machine Learning Model proposed in [20] to predict the power consumption on trips. None of these approaches take the stochastic nature of a PEV's usage into account, as the problem is too complex due to the variations in the driver's routine, traffic conditions, and energy prices.

Scheduling PEV charging/discharging from an individual driver's perspective is gaining attention in recent years. A route selection and charging navigation optimization model with the aid of crowd sensing has been proposed in [21]. The study of the charging scheduling problem under a parking garage scenario that aims to promote the total utility for the charging operator subject to the time-of-use pricing has been conducted in [22]. Methods using artificial intelligence such as reinforcement learning (RL) or artificial neural network to solve the challenges of statistical features are also gaining popularity in research [23]–[24][25]. Methods using existing user behaviour data to schedule a single PEV are examined in [26]–[27][28]. The authors of [26] use an artificial neural network trained using historical household power comsumption and EV energy demand data with two hidden layers to predict whether the PEV should charge or discharge. In [27], an off-line RL charging scheduling using fitted Q-iteration is proposed. The problem is formulated as a Markov Decision Process (MDP). This proposal achieved good results under the assumption that the user behaviour is known ahead of time. Two infinite horizon average cost MDP formulations are described for both hybrid vehicles and PEVs in [28]. A method to minimize the total overhead from users' perspective by leveraging the techniques of software defined networking and vehicular edge computing to investigate a joint problem of fast charging station selection and EV route planning using deep reinforcement learning has been proposed in [29]. A hybrid smart grid communication architecture integrating fiber optic and WiFi-based mesh networks for data acquisition in smart grid, which aids many artificial intelligence methods that require big data, has been proposed in [30]. The MDPs are built from historical data on vehicle usage. In the real world, user's previous driving behavior is often unknown and varies from owner to owner. These off-line methods are slow to adapt if there are changes to the PEV owner's driving routine.

Model-free RL methods are developed in [31]–[32][33]. Q-Table based method where electricity price and time are discretized is utilized in [31] to discover an optimal demand response. However, this method becomes impractical when energy price and time have a large number of states. The authors of [32] solve the problem of having a large number of states by using a linear approximator to approximate Q-values. However, this method can only approximate linear functions, while such relationships are generally non-linear in a real life PEV charging scenario. In [33], the PEV charging scheduling problem is formulated as an MDP and an optimal charging policy is found using a model-free deep-Q network with memory replay. This method takes into account the user's stochastic behaviour and the fluctuating energy price, and is able to achieve good results. However, the simulation assumes that the PEV's battery SOC will never have insufficient energy for trip, and ignores the relationship between trip length and energy usage's effect on the PEV's battery SOC before and after trips. In addition, the suggested method does not take full advantage of the PEV owner's routine behaviour. Using the above strategy under a more realistic scenario, the PEV can run out of battery during trips quite often. Furthermore, a pure model-free RL strategy takes a long time to discover an optimal charging policy.

In this paper, a cost-efficient PEV charging/discharging scheduling strategy is proposed from the PEV owner's perspective. The proposed strategy fulfills the usage requirements of an individual PEV owner and takes full advantage of the fluctuating energy prices to reduce the PEV's charging cost by buying or selling electricity when prices are low or high respectively, such as methods shown in [34] and [35]. The problem is formulated as an MDP. A combination of a model-based and a model-free RL method called Dyna-Q algorithm is utilized. A modified version of deep-Q network is also utilized to decide the optimal charging action when given the current time of day, day of the week, current SOC of the PEV, current energy price, and the difference between the current energy price and the predicted energy price 6 hours into the future. The modified deep-Q network uses model generated experiences instead of memory replay to learn a charging scheme that reduces the number of times a PEV's battery depletes during trips. The approach considers a real-life scenario that takes into consideration the relationship of battery SOC and the variability in trip lengths. The proposed method does not require any PEV owner's historical driving data. The effectiveness of this approach is proven in simulations. The simulation shows that, with appropriate initial parameter values for the models, our approach can ensure that the PEV never fully depletes its SOC during any trips, and significantly reduces the PEV owner's charging cost and the burden of being stuck on the road.

The contributions of this paper are as follows:

- A single PEV charging problem from the owner's perspective is formulated as an MDP. The owner's driving routine, the relationship between battery SOC and trips, and fluctuating energy prices are taken into account.
- An optimal policy is discovered using a three-layer deep-Q network. The neural network can generate an optimal charging policy based on PEV owner's driving behaviour and fluctuating energy price while keeping the PEV battery SOC above zero during trips.
- A combination of model-based and model-free RL is utilized in training the deep-Q network. This ensures that an optimal charging policy is discovered quickly. The model for model-based RL is updated from real driving experience after its initial creation. The RL agent will learn from both real experience and experience generated from the model.

Although our problem is formulated to be for at-home PEV charging, this method can be applied to other scenarios, such as charging at the workplace, with possible extensions. This is possible due to the fact that the probability of the PEV plug-in and plug-out are stored for every minute of the week. For instance, if we want to take the idle/busy status or current wait times of chargers at specific workplaces into account for our charging strategy, we can make slight modifications to our method to incentivize the RL agent to charge when idle chargers are scarce. For example, if the chargers at the driver's workplace are often busy, then the RL agent would learn to charge the PEV as much as possible when plugged in. We can include additional parameters in our state space and modify our reward function to penalize the RL agent for discharging during times where the driver is specifically waiting for the PEV to charge.

The rest of the paper is organized as follows. The system model for the charging/discharging of a single PEV is formulated as an MDP in Section II. The deep-Q network is discussed in detail and its application in choosing the optimal action when given a state is proposed in Section III. The model generation and model update for the model-based portion of the deep-Q network update is described in Section IV. Simulation results that show the effectiveness of this approach is presented in Section V. Section VI concludes this paper.

SECTION II. System Model

The problem of finding a PEV charging/discharging strategy to minimize the overall long-term charging cost for the PEV owner is formulated as a finite MDP with discrete time steps and unknown transition probabilities. We define short-term charging cost as the immediate minutely cost of charging, and long-term charging cost as the cumulative charging cost of the PEV over a period that is longer than a week. MDP is an effective method for decision making under uncertainty [36]. An MDP is a 4-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ is a finite set of states; $\mathcal{A}$ is a finite set of actions; $\mathcal{P}$ is a set of transition probabilities that reflects the probability of arriving at a specific next state S_{t+1}, after taking an action a in a previous state $\mathcal{S}$ for all states and actions; and $\mathcal{R}$ is a set of rewards that contains all the immediate reward after transitioning from any one state to any next state from taking any actions.

A. MDP State Space

The state space for the proposed MDP formulation is defined as follows: a state $\mathcal{S}$ encapsulates six variables: $S = [d\ m\ c\ p\ \Delta\ P\ k]$. The method of price estimation for ΔP can be averaging historical prices or more advanced methods such as the long-short term memory (LSTM) neural network. The long short-term memory neural network proposed in [33] is a more accurate method of price prediction. There are no state representations for when the PEV is unplugged. Specifically, if the PEV becomes unplugged at time t (when owner takes a trip), the next state S_{t+1} will be when the PEV is plugged in again. There are predictable patterns in the user's weekly driving routines, and with the six variable state space, the PEV owner's driving behavior should be sufficiently represented. Since the MDP has unknown transition probabilities, $\mathcal{P}$ will be "learned" from real-life experience using RL techniques. The transition probabilities are taken into consideration implicitly by the deep-Q network when deciding on the optimal policy, which will be discussed in Section III.

B. MDP Action Space

The action space A consists of these actions: discharge, idle, or charge, represented by $a = -1$, $a = 0$, and $a = 1$ respectively. The PEV cannot exceed its maximum battery capacity when charging, or its minimum battery capacity when discharging, shown as the following constraint:

$$0 \leq c + r \leq 1.$$

(1)

Because of the above constraint, there are only two valid actions in the states with $c = 0$ or $c = 1$.

C. MDP Reward

The immediate reward of an action is the cost of charging or discharging, shown as:

$$R = -p \cdot a \cdot r,$$

(2)

The reward has a negative sign because if a charging action is taken, ie. $a = 1$, then the reward is negative, as a cost has incurred for charging the PEV. Similarly, discharging the vehicle will yield a positive reward. If the PEV owner takes a trip, and has insufficient energy to reach destination, then a penalty $-P$ plus the cost of recharging during a trip $-u$ is obtained as the reward. To summarize:

$$R = \begin{cases} -p \cdot a \cdot r, \text{ if } d_{t+1} - d_t = 0 \text{ and} \\ m_{t+1} - m_t = 1; \text{or} \\ d_{t+1} - d_t = 1 \text{ and } m_{t+1} = 0 \\ -P - u, \text{ if } m_{t+1} - m_t \neq 0 \text{ and } k = 1;\ \text{or} \\ d_{t+1} - d_t \neq 0 \text{ and } k = 1. \end{cases}$$

(3)

D. MDP State Transition

When the PEV is plugged in at both state S_t and the next state S_{t+1}, then the day, time and the PEV's battery SOC are fully deterministic. The day and the minute of the next state S_{t+1} are:

$$d_{t+1} = \begin{cases} d_t, & \text{if } t \neq 1439 \\ d_t + 1, & \text{else,} \end{cases}$$
$$m_{t+1} = \begin{cases} m_t + 1, & \text{if } t \neq 1439 \\ 0, & \text{else,} \end{cases}$$

(4)(5)

where the number 1439 is the last minute of the day, and a new day starts when t reaches 1440.

The battery SOC of the next state is:

$$c_{t+1} = \begin{cases} 0, & \text{if } c_t = 0, a = 0 \\ 1, & \text{if } c_t = 1, a = 0 \\ c_t + a \cdot r, & \text{else.} \end{cases}$$

(6)

The electricity price of the next state is unknown, therefore the transition probabilities $\mathcal{P}$ is also unknown.

If the PEV owner takes a trip, then the day, time, and the PEV's battery SOC are no longer deterministic, as knowledge of the duration of the trip and the battery charges used are unknown. Due to uncertainties in road conditions, nature of user's trips, and energy price, it becomes difficult to determine S_{t+1}. This challenge is resolved using deep-Q network which is discussed in Section III.

E. State-Action Quality

Since the transition probability of the MDP is unknown and there exists a large state-action space, we use a deep reinforcement learning strategy to discover the optimal charging policy.

A selection criteria is needed to choose an action for a given state. Knowledge of the short term reward $\mathcal{R}$ is insufficient when choosing the optimal action. Therefore a long-term expected reward is required. The Bellman equation represents the long-term expected reward when following a policy as:

$$Q^{\Pi}(S_t, a_t) = E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ..].$$

(7)

The PEV charging/discharging policy Π selects an action for each state. The variable γ represents the weight of future rewards, and takes a range from 0 to 1. Having a discount factor of 1 means that the future and present has the same importance, and conversely, having a discount factor of 0 means that the future is of no importance. The goal of reinforcement learning is therefore to find the policy that maximizes $Q(S_t, a_t)$ at every time step t:

$$\Pi'(s) = \underset{a \epsilon A}{\operatorname{argmax}}\ Q^{\Pi}(s, a).$$

(8)

SECTION III. Deep Q Network

A traditional Q-table update method, in which the Q value of every state-action pair is stored in a table, updates according to the Q-learning algorithm [37]:

$$Q(S_t, a_t) \leftarrow Q(S_t, a_t) + \alpha(R_t \\ +\gamma \max_{a_{t+1}} Q(S_{t+1}, a_{t+1}) - Q(S_t, a_t)).$$

(9)

The update requires the current state St, the action taken at, the immediate reward Rt, the next state S_{t+1}, and the action a_{t+1} to be taken in S_{t+1} that yields the highest state-action pair value $Q(S_{t+1}, a_{t+1})$. However, the Q-table method is impractical for the PEV charging problem as the state-action space is too vast. Therefore we use a neural network to estimate the Q value instead, as this allows the approximation of Q value of unvisited state-action pairs. This method is called the Deep-Q Network (DQN) and is used to approximate the $Q(S_t, a_t)$ values of all existing state-action space. The DQN will update from both real and model generated experience as this allows a reduced learning time when compared to a pure model-free strategy where only real experiences are used for training. For a DQN, instead of updating the table value of $Q(S_t, a_t)$, the weights θ, of the neural network are updated instead. Traditionally, the stochastic gradient descent algorithm is used to update the neural network weights. But recent studies have found that the adaptive moment estimation (ADAM) optimizer can reach convergence faster when the user behaviour is stationary [38]. The ADAM update performed seeks to minimize the following objective function with respect to θ:

$$f(\theta) = \left(R_t + \gamma \max_{a_{t+1}} Q(S_{t+1}, a_{t+1}; \theta) - Q(S_t, a_t; \theta)\right)^2.$$

(10)

The update performed is therefore on the squared difference of the predicted value and the one step reward plus the predicted value of the next step with respect to the weights θ. This squared difference is used to update the neural network through back-propagation.

When choosing actions, the RL agent cannot simply choose the action that has the highest $Q(s, a)$ value, in other words, follow a greedy policy. The reason is that the agent following a greedy policy will not "explore" to discover unexplored states. Therefore an ϵ -greedy policy is taken instead, where the agent will have an ϵ chance of taking a random action, and $1 - \epsilon$ chance of taking the greedy action. To allow more exploration in the earlier episodes and more exploitation in the later episodes, an ϵ with exponential decay is used [39]. The variable ϵ is chosen and shown in equation (11):

$$\epsilon = \epsilon_{min} + (\epsilon_{max} - \epsilon_{min}) \cdot e^{-\lambda \cdot N},$$

(11)

where ϵ_{min} and ϵ_{max} are the minimum and maximum ϵ respectively; λ is the decaying factor; and N is the number of episodes.

Taking an ϵ -greedy policy is not ideal in the real world, because there is a probability of taking an unwanted random action. Therefore the RL agent will only follow the ϵ -greedy policy in model-based aspect of the RL scheme, where experiences are artificially generated. The RL agent will follow a pure greedy policy when taking actions in the real world.

There is a problem when updating the deep neural network using the ADAM optimizer. The problem lies in that the neural network used to predict $Q(S, a; \theta)$ is the same neural network that is currently being updated, and this will create instability as the neural network is chasing an ever moving target. To combat this issue, a separate target network is created, in which we hold the current weights θ' constant in a separate memory for T time steps, and update the actual weights θ of the neural network using the state-action value predicted with the memorized weights θ'. More precisely, instead of using $\mathrm{argmax}_{a_{t+1}} Q(S_{t+1}, a_{t+1}; \theta)$ to perform Q-learning update, we use $\mathrm{argmax}_{a_{t+1}} Q'(S_{t+1}, a_{t+1}; \theta')$; where Q' is predicted using memorized weights θ' from T time steps ago. After T time steps, we set the memory weights equal to the actual weights of the neural network, as in $\theta = \theta'$ [40].

Traditionally, deep-Q networks have a memory replay, where past experiences are reused to train the neural network. In our work, we have substituted the memory replay with user behaviour models, in other words, instead of training the neural network with past experiences, we will use model generated experiences instead. The reason we made this change, is to allow the DQN to experience PEV battery depletion before it happens in reality. If a traditional DQN with memory replay is used, then initially, the PEV must deplete its batteries many times for the DQN to be trained to avoid that.

The model creation and update process are discussed in detail in Section IV, and this approach is a modified version of the Dyna-Q algorithm. Every time a real experience occurs, the models are updated alongside the DQN, and the DQN then samples from a model generated mini batch of experiences and updates its weights. There is an endless loop in the algorithm because it updates on-line at every time step t, and the RL agent will use the most recently updated policy. Refer to Fig. 1 for a representation of the modified Dyna-Q algorithm. The full DQN update algorithm, with the modified Dyna-Q algorithm and separate target network, is shown in Algorithm 1.

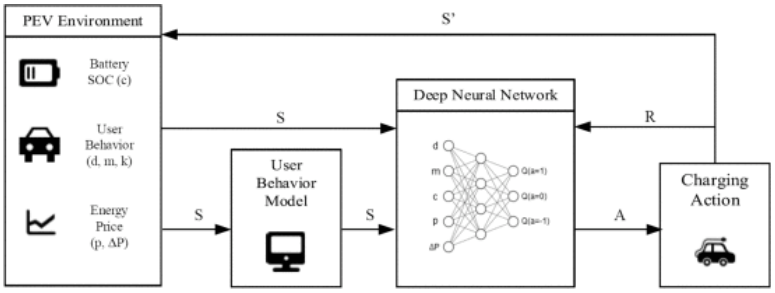

Fig. 1. Dyna-Q environment.

SECTION Algorithm 1: Deep Q-Network.

```
Initialize action-value function $Q$ with random weights $\theta$.
Initialize target action-value function $Q'$ with weights $\theta' = \theta$.
Initialize counter $c = 0$ and time step $U$.
loop
    Obtain current state $S_t$ from real experience.
    With probability $\varepsilon$ select a random charging action $a_t$.
    Otherwise predict all state-action value in state $S_t$ with neural network and select $a_t = argmax_a Q(S_t, a; \theta)$.
    Execute action at and observe reward $R_t$ and next state $S_{t+1}$.
    Perform Adam Optimization on $(R_t + \gamma \underset{a+1}{max} Q'(S_{t+1}, a_{t+1}; \theta') - Q(S_t, a_t; \theta))^2$ with respect to the weights $\theta$.
    Update model with the obtained $S_t, a_t, R_t, S_{t+1}$.
    Generate a mini batch of transitions $S_t, a_t, R_t, S_{t+1}$ from model.
    Perform Adam Optimization on mini batch generated from model with respect to weight $\theta$.
    Increment counter $c$.
    if $c = U$ then
```

```
        Set weights $\theta' = \theta$, Reset counter $c = 0$.
    else
        Do nothing.
    end if
end loop
```

SECTION IV. User Experience Model Generation

Model generated experiences are used to train the DQN alongside the real experiences. There are three aspects of a PEV driver that requires modeling:

- Probability of the PEV being plugged in during each time step: this models the probability that the PEV is plugged in at home for every minute of a day.
- Trip duration: this models the trip duration of trips, when the departure time is known.
- Charge used: this models the charge used for trips, when the departure time and the trip duration are known.

We use the above three aspects to represent a user's driving pattern, as they reflect when the PEV owner takes a trip, the trip duration, and the energy used per given trip. These three parameters implicitly take into consideration other factors related to a trip, such as the length of the trip, road conditions, and traffic conditions.

To give an example of what our model would capture: if the user goes to the grocery store on Saturday mornings, spends 20 minutes driving and 40 minutes shopping, and uses 10% charge, the model would capture all these parameters. The model is updated every time a real experience occurs. With enough similar experiences (assuming the user goes grocery shopping often on Saturday mornings), the model will be able to “guess” that a trip lasting an hour, uses around 10% of the total charge, will occur sometime on Saturday mornings. This is sufficient for our problem, even if the model does not directly contain any information regarding the nature of the trip or the actual length of the trip.

In the case when the driver's behaviour or road condition for trips varies over time, the variation would be reflected as variance in the model, and experience generated for that departure time will have a bigger variance.

The model can be initialized with values if there are some knowledge about the driver's behaviours, the details will be discussed in the following subsections.

A. PEV at Home Probability

The probability that the PEV is plugged in with respect to the time and day is stored in a look-up table. This table is continuously updated as data is gathered on the PEV owner's driving routine. The value stored for each time slot is the mean of all the times the PEV is plugged-in or away during that time. During each update, we assign a “1” for when the PEV is at home, and a “0” for when the PEV is away. Table II shows an example of the look-up table. For instance, according to Table II, the probability that the PEV is home and plugged-in at midnight on Sunday is 0.875.

TABLE I List of Notations

Variable	**Description**
S_t	Finite set of PEV environment states at time t
d	Day of the week, $d \epsilon [1, \ldots, 7]$

m	Minute of the day, $m \, \epsilon \, [0, \dots , 1439]$
c	Current PEV battery SOC in percentage, $c \, \epsilon \, (0, 1)$
p	Current electricity price in dollars per kWh
ΔP	An estimated difference between the energy price 6 hours in the future and the current energy price
k	Represents whether the PEV has ran out of battery on a given trip, $k \, \epsilon \, [0,1]$
r	The amount of electricity charged into or discharged from the battery in percentage relative to its nominal capacity, which can be positive or negative
a	Discharge/idle/charge action taken , $a \, \epsilon \, [-1 \, , 0, 1]$
$\prod$	PEV charging/discharging policy
γ	Reinforcement learning discount factor
$\mathcal{R}$	Immediate reward following an action at time step t
$Q(S_t, a_t)$	The expected sum of the discounted reward at time step t with a discount factor γ, when in states and taking action a
N	A counter that represents the total number of times a real experience was has occurred and gathered for a specific day, minute, departure time slot, or length of trip.
$P\,[d, m]$	The estimated probability that the PEV is plugged in during day d and minute m
g	Time departure interval for user behaviour models
l	Trip duration truncated to three lengths: short, medium, and long, used for user behaviour models
$\bar{v}[d, g]$	Mean of trip duration on day d and time departure interval g
$\sigma[d, g]$	Standard deviation of trip duration on day d and time departure interval g
$\overline{w}\,[d, g\,, l]$	Mean of charge used per trip on day d, time departure interval g, and trip duration bucket l
$\varphi[d, g, l]$	Standard deviation of charge use d per trip on day d, time departure interval g, and trip duration bucket l
H	A binary number that is obtained from real experience and represents the real plug-in status of the PEV, $H \, \epsilon \, [0, 1]$
V	Duration for the most recent trip obtained from real experience for the trip duration model
W	Charge used for the most recent trip obtained from real experience for the charge used per trip model

TABLE II PEV Plug-in Probability

Min, Day	Sun	M	T	W	Th	F	Sat
0	0.875	0.995	0.995	0.993	0.992	0.984	0.943
1	0.867	0.994	0.994	0.993	0.998	0.976	0.934
2	0.834	0.995	0.993	0.995	0.995	0.994	0.965
3	0.894	0.997	0.994	0.992	0.993	0.996	0.943
. .	. .	. .	..	..	..	..	..
..	..	..	..	..	..	..	..
1439	0.893	0.987	0.974	0.98 2	0.994	0.994	0.941

If nothing is known about the user's driving pattern initially, a constant can be assigned to each value in the table. Assigning a lower percentage to the PEV being plugged-in implies that the probability of the user taking a trip is high, and the RL agent will try to avoid the battery SOC running out during a trip by keeping battery SOC high at all times. Since updating the table requires computing the mean of all previous data, memory and computation requirement will grow larger as more user experience is acquired. A more practical on-line update algorithm, in which we can reduce the computing and memory requirements, is shown below,

$$N[d,m] \leftarrow N[d,m] + 1.$$
$$P[d,m] \leftarrow P[d,m] + \frac{1}{N[d,m]}(H - P[d,m]).$$

(12)(13)

Every time a real experience during a time slot occurs, the counter $N[d,m]$ increments, and the probability that the PEV is home or away is updated towards the real experience. At every time slot of every day of the week, the plug-in status of the PEV is obtained, and the probability table is updated. Over time, the PEV owner's plug-in behaviour can be fully described by the probability table.

This accurately represents the plug-in and plug-out behaviour of the PEV owner, and with this it can be assumed that when the PEV is plugged-out, the user has taken a trip.

The PEV plug-in probability table can be further applied to workplace charging, and it is not limited to at home charging. The underlying principle of Dyna-Q reinforcement learning and our formulation of the problem is applicable to any scenarios as long as the PEV has a plug-in and plug-out time.

B. Trip Duration

The duration of trips are based on the time and day of departure. We assume the user has routines for each specific day of the week. Specifically, if the user has work on Mondays at 9am, he will leave home at 8am, and come back home from work around 5:50pm to plug in his vehicle. Therefore we assume the duration of his trip will be around 9 hours and 50 minutes given any Monday with a departure time around 8am. Some variances can be expected for each trip time, but due to the routinely nature of the user's driving behaviour, each trip time will be centred around a mean. A truncated normal distribution for every departure time interval is used to model the probability of trip duration, with the variable bounded above 0, as trip duration cannot be less or equal to 0. The departure time intervals, represented by g, are set to every 15 minutes, ie. 8:00am - 8:15am, 8:15am-8:30am, and so on. The truncated normal distribution would have a mean of $\overline{v} = \frac{1}{n}\Sigma_{i=1}^{n} v_i$, and variance $\sigma^2 = \frac{1}{n}\Sigma_{i=1}^{n}(v_i - \overline{v})^2$, where vi is the length of current trip i. This function will be updated every time the PEV owner takes a trip. Updating the PDF requires the mean and variance of the trip duration, which needs to be computed using all previous trip samples. This requires ever growing memory and computation power. Therefore a more efficient on-line updating algorithm called Welford Online algorithm is used [41], shown as,

$$N[d,g] \leftarrow N[d,g] + 1.$$
$$\overline{v}[d,g]_i \leftarrow \overline{v} \quad [d,g]_i + \frac{1}{N[d,g]}(V - \overline{v}[d,g]_{i-1}).$$
$$\sigma^2[d,g]_i \leftarrow \sigma^2 \quad [d,g]_{i-1} + \frac{1}{N[d,g]}[(V - \overline{l}[d,g]_{i-1})(V - \overline{v}_i) - \sigma_{i-1}^2].$$

(14)(15)(16)

C. Charge Used Per Trip

The charge used per trip is dependent on the nature of the trip. The user's time and day of departure, and the length of the trip is a good indicator of the expected charge used on that trip. The time of departure is separated into morning, afternoon, evening, and night. The length of the trip is separated into short, medium, and long for trips under 3 hours, from 3 hours to 7 hours, and longer than 7 hours respectively. A truncated normal distribution for each day and time of departure, and trip length is used to model the probability of the charge that will be used on a particular trip, the distribution is bounded above 0, as no trip can use less or equal to 0 charge. More specifically, if a user drives his PEV every Monday to work from 8am to 5:50pm, he will use a similar amount of charge for that trip, and the normal distribution for departure on Monday mornings, with a long trip length, will be centered around that amount. Similar to trip length modelling, Welford's Online algorithm is used, and is shown below,

$$N[d,g,l] \leftarrow N[d,g,l] + 1.$$

$$\overline{w}[d,g,l]_i \leftarrow \overline{w} \qquad [d,g,l] + \frac{1}{N[d,g,l]}(M - \overline{w}[d,g,l])_{i-1}.$$

$$\varphi^2[d,g,l]_i \leftarrow \varphi^2 \qquad [d,g,l]_{i-1}$$

$$+\frac{1}{N[d,g,l]}[(W - \overline{w} \qquad [d,g,l]_{i-1})(W - \overline{w}_i) - \varphi^2_{i-1}].$$

(17)(18)(19)

D. Initial Parameter Values

The initial parameter values can be set to any arbitrary value, however the more knowledge we have on the user the better we will be able to set the initial parameter values. The initial parameter values are insignificant as they will adapt over time to truly match the user's driving behaviour from the updating policies shown in the previous subsections. The physical parameters that affect driving cycles, such as road and traffic conditions, are taken into account implicitly by the three models, which reflects a given user's PEV's probability of being plugged in, the mean and variance of the duration of a trip, and the mean and variance of charges used during a specific trip. The PEV plug-in probability model represent the plug-in probability of the PEV for every minute for the specific user. The trip duration model represent the duration of a trip taken, with trips segmented by their time of departure. Specifically, trips are segmented into morning, afternoon, and night, for each day of the week. To model charge used per trip, trips are segmented by their time of departure and length of trip. The time of departure is segmented the same way as the trip duration model, while the duration of trip is segmented into short, medium, and long.

Therefore each simulated trip, given its time of departure and the obtained PEV owner's driving experience, will accurately reflect the patterns created by factors such as road conditions, the driver's driving style, and traffic for any specific route and time of a given trip for that specific driver (since there are routine behaviours for any driver, and if a driver's routine is very stochastic, then we expect a high variance for the models). These three models implicitly, but sufficiently take into consideration the physical parameters of a trip, and through experience, will reflect accurately the driving behaviour of any given PEV owner.

E. Model State Generation

To generate a PEV charging/discharging experience from model, independent state transitions are created. Specifically, the state at time $S_t = [d_t, m_t, c_t, p_t, \Delta P_t]$ and the next state at time $S_{t+1} = [d_{t+1}, m_{t+1}, c_{t+1}, p_{t+1}, \Delta P_{t+1}]$ needs to be generated. The day, minute, and SOC of the initial state S_t, are generated with a uniform distribution, with $d\epsilon[1,..,7]$, $m\epsilon[0,..,1439]$, and $c\epsilon[0,..,100]$. The reason for uniformly generating the starting state S is to ensure the probability of exploring all states are equal. The price p is sampled from historical price that has the same day and time, and ΔP is also sampled from historical price averaged over its next 6 hours. The action a is chosen based on the ϵ -greedy policy using values predicted by the neural network. The variable k is represented implicitly by the state generation, since the probability of the PEV being plugged-in or not (taken on a trip) is fully modelled.

The next state S_t is generated by firstly checking if the PEV is plugged in, this is done by generating its plug-in status based on the plug-in probability table shown in Table II. If the PEV is plugged in at time t then the next state's day, minute, and SOC are deterministic and can be calculated using equations (4)–(6).

If the PEV is not plugged in, then a trip duration is sampled from the trip duration model based on the departure day and time. The charge used on that trip will be sampled according to the departure day and time, and the length of the trip. The next state's day and time is when the PEV is plugged-in again after the trip, which is deterministic when the trip length is known. The state SOC is the previous state's SOC minus the charge used during the trip. The electricity price of all the states is looked up from historical price.

This process of generating state transitions is repeated to generate a small number of transitions to be used for the training of the DQN.

SECTION V. Simulations

In this section, we discuss the experimental set-up, and evaluate the performance of the proposed approach compared to some existing approaches of single PEV charging.

A. Experimental Set-Up

1) PEV Environment

To evaluate the performance of the proposed method, we modelled the BYD e6, which has a battery capacity of 82 kWh. Battery capacities are growing ever-larger, and the BYD e6 has one of the largest battery capacities, which reflects the size of PEV batteries in the near future. The charge/discharge rate of the PEV is 19.2 kW, which is the current level 2 charging rate and most residential homes have the capability to install level 2 charging.

The minutely electricity price is taken from Energy Exchange Austria. Three months of electricity price data, from August 2018 to November 2018, is taken to find the average historical electricity price. Price taken from November 2018 to December 2018 is used as the real minutely price in the actual simulation.

Three types of PEV owner behaviour are simulated:

- Normal user behavior: the PEV owner leaves his home from 8:00 am to 9:00 am on weekdays, and uses a charge between 25% to 30% with a trip duration of 7.5 to 8.5 hours. On weekends the user leaves his home from 5:00 am to 5:00 pm and uses a charge between 15% to 50%, with a trip duration of 6 to 12 hours. The user's driving routine changes within that range from week to week.
- Stationary user behavior: the PEV owner leaves his home at 8:00 am on weekdays, and from 10:00 am on weekends. The trip duration on weekdays is 8 hours, and 4 hours on weekends. The charge used is based on the length of trip, and is between 25% to 28% on weekdays and between 25% to 28% on weekends. The owner driving behavior will remain exactly the same from week to week.
- Stochastic user behavior: the PEV owner leaves his home from 7:30 am to 9:00 am on weekdays, and from 5:00 am to 5:00 pm on weekends. The trip duration are between 7.5 hours to 9 hours on weekdays, and 3 hours to 15 hours on weekends. The charge used is based on the length of the trip, and is between 5% to 50% on weekdays, and between 3% to 40% on weekends. The driving behaviour of the PEV also changes every week, which makes it highly unpredictable.

All PEV simulation starts with an initial SOC of 100%, and the time starts on Monday at 12:00 am.

The reward received by the RL agent upon taking any charging/discharging action is the cost of electricity at that minute multiplied by the PEV's rate of charge. It is positive when discharging and negative when charging. The reward for idling, charging the PEV at full battery SOC, and discharging the PEV at 0 battery SOC is 0. Having the PEV battery running out of energy during trips will cause an inconvenience. In the best case scenario the PEV owner has to find a charging station to charge his vehicle, and in the worst case scenario the PEV owner must tow his vehicle. Therefore a reward of −35 plus the cost of charging is obtained as a reward when the PEV battery is depleted during trips. The −35 is a rough estimate on the inconvenience experienced by the owner when the PEV runs out of battery during trips, including the rare times when he needs to get his vehicle towed. If the RL agent's previous action before the trip was to charge the vehicle, then a reward of −27 plus the cost of charge used is given. This incentivizes charging when the battery SOC is near 0 and lets the RL agent learn to keep sufficient battery charge faster.

2) Neural Network

- Input layer has 5 nodes: time of day, day of week, current battery SOC, current electricity price, and average historical electricity price of the next 6 hours
- Output layer has 3 nodes: the state-action quality, $Q(s, a)$, of charging the PEV, remaining idle, and discharging the PEV
- Three fully connected hidden layers, first layer has 4 nodes, second layer has 3 nodes, and third layer has 2 nodes. This configuration seems to perform better than other tried configurations (50-50-50, 10-7-5, and 50-40-30)
- Rectified Linear activation function at each hidden layers, and no activation function at the output layer for regression

3) Training Process

The RL agent is trained over 40000 decision epochs. A decision epoch occurs when a charging/discharging action needs to be taken. When the PEV is plugged in, a charging/discharging action occurs every minute. However, when the PEV owner is on a trip, there are no actions to be taken. The next decision epoch occurs when the PEV is plugged in at home. This set-up means that the 40000 decision epochs span over a time frame of roughly 50000 real life minutes, as the additional 10000 minutes are spent on trips, where there are no decisions to be made.

At every decision epoch, the RL agent also samples from model generated experience. The model generates 32 experience per real experience.

B. Experimental Results

The proposed scheme is compared to five other charging schemes:

- Model-free DQN proposed in [33].
- Cheap Scheme: The PEV charges/discharges when the current electricity price is lower/higher than the historical average price for the same month and date in the previous year.
- Always Charge Scheme: The PEV always charges when it is plugged in.
- Low SOC and Cheap Scheme: The PEV will always charge when it has less than 20% battery SOC, and will charge/discharge when the electricity price is lower/higher than the historical average price otherwise.

Fig. 2 shows the configuration of the neural network used in the DQN for $Q(s, a)$ prediction. The neural network used for the DQN is configured as follows:

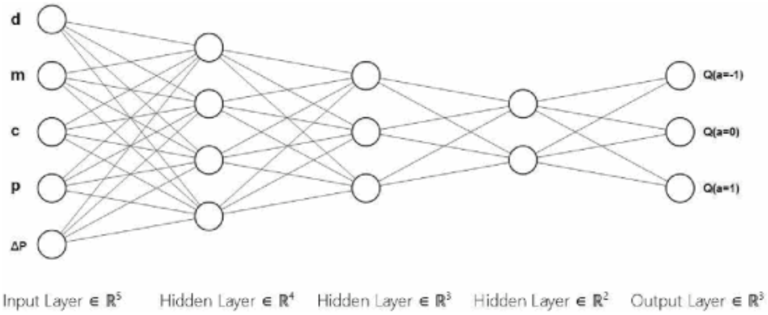
d
m
c
p
ΔP
Q(a=-1)
Q(a=0)
Q(a=1)
Input Layer ∈ $\mathbb{R}^5$
Hidden Layer ∈ $\mathbb{R}^4$
Hidden Layer ∈ $\mathbb{R}^3$
Hidden Layer ∈ $\mathbb{R}^2$
Output Layer ∈ $\mathbb{R}^3$

Fig. 2. Neural network configuration.

The cumulative reward of the RL agent when the PEV owner's driving nature is regular, stationary, and stochastic is shown in Fig. 3 to 5 respectively. The proposed scheme achieves the highest reward overall, and drastically outperforms a purely model-free DQN strategy. This is due to the fact that only 30 days are simulated, and this is far too little experience for a pure model-free DQN to discover a policy that ensures the PEV does not deplete its battery during trips. The biggest penalty that the model-free DQN strategy suffers from is its battery depletion during trips, as seen in Fig. 6. This scheme has ran out of battery during trips a total of 22 times in 30 days.

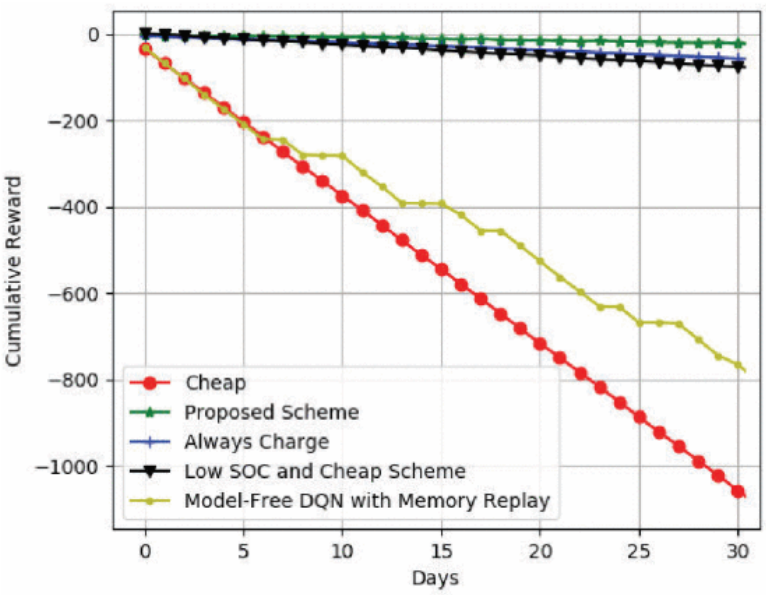

Fig. 3. RL agent cumulative reward for normal owner behaviour.

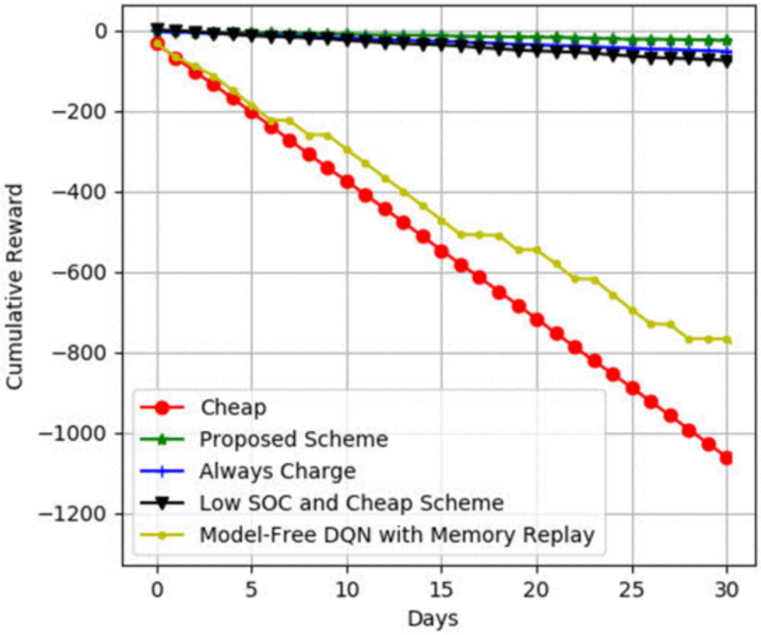

Fig. 4. RL agent cumulative reward for stationary owner behavior.

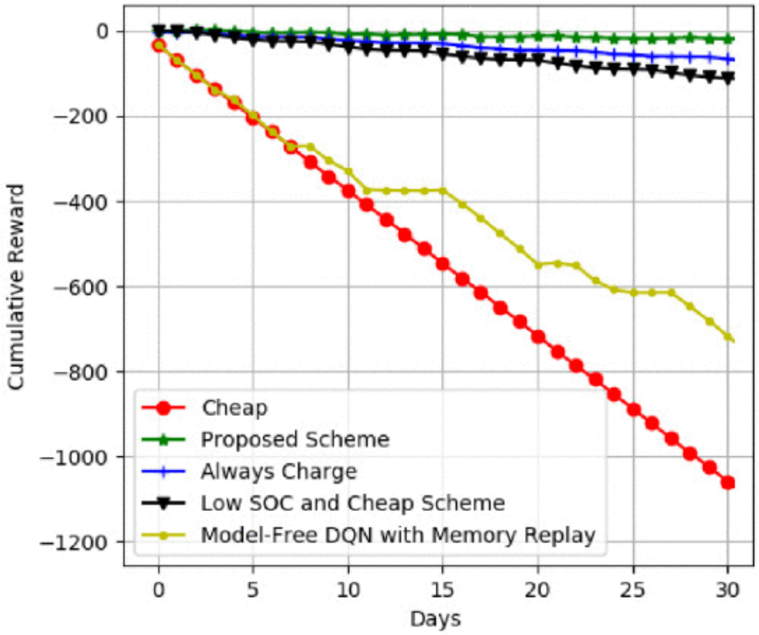

Fig. 5. RL agent cumulative reward for stochastic owner behavior.

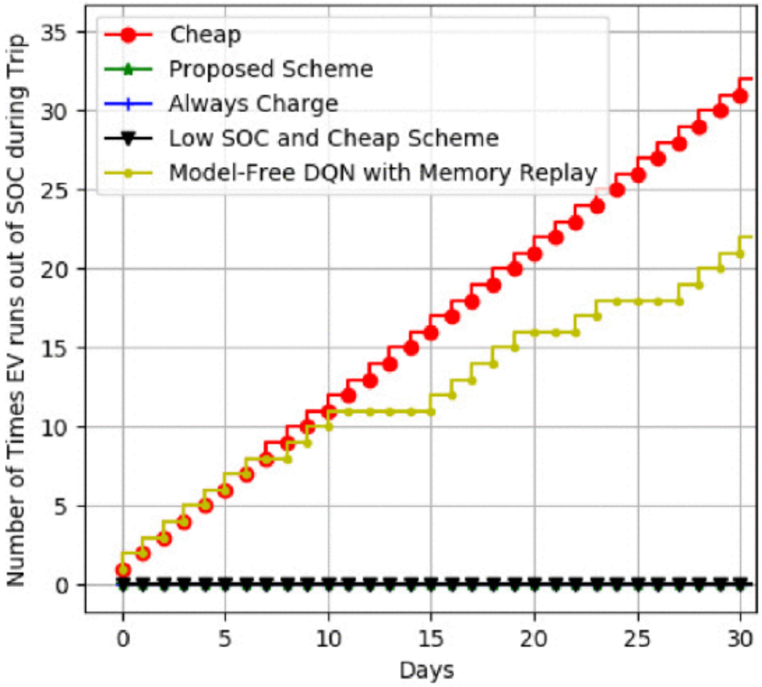

Fig. 6. Number of times PEV runs out of battery during trips.

Due to the nature of the always charge scheme and low SOC and cheap scheme, the PEV owner will never completely deplete the PEV battery during trips. The proposed scheme performs just as well in this regard, and never completely deplete its battery during trips, this is shown in Fig. 6. The above mentioned three schemes are overlapped in Fig. 6, due to the fact that all three schemes has ran out of battery exactly 0 times for the 30 simulated days.

The always charge scheme was able to outperform the low SOC and cheap scheme because the historical electricity prices are not always a good predictor of future prices, and some charging decisions made by the low SOC and cheap scheme are likely not optimal.

Although it may appear that the proposed scheme only outperforms the always charge scheme by a little, in actuality, if we observe Fig. 7, which is a close-up of the previous figures that also includes schemes using LSTM network predicted energy prices, the cumulative reward for the proposed scheme was −22, while the always charge scheme has a cumulative reward of −43. Therefore the proposed scheme actually reduces the cost of charging by 51% when compared with the always charge scheme.

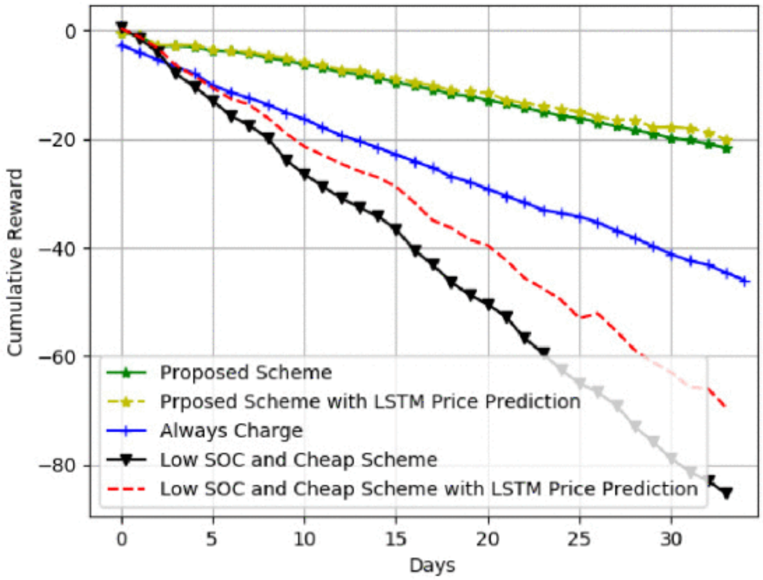

Fig. 7. Close-Up of relevant schemes with and without LSTM price prediction.

A LSTM neural network price prediction model is used to show the effects of a more accurate price prediction on our charging/discharging strategies. The neural network is trained from a year's worth of historical data, taken from Energy Exchange Austria, and trained using data from the period November 2016 to November 2017 as input and data from December 2017 as output. This neural network was used to predict energy prices for December 2018.

As seen in Fig. 7, the LSTM did not improve the model-free DQN scheme by much, this is due to the fact that having more accurate price prediction as an input to the DQN does not overcome the issue of model-free DQN learning slowly. If we take a closer look, in Fig. 7, we can see that a more accurate price prediction has a bigger effect on the low SOC and cheap scheme, this is because the low SOC and cheap scheme depends directly on the accuracy of price prediction. It is shown that using a LSTM price prediction has reduced the low SOC and charge scheme's charging cost by nearly 20%. On the other hand, a more accurate price prediction only moderately improved our proposed scheme, this is due to the fact that the more accurate price prediction is only one input to the DQN, and the DQN does not take the more accurate price into consideration directly, but indirectly through the DQN. It is shown that the LSTM price prediction improved the DQN model by roughly 10%. Although the improvement of LSTM in our proposed approach is smaller than the improvement in a rule-based approach, our scheme still significantly outperforms the low SOC and cheap scheme, due to our approach's ability to take into consideration the user's driving behaviour.

The initializing parameter values are set to what we assume to be an average user's behaviour. For instance, we assumed that there is a higher probability that the user will be home during evenings and nights, at 80%, and less probability for the user to be home during the day, at 20%. We also set the parameters of trip departure time to be around 8:00 am on weekdays and have a duration of 8 hours. For weekends we assumed the user would have a departure time around 11:00 am and have a trip duration of 5 hours. These parameters potentially explains the reason that the our model can consistently have sufficient charges for trips starting at day 0. Our assumption is not entirely accurate but is very similar to a normal working person's driving pattern.

The above simulations are done by setting the initial parameter values from assumptions made about the PEV owner, in other words, the initial parameter values are more ideal. Fig. 8 shows the daily reward for the model-free DQN scheme and our proposed scheme with non-ideal initial parameter values, in this case we assumed that the user would be home for over 90% of time, and each trip would only last 10 minutes and use a charge of

5%. We can assume that having a reward of around −35 means that the battery has been depleted during a trip. There we can see, that even our proposed scheme, under non-ideal initial parameter values, would deplete the PEV's batteries for many days initially. These settings caused our scheme to learn to avoid battery depletion during trips, because our model does not produce enough trip depletion experience for the DQN to “learn” to avoid it. However, we can also observe that in Fig. 8 that our method converges much quicker than a model-free DQN. The reason is that our scheme, for every minute, runs 32 simulated experiences, therefore it quickly discovers the huge negative reward associated with depleting energy during trips. On the other hand, the model-free DQN scheme must experience battery depletion many times in reality before it discovers the huge negative reward associated with it, and learns to avoid it.

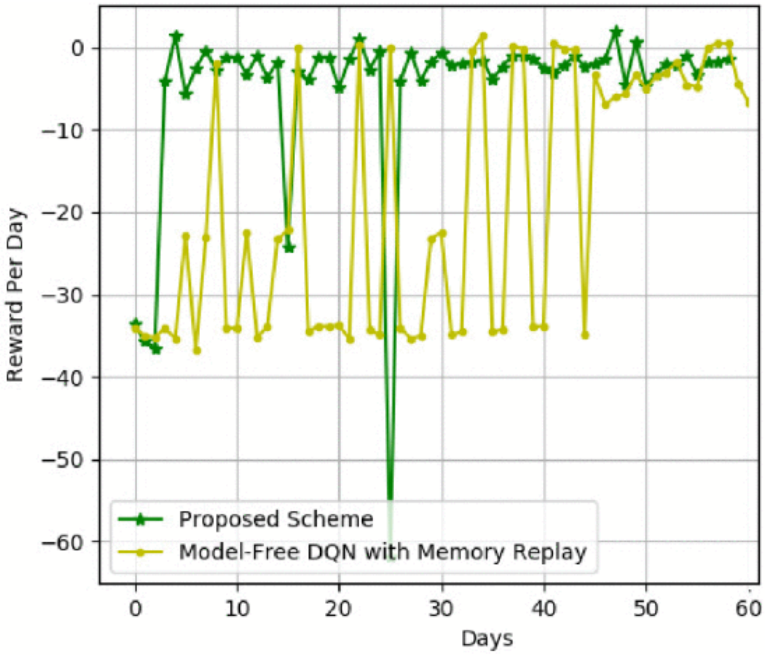

Fig. 8. Daily reward with Non-Ideal initial parameter values.

Fig. 9 demonstrates that our scheme improves faster when compared with a pure model-free method. The simulation compares the mean absolute error (MAE) of our scheme versus a pure model-free scheme over time. The MAE is defined as:

$$\text{Mar.} = \left| R_t + \gamma \underset{a_{t+1}}{max} Q(S_{t+1}, a_{t+1}; \theta) - Q(S_t, a_t; \theta) \right|.$$

(20)

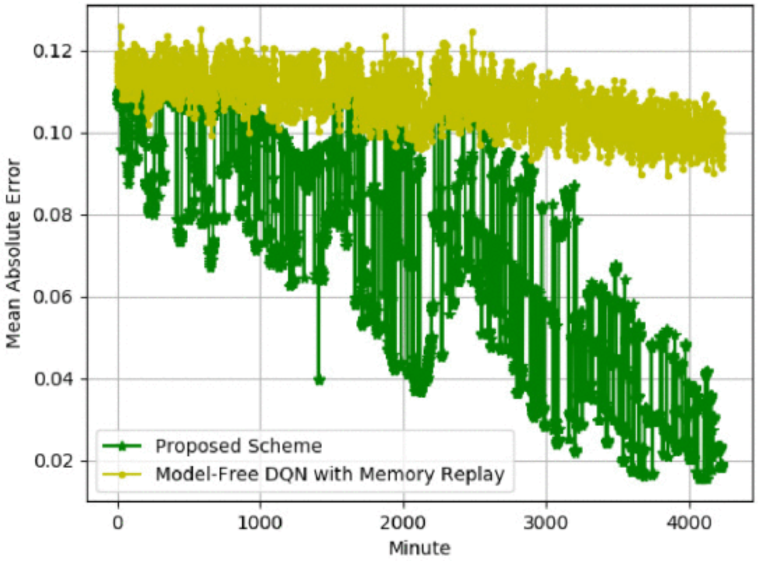

Fig. 9. Mean absolute error over time.

We can see that the error for our scheme decreases faster compared to the error of the pure model-free scheme. For the purpose of demonstration and to ensure that the graph is readable at a smaller scale, we have removed instances where the PEV have insufficient charges during a trip, as this will cause a much larger error and make the figure unreadable. Compared to Fig. 6, which demonstrates that our scheme can “learn” to avoid having insufficient charges on a trip, Fig. 9 demonstrates that our scheme is able to learn to buy/sell electricity according to the user's needs and energy prices more accurately over time.

The computational complexity of our approach should be considered to adapt practical applications. The LSTM price prediction model can be trained offline and thus the corresponding computational complexity is not a major concern. The time complexity of simulating the DQN for 30 days of real time was 1 hour and 37 minutes on an Intel i5-4670 K CPU, which has four cores. This equates to 0.054 seconds of computing for every minute that elapses in real time, which can be practically achieved as long as the PEV is equipped with a sufficient CPU.

SECTION VI. Conclusion

A method that combines both model-free and model-based reinforcement learning for PEV charging was proposed in this paper. The proposed method implicitly takes into consideration the user's driving behavior, traffic conditions, electricity price, and PEV energy usage. This method ensures that the owner has sufficient charge for trips, all the while charging the PEV cheaply according to the user's needs. The effectiveness of this method is demonstrated in simulations, where the target vehicle never runs out of battery during trips and sees less charging costs when compared with other methods.

Future works can extend to the use of multi-agent reinforcement learning to discover an optimal charging scheme for an entire community. The trade-off between reducing individual charging cost and community charging cost can be adjusted according to situation by adjusting the defined reward function for the multi-agent reinforcement learning environment. Another potential interest is in the development of a charging scheme that helps with load balancing. By adjusting the reward function to include the reward of RL agent behaviours which alleviate peak demand, an RL agent is able to find a balance between reducing owner cost and alleviating peak demand.

References

1. M. Li, J. Gao, N. Chen, L. Zhao and X. Shen, "Decentralized PEV power allocation with power distribution and transportation constraints", *IEEE J. Select. Areas Commun.*, vol. 38, no. 1, pp. 229-243, Aug. 2020.
2. L. Cai, J. Pan, L. Zhao and X. Shen, "Networked electric vehicles for green intelligent transportation", *IEEE Commun. Standards Mag.*, vol. 1, no. 2, pp. 77-83, Jul. 2017.
3. G. Tal and M. A. Nicholas, "Studying the PEV market in California: Comparing the PEV PHEV and hybrid markets", *Proc. World Elect. Veh. Symp. Exhib. (EVS27)*, pp. 1-10, Nov. 2013.
4. E. Sortomme and M. A. El-Sharkawi, "Optimal charging strategies for unidirectional vehicle-to-grid", *IEEE Trans. Smart Grid*, vol. 2, no. 1, pp. 131-138, Dec. 2011.
5. S. Shafiee, M. Fotuhi-Firuzabad and M. Rastegar, "Investigating the impacts of plug-in hybrid electric vehicles on power distribution systems", *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1351-1360, Sep. 2013.
6. T. Namerikawa, N. Okubo, R. Sato, Y. Okawa and M. Ono, "Real-time pricing mechanism for electricity market with built-in incentive for participation", *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 2714-2724, Nov. 2015.
7. J. R. Pillai and B. Bak-Jensen, "Integration of vehicle-to-grid in the western danish power system", *IEEE Trans. Sustain. Energy*, vol. 2, no. 1, pp. 12-19, Jan. 2011.
8. C. Le Floch, F. di Meglio and S. Moura, "Optimal charging of vehicle-to-grid fleets via PDE aggregation techniques", *Proc. Amer. Control Conf. (ACC)*, pp. 3285-3291, Jul. 2015.
9. J. Mohammadi, S. Kar and G. Hug, "Distributed cooperative charging for plug-in electric vehicles: A consensus+innovations approach", *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, pp. 896-900, Dec. 2016.
10. J. Mohammadi, M. G. Vayá, S. Kar and G. Hug, "A fully distributed approach for plug-in electric vehicle charging", *Proc. Power Syst. Comput. Conf. (PSCC)*, pp. 1-7, Jun. 2016.
11. H. Wu, M. Shahidehpour, A. Alabdulwahab and A. Abusorrah, "A game theoretic approach to risk-based optimal bidding strategies for electric vehicle aggregators in electricity markets with variable wind energy resources", *IEEE Trans. Sustain. Energy*, vol. 7, no. 1, pp. 374-385, Jan. 2016.
12. M. González Vayá and G. Andersson, "Optimal bidding strategy of a plug-in electric vehicle aggregator in day-ahead electricity markets under uncertainty", *IEEE Trans. Power Syst.*, vol. 30, no. 5, pp. 2375-2385, Sep. 2015.
13. M. Li and L. Zhao, "A decentralized load balancing approach for neighbouring charging stations via EV fleets", *Proc. IEEE 86th Veh. Technol. Conf. (VTC-Fall)*, pp. 1-5, Sep. 2017.
14. M. A. Ortega-Vazquez, "Optimal scheduling of electric vehicle charging and vehicle-to-grid services at household level including battery degradation and price uncertainty", *IET Gener. Transmiss. Distrib.*, vol. 8, no. 6, pp. 1007-1016, Jun. 2014.
15. J. Zhao, C. Wan, Z. Xu and J. Wang, "Risk-based day-ahead scheduling of electric vehicle aggregator using information gap decision theory", *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1609-1618, Jul. 2017.
16. Y. Zhao, C. Feng, Z. Lin, F. Wen, C. He and Z. Lin, "Development of optimal bidding strategy for an electric vehicle aggregator in a real-time electricity market", *Proc. IEEE Innovative Smart Grid Technol. Asia (ISGT Asia)*, pp. 288-293, May 2018.
17. D. Wu, H. Zeng, C. Lu and B. Boulet, "Two-stage energy management for office buildings with workplace EV charging and renewable energy", *IEEE Trans. Transp. Electrific.*, vol. 3, no. 1, pp. 225-237, Mar. 2017.
18. L. Yao, W. H. Lim and T. S. Tsai, "A real-time charging scheme for demand response in electric vehicle parking station", *IEEE Trans. Smart Grid*, vol. 8, no. 1, pp. 52-62, Jan. 2017.
19. M. Li, J. Gao, L. Zhao and X. Shen, "Task time allocation and reward scheme for PEV charging station advertising", *IEEE Int. Conf. Commun.*, May 2019.

20. B. Zheng, P. He, L. Zhao and H. Li, "A hybrid machine learning model for range estimation of electric vehicles", *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016.
21. H. Yang, Y. Deng, J. Qiu, M. Li, M. Lai and Z. Y. Dong, "Electric vehicle route selection and charging navigation strategy based on crowd sensing", *IEEE Trans. Ind. Inform.*, vol. 13, no. 5, pp. 2214-2226, Mar. 2017.
22. Z. Wei, Y. Li, Y. Zhang and L. Cai, "Intelligent parking garage EV charging scheduling considering battery charging characteristic", *IEEE Trans. Ind. Electron.*, vol. 65, no. 3, pp. 2806-2816, Aug. 2018.
23. N. Kato et al., "The deep learning vision for heterogeneous network traffic control: Proposal challenges and future perspective", *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 146-153, Dec. 2017.
24. Z-M. Fadlullah et al., "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems", *IEEE Commun. Surveys Tut.*, vol. 19, no. 4, pp. 2432-2455, May 2017.
25. N. Ye, X-M. Li, H. Yu, L. Zhao, W. Liu and X. Hou, "DeepNOMA: A unified framework for NOMA using deep multi-task learning", *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2208-2225, Jan. 2020.
26. S. Morsalin, K. Mahmud and G. Town, "Electric vehicle charge scheduling using an artificial neural network", *Proc. IEEE Innovative Smart Grid Technol. - Asia (ISGT-Asia)*, pp. 276-280, Nov. 2016.
27. A. Chiş, J. Lundén and V. Koivunen, "Reinforcement learning-based plug-in electric vehicle charging with forecasted price", *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 3674-3684, May 2017.
28. J. Donadee, M. Ilic and O. Karabasoglu, "Optimal autonomous charging of electric vehicles with stochastic driver behavior", *Proc. IEEE Vehicle Power Propulsion Conf. (VPPC)*, pp. 1-6, Oct. 2014.
29. J. Liu, H. Guo, J. Xiong, N. Kato, J. Zhang and Y. Zhang, "Smart and resilient EV charging in SDN-enhanced vehicular edge computing networks", *IEEE J. Sel. Areas Commun.*, vol. 38, no. 1, pp. 217-228, Jan. 2020.
30. H. Guo, J. Liu and L. Zhao, "Big data acquisition under failures in FiWi enhanced smart grid", *IEEE Trans. Emerg. Topics Comput.*, vol. 7, no. 3, pp. 420-432, Jul. 2019.
31. Z. Wen, D. O’Neill and H. Maei, "Optimal demand response using device-based reinforcement learning", *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2312-2324, Sep. 2015.
32. S. Vandael, B. Claessens, D. Ernst, T. Holvoet and G. Deconinck, "Reinforcement learning of heuristic EV fleet charging in a day-ahead electricity market", *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1795-1805, Jul. 2015.
33. Z. Wan, H. Li, H. He and D. Prokhorov, "Model-free real-time EV charging scheduling based on deep reinforcement learning", *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5246-5257, Sep. 2019.
34. A. Y. S. Lam and V. O. K. Li, "Opportunistic routing for vehicular energy network", *IEEE Internet Things J.*, vol. 5, no. 2, pp. 533-545, Sep. 2018.
35. N. Chen, J. Ma, M. Li, M. Wang and X. Shen, "Energy management framework for mobile vehicular electric storage", *IEEE Netw.*, vol. 33, no. 6, pp. 148-155, Oct. 2019.
36. M. Li, L. Zhao and H. Liang, "An SMDP-based prioritized channel allocation scheme in cognitive enabled vehicular ad hoc networks", *IEEE Trans. Veh. Tech.*, vol. 66, no. 9, pp. 7925-7933, Sep. 2017.
37. J. Liang and J. Xu, "A novel contour extraction approach based on Q-learning", *Proc. Int. Conf. Mach. Learn. Cybern.*, pp. 3807-3810, Aug. 2006.
38. S. Ruder, "An overview of gradient descent optimization algorithms", *CoRR*, vol. abs/1609.04747, 2016.
39. A. Masadeh, Z. Wang and A. E. Kamal, "Reinforcement learning exploration algorithms for energy harvesting communications systems", *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018.
40. S. Gu, T. P. Lillicrap, I. Sutskever and S. Levine, "Continuous deep Q-Learning with model-based acceleration", *CoRR*, vol. abs/1603.00748, 2016.
41. B. P. Welford, "Note on a method for calculating corrected sums of squares and products", *Technometrics*, vol. 4, no. 3, pp. 419-420, Aug. 1962.