This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2018.2879572, IEEE Transactions on Smart Grid

Model-Free Real-Time EV Charging Scheduling Based on Deep Reinforcement Learning

Zhiqiang Wan, *Student Member, IEEE,* Hepeng Li, *Member, IEEE,* Haibo He, *Fellow, IEEE,* and Danil Prokhorov, *Senior Member, IEEE*

***Abstract*—Driven by the recent advances in electric vehicle (EV) technologies, EVs become important for smart grid economy. When EVs participate in demand response program which has real-time pricing signals, the charging cost can be greatly reduced by taking full advantage of these pricing signals. However, it is challenging to determine an optimal charging strategy due to the existence of randomness in traffic conditions, user's commuting behavior, and pricing process of the utility. Conventional model-based approaches require a model of forecast on the uncertainty and optimization for the scheduling process. In this paper, we formulate this scheduling problem as a Markov Decision Process (MDP) with unknown transition probability. A model-free approach based on deep reinforcement learning is proposed to determine the optimal strategy for this problem. The proposed approach can adaptively learn the transition probability and does not require any system model information. The architecture of the proposed approach contains two networks: a representation network to extract discriminative features from the electricity prices and a Q network to approximate the optimal action-value function. Numerous experimental results demonstrate the effectiveness of the proposed approach.**

***Index Terms*—Deep reinforcement learning, model-free, EV charging scheduling.**

I. Introduction

WITH THE recent advances in EV technologies, EVs are becoming popular because of its various benefits [1], [2]. EVs provide a sustainable alternative to fossil-fuel vehicles and can significantly reduce the transport-related pollution. Another benefit of EVs is the cost reduction for consumers since it is cheaper to charge an EV than fill up with gasoline [3]. As the real-time electricity price has been adopted by many utility companies to encourage shifting energy usage to off-peak hours [4], the charging cost can be reduced by optimizing the charging schedules [5]. In addition, an EV can discharge energy back to the electric grid by working in the vehicle-to-grid (V2G) mode and thereby make money [6].

Due to the existence of randomness in traffic conditions, user's commuting behavior, and pricing process of the utility, EV arrival and departure time, EV energy consumption, and electricity prices are dynamic and time-varying. Therefore, efficiently managing EV charging/discharging to reduce the cost becomes challenging.

In recent years, numerous day-ahead scheduling approaches have been proposed for this problem [7]–[18]. For instance, in order to handle the uncertainty in electricity price, M. A. Ortega-Vazquez [7] developed a robust optimization approach for residential EV charging scheduling. Similarity, J. Zhao et al. [8] proposed an information-gap-decision based approach to deal with the uncertainty in electricity price and optimize day-ahead scheduling of EV fleet. In [9], [10], EV fleet was formulated as a probabilistic virtual battery model, and scenario-based robust approaches were proposed to deal with the uncertainty of the EV users' commuting behavior and the balancing requests. M. R. Sarker et al. [11] studied the day-ahead scheduling of battery swapping stations where the uncertainty of the battery demand and the electricity price was modeled by inventory robust optimization and multi-band robust optimization, respectively. Apart from the above robust optimization approaches, stochastic optimization is also widely used to handle the uncertainty. To name a few, in order to handle the randomness in the charging demand, D. Wu et al. [12] developed a two-stage stochastic optimization method for workplace EV charging management. In a similar way, Y. Guo et al. [13] proposed a two-stage framework for the economical operation of a microgrid-like EV parking deck while taking the intermittency of renewable outputs into account. I. Momber et al. [14] developed a two-stage stochastic linear programming to maximize the EV aggregator's profit in both day-ahead and balancing market while the uncertainties of EV fleet mobility and market price were considered. In [15], [16], the EV aggregator was considered to bid in the day-ahead market and offered ancillary services. Stochastic programming approaches were applied to manage the EV fleet charging while taking into account the randomness of the regulation signals. Although the aforementioned methods achieved some success in day-ahead charging/discharging scheduling, they may be unsuitable for real-time scenarios where the variations in EV charging demand and the electricity prices are much more complex.

Real-time scheduling strategies that can respond to dynamic charging demand and time-varying electricity prices have attracted a lot of attention recently. For example, L. Yao et al. [19] developed a binary programming-based strategy to coordinate multiple EVs charging in a parking station in response to real-time curtailment request from the utility. G. Binetti et al. [20] offered a formulation for the coordinated charging problem which considered the plug-in and plug-off frequency. Then, a real-time greedy algorithm is designed to

This work was supported by the Office of Naval Research under award number N00014-18-1-2396. *(Corresponding author: Haibo He)*

Z. Wan and H. He are with the Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, RI 02881 USA (e-mail: zwan@ele.uri.edu; he@ele.uri.edu).

H. Li is with Lab. of Networked Control Systems, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016 China (e-mail: cn.h.li@ieee.org).

D. Prokhorov is with the Toyota Research Institute, North America, Ann Arbor, MI 48105 USA (e-mail: dvprokhorov@gmail.com).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2018.2879572, IEEE Transactions on Smart Grid

solve the formulated problem in a decentralized manner. Y. T. Liao et al. [21] employed a stochastic dynamic programming method which maximizes the operator's profit for the real-time dispatch of an EV charging station equipped with photovoltaic panels. Q. Huang et al. [22] presented a Markov Decision Process (MDP) formulation for the EV scheduling problem while considering the uncertainty and dynamics in wind energy supply. A rollout algorithm is used to derive the optimal scheduling policy. The above methods formulate this scheduling problem as a model-based control problem. These model-based approaches have obtained good results for this EV charging scheduling problem.

Recently, model-free approaches which do not need any system model information have achieved great success in complex decision-making application [23]. This success has inspired the development of model-free approaches for smart grid applications [24]–[28]. Compared to model-based approach, the advantage of the model-free approach is that it can learn a good control policy based on reinforcement learning (RL) and does not rely on any knowledge of the system [28]. An action-value function is proposed in these approaches to assess the quality of the charging schedule. The main difference of these approaches is how to approximate the optimal action-value function. For instance, Z. Wen et al. [24] use a Q-Table to estimate this function by discretizing the electricity price and charging actions. The limitation of this method is that it can only handle a small number of states and actions. In addition, the discretization step greatly influences the performance and should be properly determined. In order to overcome the downside of the Q-Table, a combination of linear basis function is implemented to approximate the action-value function in [25]. However, the linear approximator is incapable of handling the nonlinearity in the real-world electricity price and commuting behavior. In addition to this linear approximator, A. Chi et al. [26] apply a non-linear kernel averaging regression operator to fit the action-value function. The drawback of this approach is that the determination of the kernel function and its parameters greatly affect its performance. Overall, the limited approximation capability of the above approaches hinders their implementation in real-world scenarios.

Neural network has the potential of being universal approximator [29] and has been widely used for RL [30]–[32]. In recent years, deep neural network achieved promising results in learning complex mapping from high-dimensional data. By utilizing deep neural network, deep RL has obtained significant success in many complex decision-making applications. For instance, a deep Q-network has achieved a level comparable to that of a professional human in the Atari 2600 [23]. However, to the best of our knowledge, application of deep RL in real-time EV charging/discharging problem has not been reported in the literature.

In this paper, the EV charging/discharging scheduling problem is formulated as an MDP from the user's perspective. The objective is to find cost-efficient charging/discharging schedules to take full advantage of the real-time electricity price while fulfilling user's driving demand. A model-free approach is proposed to determine the optimal schedules in a real-world scenario based on the deep RL. The proposed approach uses the past electricity prices and battery SOC as inputs, and outputs real-time charging/discharging schedules. Unlike the traditional model-based methods, the proposed approach does not require any system model information. Numerous experimental results demonstrate the effectiveness of the proposed approach.

The contributions of this paper are threefold.

- An MDP with unknown transition probability is constructed from the user's perspective to formulate the EV charging/discharging scheduling problem. The randomness of the electricity price and the commuting behavior is taken into consideration to formulate the real-world scenarios.
- A deep RL based model-free approach which does not require any system model information is proposed to determine an optimal strategy for this real-time scheduling problem.
- A representation network is designed to extract the features from the electricity prices. After concatenating these features with battery SOC, the concatenated features are fed into a Q network to approximate the optimal action-value function.

The rest of the paper is organized as follows. The problem formulation is introduced in Section II. Then, a deep RL based approach is proposed in Section III to solve this problem. In Section IV, numerous experiments are presented to demonstrate the effectiveness of the proposed approach. Finally, Section V gives a conclusion.

II. Problem Formulation

We formulate the real-time EV charging/discharging scheduling problem from the user's perspective. When the EV is at home, we determine the charging/discharging action every hour. A finite MDP with discrete time step is applied to formulate this problem. Specifically, the time interval between two adjacent steps is one hour. At time step t, we observe the system state s_t which includes the information about the remaining energy in the EV battery and the past 24-hour electricity prices. Based on this information, we will choose the charging/discharging action a_t. This action represents the amount of energy that the EV battery will be charged or discharged during this time interval. After executing this action, we can observe the new system state s_{t+1} and choose the new charging/discharging action a_{t+1} for time step $t+1$.

MDP provides a mathematical architecture for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker. A MDP is a five-tuple $(\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{P}_{\cdot}(\cdot,\cdot), \boldsymbol{R}_{\cdot}(\cdot,\cdot), \gamma)$, where $\boldsymbol{S}$ is the system states, $\boldsymbol{A}$ is a finite set of actions, $\boldsymbol{P}_{\cdot}(\cdot,\cdot)$ is the state transition probability, $\boldsymbol{R}_{\cdot}(\cdot,\cdot)$ is the immediate reward, and γ is a discount factor. Considering the randomness of traffic conditions and driver's commuting behavior, the daily EV energy consumption, arrival time and departure time are supposed to change randomly. The details about the MDP formulation are shown as follow.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2018.2879572, IEEE Transactions on Smart Grid

1) State: The system state at time step t is defined as a vector $s_t = (u_t, E_t, P_{t-23}, \ldots, P_t)$. This vector encapsulates three types of information: (1) u_t indicates whether the EV is at home or not; (2) E_t represents the remaining energy in the EV battery; (3) $(P_{t-23}, \ldots, P_t)$ denotes the past 24-hour electricity prices.

2) Action: Given the state s_t, the action a_t represents the charging/discharging power. Let a_t be positive when the EV is charging and negative when discharging. The EV user can make money by charging the battery when the electricity price is low and discharging the battery when the price is on-peak. The charging and discharging power is constrained as below,

$$-e_{dis}^{max} \leq a_t \leq e_{ch}^{max}, \tag{1}$$

where e_{ch}^{max} and e_{dis}^{max} are the allowed maximum charging and discharging power of the EV battery, respectively. As suggested by [33]–[35], we assume that the EV charger provides discrete charging power, i.e., $a_t \in \{e^1, ..., e^L\}$.

3) State transition: The state transition from the state s_t to s_{t+1} is denoted as

$$s_{t+1} = f(s_t, a_t, \omega_t), \tag{2}$$

where the state transition is not only controlled by the action a_t but also influenced by the randomness ω_t. Specifically, the state transition for E_t is controlled by a_t and can be explicitly expressed by a deterministic battery model $E_{t+1} = E_t + a_t$. For u_t and P_t, the state transition is subject to randomness since the arrival time, departure time, and next-hour electricity price are unknown. Finding an accurate distribution model of the randomness ω_t can be difficult since it is influenced by many factors, such as traffic condition, user's commuting behavior, pricing process of the utility, etc. To solve this problem, a model-free approach is proposed to learn the state transition from real system data as shown in Section III.

4) Reward: The reward at time step t is defined from user's perspective as

$$r_t = \begin{cases} -P_t \cdot a_t - C^{a_t}, & t \neq t_\beta \\ -P_t \cdot a_t - C^{a_t} - \tau \cdot (E_{max} - E_t)^2, & t = t_\beta \end{cases} \tag{3}$$

where t_β is the time step when the EV leaves home.

In this reward, $P_t \cdot a_t$ represents the charging cost at time step t. When the EV is charging, this term is positive. On the contrary, if the EV discharges energy back to the power grid to make money, this term is negative. As suggested by [7], [11], the price of selling electricity to the grid is the same as that of purchasing electricity. This scenario is based on the net metering arrangement [36] under which a bi-directional meter is applied to record both the electricity purchased from the grid and the electricity fed back to the grid. Under this arrangement, selling electricity and purchasing electricity have the same price.

C^{a_t} denotes the cost of battery degradation for a charging/discharging of a_t kWh. According to [7], we assume that the battery degradation is only sensitive to the number of cycles, and the degradation cost is estimated as

$$C^{a_t} = C^E \left| \frac{m_k}{100} \right| \frac{a_t}{E_{max}} \tag{4}$$

where C^E denotes the total battery cost, m_k represents the slope of the linear approximation of the battery life as a function of the cycles, and E_{max} is the battery capacity.

$(E_{max} - E_t)^2$ measures user's "range anxiety" which is the fear that the EV has insufficient energy to reach its destination. This term is proposed to penalize the amount of uncharged battery energy, $E_{max} - E_t$. Larger uncharged energy will result in bigger penalty. In practice, some users are likely to tolerate large range anxiety to obtain low charging cost. In order to measure the user's preference towards the cost-saving objective and the range anxiety reducing objective, an anxiety coefficient τ is introduced. τ is measured in \$/kWh^2 such that the range anxiety term has the same measurement unit as the charging cost term.

5) Action-value function: The quality of a charging/discharging schedule a under a given system state s is assessed by the expected total sum of future rewards for the horizon of K time steps as follow

$$Q_{\boldsymbol{\pi}}(s, a) = \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{k=0}^{K} \gamma^k \cdot r_{t+k} \middle| s_t = s, a_t = a \right], \tag{5}$$

where $Q_{\boldsymbol{\pi}}(s, a)$ is called action-value function, $\boldsymbol{\pi}$ is the EV charging/discharging policy which maps from a system state to a charging/discharging schedule, and $0 < \gamma < 1$ is the discount factor, which balances the importance between the immediate reward and future rewards. For instance, when $\gamma = 1$, future reward is as important as the immediate reward and the policy is foresighted. When $\gamma = 0$, only immediate reward is taken into consideration and the policy is shortsighted.

The objective of this scheduling problem is to find an optimal $\boldsymbol{\pi}^*$ to maximize the action-value function as

$$Q^*(s, a) = \max_{\boldsymbol{\pi}} Q_{\boldsymbol{\pi}}(s, a), \tag{6}$$

where $Q^*(s, a)$ represents the optimal action-value function.

III. Proposed Approach

It is difficult to analytically determine the optimal policy $\boldsymbol{\pi}^*$ since the future electricity prices and user's commuting behavior are unknown. A reinforcement learning (RL) solution is to iteratively update the action-value function $Q(s, a)$ based on the Bellman equation [37]

$$Q_{i+1}(s, a) = \mathbb{E}\left[r_t + \gamma \max_{a_{t+1}} Q_i(s_{t+1}, a_{t+1}) \middle| s_t = s, a_t = a \right]. \tag{7}$$

As the number of iterations $i \to \infty$, $Q(s, a)$ will converge to the optimal action-value function $Q^*(s, a)$ [38]. Then, the optimal schedules are determined by a greedy strategy

$$a^* = \underset{a \in \mathcal{A}}{\operatorname{argmax}} Q^*(s, a). \tag{8}$$

$Q^*(s, a)$ is generally approximated by a look-up table [38]. However, since the electricity price in our problem is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2018.2879572, IEEE Transactions on Smart Grid

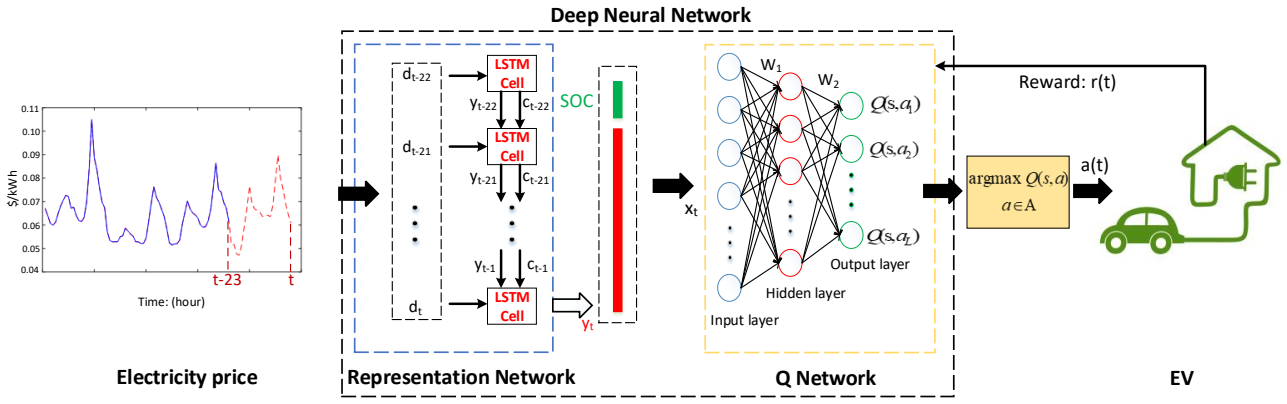


Fig. 1. The Overall diagram of the proposed approach for EV charging/discharging scheduling. The deep RL based approach uses the past 24-h electricity prices and battery SOC as inputs, calculates the reward $r(t)$, and outputs real-time charging/discharging schedules.

continuous and high-dimensional, an extremely large table is required to approximate $Q^*(s,a)$ and it is intractable to update such a large table. In this paper, a Deep Neural Network (DNN) is proposed to approximate $Q^*(s,a)$, and this approach is called deep reinforcement learning (RL).

The overall diagram of the proposed approach based on deep RL for EV charging/discharging scheduling is illustrated in Fig. 1. The representation network extracts discriminative features from the electricity price. After concatenating these features with battery SOC, the concatenated features are fed into a Q network to approximate the action-value of all feasible schedules under the given input state. The schedule with the largest action-value is selected as the EV charging/discharging schedule.

A. Architecture of Deep Neural Network

1) Representation Network: Extracting discriminative features from the raw data is a crucial step to improve the action-value function approximation. Good features should contain the information about the electricity price trends. With these features, the scheduling strategy can minimize the charging cost. In this paper, a representation network is proposed to extract these features.

Since electricity price fluctuates in a quasiperiodic way and has a natural temporal ordering, it is reasonable to infer future price trends from past electricity prices. Long Short-Term Memory (LSTM) network is known for its strong ability to model the time dependencies of time-series data [39], [40], and has achieved promising results in smart grid applications, such as load forecasting [41], [42]. In the representation network of Fig. 1, electricity price trends are captured by a LSTM network. Its input is the past 24-h electricity price and its output is the features containing information about future price trends.

The idea behind the LSTM network is to make use of sequential information, such as the real-time electricity prices. LSTM network performs the same processing for every element of the sequence, with the output being dependent on the previous computations. The information about what has been

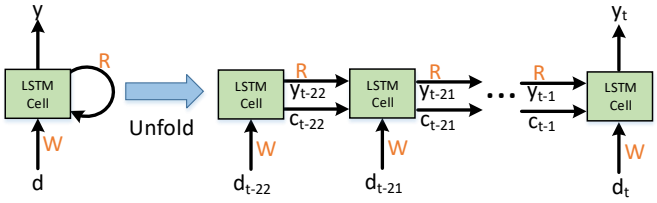


Fig. 2. An unfolded LSTM network.

calculated so far can be stored or "memorized" in the LSTM cells. The typical structure of an LSTM network is shown in Fig. 2.

Fig. 2 shows an LSTM network being unfolded into a full network. By unfolding, we simply mean that we write out the network for the complete sequence. For this EV charging scheduling problem, the LSTM network would be unfolded into a 23-layer neural network. Specifically, the input of the first layer is $d_{t-22} = P_{t-22} - P_{t-23}$ where P_{t-22} and P_{t-23} represent the electricity price at time step $t-22$ and $t-23$, respectively. W and R represent the corresponding parameters which are shared across all the layers. y_{t-22} denotes the output of the first layer, and c_{t-22} denotes its cell state. y_{t-22} and c_{t-22}, which contain the information of the past electricity price, are passed into the second layer. This process is repeated until the last layer.

The structure of the LSTM cell is shown in Fig. 3. The key to the LSTM network is the cell state c_t. The LSTM network has the ability to add information into or remove information from the cell state, carefully regulated by structures called gates. Gates are a way to optionally let information through. Specifically, an input gate determines the amount of information to be added into the cell state while a forget gate determines the amount of information to be inherited from the previous cell state c_{t-1}. The input gate and forget gate are shown in Eq. (9) and Eq. (10), respectively,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2018.2879572, IEEE Transactions on Smart Grid

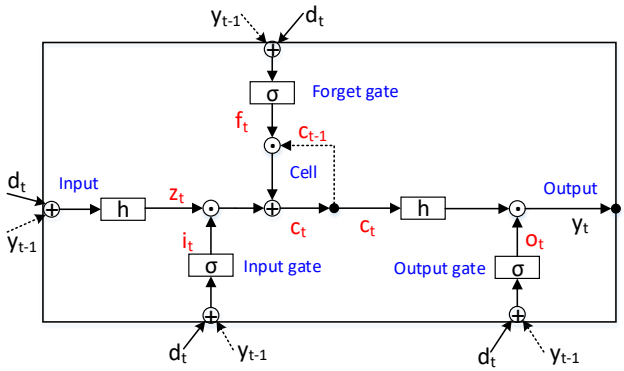


Fig. 3. The structure of the LSTM cell [43].

$$i_t = \sigma\left(W_i * d_t + R_i * y_{t-1} + b_i\right) \tag{9}$$

$$f_t = \sigma\left(W_f * d_t + R_f * y_{t-1} + b_f\right), \tag{10}$$

where W_i, R_i, W_f, and R_f are the matrices of weights for the input gate and forget gate; b_i, and b_f are the vectors of biases for these gates; σ is the sigmoid function that outputs numbers between 0 and 1, describing how much of information should be let through.

The input information z_t is shown as

$$z_t = h\left(W_z * d_t + R_z * y_{t-1} + b_z\right), \tag{11}$$

where h denotes the hyperbolic tangent function; W_z and R_z are the matrices of weights; b_z is the vector of biases. The input gate would determine the amount of z_t to be added into the cell. Therefore, the cell state is calculated as

$$c_t = i_t \odot z_t + f_t \odot c_{t-1}, \tag{12}$$

where $\odot$ represents the element-wise multiplication operation; $i_t \odot z_t$ denotes the amount of information to be added from z_t; $f_t \odot c_{t-1}$ denotes the amount of information to be inherited from the previous cell state c_{t-1}.

The output of the cell is determined by the output gate $o_t = \sigma\left(W_o * d_t + R_o * y_{t-1} + b_o\right)$ where W_o, R_o are the matrices of weights, and b_o is the vector of biases. Thus, the output of the cell is shown as

$$y_t = o_t \odot h\left(c_t\right). \tag{13}$$

For detailed information about LSTM network, readers can refer to [43].

The output of the LSTM network, y_t, is concatenated with the battery SOC which is a scalar. These concatenated features, x_t, contain information about both the future price trends and the battery SOC. The information of the future price trends is essential to reduce the charging cost, while the information of the battery SOC is important to ensure the EV can be well-charged. Then, these concatenated features are fed into the Q network to approximate the optimal action-value function.

2) Q Network: The Q network is a three-layer fully-connected neural network which can uniformly approximate continuous functions [29]. The input layer is fully-connected to a hidden layer with V units, and the value of the hidden unit is

$$v_t = g\left(W_1 * x_t + b_1\right), \tag{14}$$

where x_t denotes the input vector, g is the rectified linear activation function, W_1 is the matrix of weights, and b_1 is the vector of biases. Then, the hidden layer is fully-connected to the output layer. The output of the Q network is the action-value for all feasible charging/discharging schedules under system state s, i.e.,

$$Q\left(s, a\right) = g\left(W_2 * v_t + b_2\right), \tag{15}$$

where W_2 is the matrix of weights, and b_2 is the vector of biases. Then, the schedule with the largest action-value is outputted as the charging/discharging schedule.

B. Training of Deep Neural Network

Algorithm 1 shows how to train the DNN based on deep RL. The parameters of the DNN are denoted as $\theta = \{\theta_1, \theta_2\}$ where $\theta_1 = \{W_z, R_z, b_z, W_i, R_i, b_i, W_f, R_f, b_f, W_o, R_o, b_o\}$ represent the parameters of the representation network, and $\theta_2 = \{W_1, b_1, W_2, b_2\}$ represent the parameters of the Q network. The input of Algorithm 1 is the electricity prices, battery SOC, and reward r. Its output is the DNN's parameters θ that include the parameters of the representation network and the parameters of the Q network.

In line 1 of Algorithm 1, the DNN's parameters θ are randomly initialized. After that, in line 2, shadow parameters $\bar{\theta}$ are copied from θ. Then, the parameters θ are updated for M epochs in the outer loop. Each epoch starts at time step t_A which represents EV arrival time. Each epoch ends at time step t_D which represents EV departure time. The time horizon of each epoch equals to $t_D - t_A$. At the beginning of each epoch, we first obtain the initial state s_{t_A}. After that, in the inner loop starting from line 5, the EV charging/discharging is scheduled from time step t_A to t_D. At each time step, the charging/discharging schedule a_t is selected based on the ε-greedy search method, i.e., the schedule is randomly selected with probability ε, otherwise the proposed DNN is applied to choose the schedule. Then, we execute the selected schedule a_t, observe the reward r_t, and process to the new state s_{t+1}. After that, in line 8, the transition (s_t, a_t, r_t, s_{t+1}) is stored in a replay memory $\mathcal{D}$. The transitions in the replay memory are utilized to update the parameters θ. Specifically, in line 9, a minibatch of transitions $\mathcal{F} = \{(s_j, a_j, r_j, s_{j+1})\}_{j=1}^{\#\mathcal{F}}$ are randomly selected from the replay memory $\mathcal{D}$. With these transitions, the target action-value $\bar{q}_j$ is calculated as

$$\bar{q}_j = r_j + \gamma Q\left(s_{j+1}, \underset{a}{\operatorname{argmax}}\, Q\left(s_{j+1}, a; \theta_t\right); \bar{\theta}\right). \tag{16}$$

Then, in line 11, the loss function can be derived as

$$L\left(\theta_t\right) = \sum_{j=1}^{\#\mathcal{F}} \left[\bar{q}_j - Q\left(s_j, a_j; \theta_t\right)\right]^2, \tag{17}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2018.2879572, IEEE Transactions on Smart Grid

Algorithm 1 Training of Deep Neural Network

Input: Electricity prices, battery SOC, and reward r
Output: DNN's parameters θ

Randomly initialize DNN's parameters θ.
Initialize shadow parameters $\bar{\theta} = \theta$.
for Epoch=1:M **do**
  Obtain the initial state s_{t_A}.
  for Time step t=t_A:t_D **do**
    Select schedule a_t based on ε-greedy search.
    Execute schedule a_t, observe reward r_t, and process to the new state s_{t+1}.
    Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$.
    Sample minibatch of transitions $\mathcal{F} = \{(s_j, a_j, r_j, s_{j+1})\}_{j=1}^{\#\mathcal{F}}$ from $\mathcal{D}$.
    $\bar{q}_j \longleftarrow r_j + \gamma Q\left(s_{j+1}, \operatorname{argmax}_a Q\left(s_{j+1}, a; \theta_t\right); \bar{\theta}\right)$.
    Calculate the loss function $L\left(\theta_t\right) = \sum_{j=1}^{\#\mathcal{F}} \left[\bar{q}_j - Q\left(s_j, a_j; \theta_t\right)\right]^2$.
    Update parameters $\theta_{t+1} = \theta_t - \eta \bigtriangledown_{\theta_t} L\left(\theta_t\right)$.
    Every B steps reset $\bar{\theta} = \theta$.
  end for
end for

which is the error between the target action-value $\bar{q}_j$ and the action-value $Q\left(s_j, a_j; \theta_t\right)$ estimated by the DNN. Then, this loss function is minimized by updating the parameters θ according to gradient rule as

$$\theta_{t+1} = \theta_t - \eta \bigtriangledown_{\theta_t} L\left(\theta_t\right), \tag{18}$$

where η is the learning rate, and $\bigtriangledown_{\theta_t} L\left(\theta_t\right)$ represents the gradient of the loss function. After that, at every B steps, the shadow parameters are reset as $\bar{\theta} = \theta$. After the training process, the parameters θ will be outputted for the EV charging/discharging scheduling.

It is worth noting that using the reply memory has two benefits [44]. First, storing the transitions in the replay memory contributes to better data efficiency because each transition can be used multiple times to update the parameters. Second, the replay memory improves the stability of the training process. This is because most minibatch optimization algorithms have the assumption that the data is independent and identically distributed (i.i.d.) [45]. However, without a replay memory, the updates of parameters would be based on consecutive transitions that are highly correlated. Learning from consecutive transitions would violate this i.i.d. assumption and result in high variance of the updates of the parameters, which leads to slower and potentially less stable learning [46], [47]. Storing historical transitions in the replay memory and randomly sampling the transitions from the replay memory break the temporal correlations between the transitions and thus alleviate this problem.

C. Real-Time EV Charging/Discharging Scheduling

The DNN's parameters trained by Algorithm 1 will be fixed for the real-time EV charging/discharging scheduling. The scheduling algorithm is presented in Algorithm 2. Its input is the electricity prices and battery SOC. Its output is the EV charging/discharging schedules.

Algorithm 2 Real-Time EV Charging/Discharging Scheduling

Input: State s_t, including electricity prices and battery SOC.
Output: EV charging/discharging schedules $a_{t_A:t_D}$.

1: Load the DNN's parameters θ trained by Algorithm 1.
2: **for** Time step t=t_A:t_D **do**
3: Obtain past 24-h electricity prices.
4: Representation network extracts features from the electricity prices.
5: Concatenate these features with battery SOC.
6: Q network calculates action-value $Q\left(s_t, a; \theta\right)$.
7: $a_t = \operatorname{argmax}_{a \in \mathcal{A}} Q\left(s_t, a; \theta\right)$
8: Output EV charging/discharging schedule a_t.
9: **end for**
10: **Return:** $a_{t_A:t_D}$

In Algorithm 2, we first load the parameters of DNN trained by Algorithm 1. In the loop starting from line 2, the DNN is implemented to generate the EV charging/discharging schedules from time step t_A to t_D. At each time step, the representation network is applied to extract features from the past 24-h electricity prices. Then, these features are concatenated with battery SOC. After that, Q network calculates action-value $Q\left(s_t, a; \theta\right)$ based on these concatenated features. Then, the EV charging/discharging schedule a_t is selected in line 7 as $a_t = \operatorname{argmax}_{a \in \mathcal{A}} Q\left(s_t, a; \theta\right)$. Finally, the schedule a_t is outputted.

IV. Experimental Results

In this section, we evaluate the proposed approach on multiple case studies and demonstrate its effectiveness by simulation analysis. The details about the experimental setup are presented in Section IV-A. We present the experimental results of case study I in Section IV-B where the battery degradation cost is ignored. Then, the effect of this degradation cost is analyzed in case study II in Section IV-C. Finally, we explain how to choose the neural network structures and the hyperparameters in Section IV-D.

A. Experimental Setup

The performance of the proposed approach is evaluated under a real-world scenario. The real-world hourly electricity price starting from 1st of February 2014 and lasting 360 days is downloaded from the California ISO [48]. This price data is separated into training and testing data. In each 30 consecutive days, the first 20 days are selected for training, and the remaining 10 days are used for performance evaluation. As suggested by [19], user's commuting behavior is modeled as random variables. Specifically, the EV arrival time, the departure time, and the battery SOC at the arrival time obey truncated normal distributions. The distributions are presented in Table. I. The arrival time t_A is sampled from $\mathcal{N}\left(18, 1^2\right)$ and is bounded between 15 and 21. For the departure time t_D, its distribution $\mathcal{N}\left(8, 1^2\right)$ is bounded between 6 and 11. The battery SOC bounded between 0.2 and 0.8 is sampled

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2018.2879572, IEEE Transactions on Smart Grid

TABLE I
RANDOM VARIABLES FOR COMMUTING BEHAVIOR.

	Distribution	Boundary
Arrival time	$t_A \sim \mathcal{N}\left(18, 1^2\right)$	$15 \leq t_A \leq 21$
Departure time	$t_D \sim \mathcal{N}\left(8, 1^2\right)$	$6 \leq t_D \leq 11$
Battery SOC	$SOC \sim \mathcal{N}\left(0.5, 0.1^2\right)$	$0.2 \leq SOC \leq 0.8$

from $\mathcal{N}\left(0.5, 0.1^2\right)$. It is worth noting that our learning-based approach does not rely on any knowledge of the distributions of these random variables. Thus, our approach can scale to different modeling mechanisms. We assume that a Nissan Leaf with battery capacity $E_{max} = 24$ kWh is used in our experiments. When the EV is at home, the charger provides 7 levels (6 kW, 4 kW, 2 kW, 0 kW, -2 kW, -4 kW, -6 kW) for EV charging and discharging. The positive values represent the charging process while the negative ones refer to EV discharging.

The deep RL based model-free approach is proposed to generate the optimal charging/discharging schedules. The discounted factor γ is set to 0.99 so that the proposed approach can obtain a foresighted strategy. In the representation network, a 128-dimension feature vector is extracted from the past 24-h electricity price. Then, this vector is concatenated with the SOC and fed into the input layer of the Q network. The number of the units in its hidden layer and output layer are 64 and 7, respectively. The parameters of the representation network and the Q network are randomly initialized and updated by gradient descent during the training process. The batch size of the sampled transitions $\mathcal{F}$ for training is 32. The value of B is set as 500. The number of training epochs, M, equals to 50,000. The training process takes about 1.5h on the computer with one NVIDIA TITAN Xp GPU and one i7-6800K CPU. After the training process, the proposed approach can be deployed for the EV charging/discharging scheduling. It takes about 3.7 ms to generate one schedule. The code is written in Python with TensorFlow, a DNN package developed by Google Brain.

B. Case Study I

In this case study, the proposed model is trained to generate EV charging/discharging schedules when the battery degradation cost is ignored. The training process is presented in Section IV-B1. Then, the proposed approach is evaluated and compared with several benchmark methods in Section IV-B2. Finally, the effect of user's range anxiety on the cost-saving objective is discussed in Section IV-B3.

1) Training Process: The proposed DNN is trained for 50,000 epochs to learn the optimal EV charging/discharging scheduling. The anxiety coefficient τ is set to 0.01. The training process takes about 1.5 hours on the workstation mentioned in Section IV-A. Each epoch starts when the EV arrives home and ends when it leaves home. In each epoch, we calculate the cumulative rewards $\sum_{t=t_A}^{t_D} r_t$. The evolution of the cumulative rewards over 50,000 epochs is illustrated in Fig. 4. In the first 7,000 epochs, the charging/discharging schedule is randomly selected from the 7 feasible levels. Then, from epoch 7,000 to epoch 14,000, the schedule is randomly selected with probability ε, otherwise, it is chosen by the proposed DNN. In this phase, the probability ε is reduced from 1.0 to 0.1, and remains 0.1 afterward. It can be observed from Fig. 4 that the cumulative rewards start to increase gradually after epoch 18,000. Then, at epoch 35,000, the cumulative rewards converge around -0.2 \$ with small oscillations. This result demonstrates that the proposed approach succeeds in learning a policy to maximize the cumulative rewards.

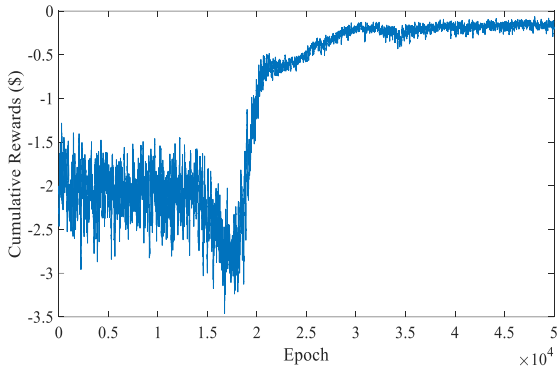


Fig. 4. The evolution of the cumulative rewards during the training process.

2) Performance Evaluation: In this section, the proposed approach is evaluated and compared with several benchmark solutions, including model predictive control (MPC), day-ahead, fitted-Q iteration (FQI), and uncontrolled strategy. For the MPC solution [49], the EV arrival time and depart time, and the battery SOC are known in advance. We evaluate MPC with two different forecasting models. For the first forecasting model, a fully-connected neural network (NN) with 24-20-20-8 units is used to provide the dynamic prediction of the future electricity prices over a rolling horizon of 8 hours. Based on the forecasted prices, a scheduling strategy over the rolling horizon is derived and only the first hour's schedule is implemented. At the next hour, the above procedure is repeated. Similarly, for the second forecasting model, an LSTM network is used to forecast the future electricity prices. This LSTM network is the same as the one in our representation network. We also evaluate MPC under an **ideal** situation where the future electricity prices are all known in advance. For the day-ahead solution, we assume the EV arrival time and depart time, and the battery SOC are known in advance. Autoregressive (AR) model is applied to forecast the electricity prices, and its order is 24, i.e. AR (p=24). Then, the EV charging/discharging scheduling problem is solved using YALMIP. For the FQI solution [50], a decision tree is used to approximate the action-value function. Then, the charging/discharging schedule is determined based on the greedy strategy in Eq. (8). For the uncontrolled solution, the EV is charged immediately with the maximum charging rate when it arrives home.

The 120 test days are used for performance evaluation. In each day, the charging cost is calculated as $\sum_{t=t_A}^{t_D} P_t \cdot a_t$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2018.2879572, IEEE Transactions on Smart Grid

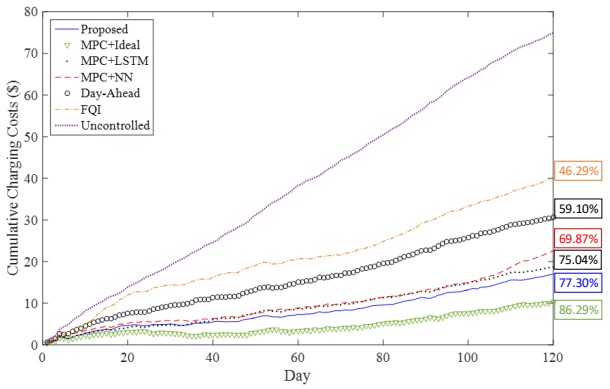


Fig. 5. Cumulative charging cost of the proposed and benchmark solutions over the 120 test days. The percentage terms on the right illustrate the cost reduction of the corresponding solutions with respect to the uncontrolled solution.

The cumulative charging costs of the proposed and benchmark solutions over the test days are presented in Fig. 5. The percentage terms on the right illustrate the cost reduction of the corresponding solutions with respect to the uncontrolled solution. We can observe that the proposed approach (blue solid line) reduces the charging cost by 77.30% in comparison with the uncontrolled solution (purple dotted line). The other four benchmark solutions, FQI (orange dash-dotted line), day-ahead (black circles), MPC+NN (red dash line), and MPC+LSTM (black dotted line) only reduce the cost by 46.29%, 59.10%, 69.87%, and 75.04%, respectively. These results demonstrate the effectiveness of the proposed approach for this EV charging/discharging scheduling problem. We also notice that MPC with real future price (green triangles) reduces the charging cost by 86.29%. Comparing the results of MPC, we can find that the performance of MPC relies on the accuracy of a forecasting model. However, our proposed approach does not need a forecasting model.

To further investigate the performance of the proposed approach, the charging/discharging patterns over 7 consecutive days are shown in Fig. 6. The green regions in each subfigure indicate the periods of time when the EV is not at home. In Fig. 6a, the hourly electricity price ($/kWh) is illustrated with the red line, while the hourly charging/discharging energy (kWh) is represented by the blue bar. One can observe that the proposed approach learns to charge when the electricity price is low and to discharge when the price is on-peak. These charging/discharging patterns verify the proposed approach's capability to forecast the electricity price trends. The remaining battery energy at each hour is presented in Fig. 6b. The battery is well-charged when the EV leaves home. These results demonstrate that the proposed approach can reduce the charging cost as well as satisfy user's driving demand.

3) Discussion of Range Anxiety: In the real-world scenario, different users may have a different preference towards the cost-saving objective and the range anxiety reducing objective. The effect of the anxiety coefficient τ on these two objectives is demonstrated in Fig. 7 where the coefficient increases from 0 to 0.01 with a step of 0.001. In Fig. 7, the range anxiety and charging cost are averaged over the 120 test days. The blue line with circle marks represents the average range anxiety, while the red line with square marks is the curve for average charging cost. The negative value of the charging cost indicates that the EV earns money by selling electricity to the utility. We can observe that large anxiety coefficient leads to small range anxiety but large charging cost. This figure demonstrates that the coefficient can balance the trade-off between the range anxiety and the charging cost. Specifically, if the user only needs to drive for a short distance, the coefficient can be set to a small value. Consequently, the proposed approach tries to reduce the charging cost. On the contrary, if the user wants to reduce the range anxiety, this coefficient can be set to a high value.

In order to further demonstrate the effect of the anxiety coefficient τ, the daily charging cost and the battery SOC when the EV leaves home are presented in Fig. 8. This figure shows the results over 120 test days with two different τ. When τ is set to 0.002, the proposed approach can achieve a lower charging cost, but the battery is about half charged. On the contrary, when τ is set to 0.008, the battery is almost fully charged, but the charging cost increases. These results demonstrate that the proposed approach can adaptively adjust to user's different preferences by setting different anxiety coefficient.

C. Case Study II

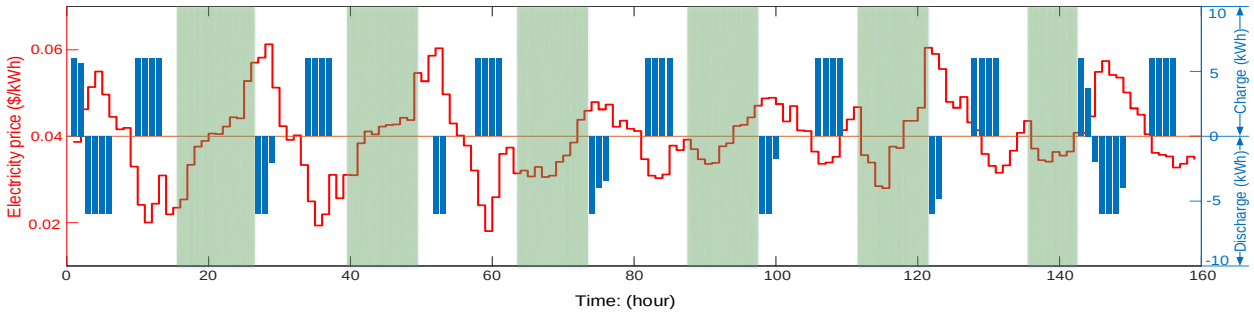
In this case study, we conduct extensive experiments to analyze the effect of the battery cost on the EV charging/discharging scheduling. In the experiments, the battery cost is set to $[400, 300, 200, 100, 80, 60, 40, 20]$ $/kWh, respectively. The slope $m_k = -0.015$.

Fig. 9 shows the cumulative charging cost, vehicle-to-grid (V2G) cost, and battery degradation cost over the 120 test days under different battery costs. The blue line with square marks illustrates the cost associated with charging the EV. The red line with triangle marks shows the V2G cost where the negative values indicate the EV discharges energy back to the grid and thereby make money. The black line with star marks shows the battery degradation cost. We can observe that the proposed model determines to discharge energy back to the grid to make money when the battery cost is less than 100 $/kWh. The smaller the battery cost is, the more energy the EV discharges back to the grid. As the battery cost increases up to 100 $/kWh, the V2G stops being attractive and the proposed model determines to only charge the EV. In conclusion, the battery degradation cost plays an important role in determining the EV charging/discharging schedules, and the proposed model can well adjust to the scenarios with different battery degradation costs.

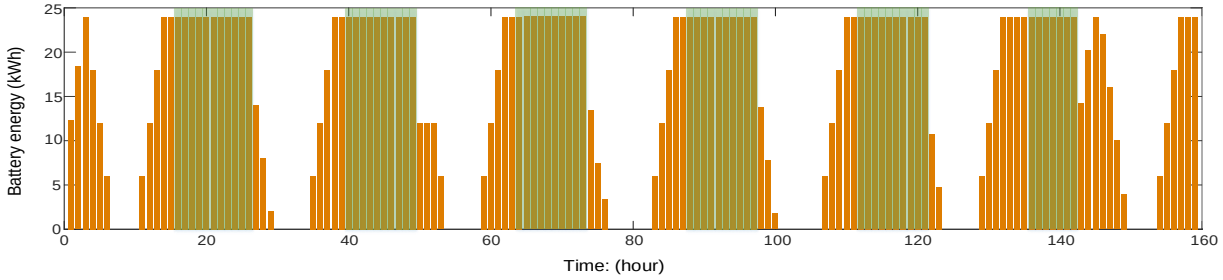
D. Discussion of neural network structures and hyperparameters

The reason we choose LSTM network to extract features from the electricity prices is that it has strong ability to model the time dependencies of time-series data [39], [40] and has

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2018.2879572, IEEE Transactions on Smart Grid



(a) Hourly electricity price (red line) and charging/discharging energy (blue bar).



(b) Remaining battery energy

Fig. 6. Charging/discharging patterns of the proposed approach over 7 consecutive days: (a) Hourly electricity price (red line) and charging/discharging energy (blue bar); (b) Remaining battery energy. The green regions indicate that the EV is not at home.

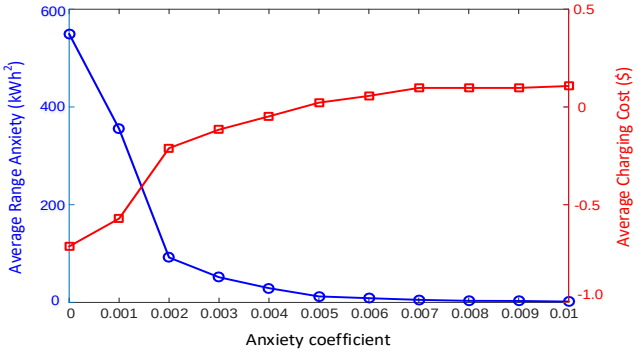
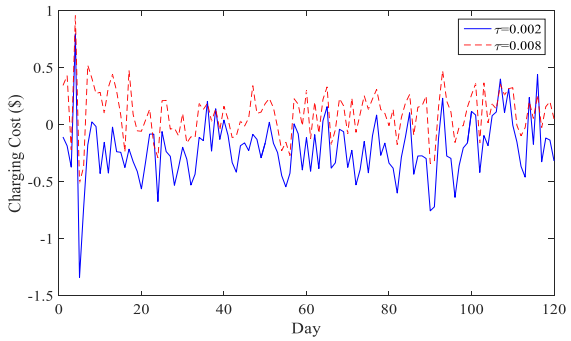


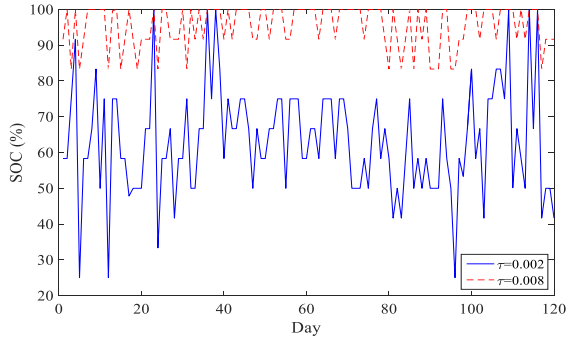
Fig. 7. Average range anxiety (blue circle) and charging cost (red square) under different anxiety coefficient.

achieved promising results in smart grid applications [41], [42]. We further clarify the importance of the LSTM network by conducting an experiment where the LSTM network is removed. In other words, the electricity prices are directly concatenated with the battery SOC. Then, these concatenated features are fed into the Q network. Without the LSTM network, the accumulated charging cost over the 120 test days is 27.71 $. However, the accumulated charging cost of our original model with the LSTM network is 17.02 $. The LSTM network contributes to a 38.59% reduction in the charging cost.

The reason we choose a three-layer fully-connected neural network for the Q network is that it can uniformly approximate continuous functions [29], and its effectiveness for deep reinforcement learning (RL) has been validated in [23]. In the



(a) Charging cost



(b) Battery SOC

Fig. 8. Daily charging cost and battery SOC when the EV leaves home under different anxiety coefficient τ.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2018.2879572, IEEE Transactions on Smart Grid

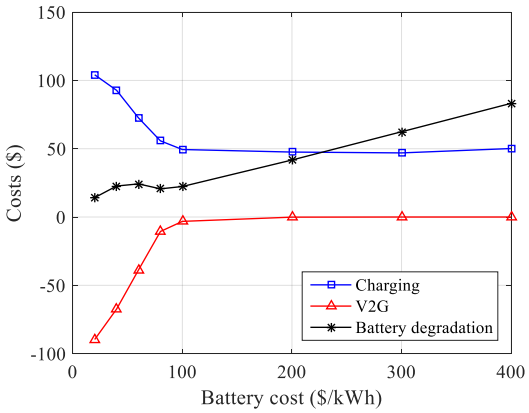


Fig. 9. Cumulative charging cost, vehicle-to-grid (V2G) cost, and battery degradation cost over the 120 test days under different battery cost.

fully-connected neural network, the number of the input units is 129, which equals to the dimension of its input. The number of the output units is 7, which equals to the number of feasible actions for the EV charging scheduling. For the number of hidden units, there is an empirically-derived rule-of-thumb, i.e., the optimal number of the hidden units is usually between the number of input units and the number of the output units. In our original model, the number of hidden units is 64, and the accumulated charging cost over the 120 test days is 17.02 $. We also conduct experiments with another two numbers of hidden units, 32 and 128. Their accumulated charging cost are 17.29 $ and 17.13 $, respectively. We can see that the performance is insensitive to the number of hidden units when this number is chosen between the number of input units and the number of the output units.

The discounted factor γ balances the importance between the immediate reward and future rewards. When $\gamma = 0$, the learned policy is "myopic" in being concerned only with maximizing immediate reward. As γ approaches 1, the future rewards are taken into consideration more strongly, and the learned policy becomes more farsighted. In our original model, $\gamma = 0.99$, which is the same as that used in [23]. For our original model, the accumulated charging cost over the 120 test days is 17.02 $. We also conduct experiments with another two γ values, 0.5 and 0.8. Their accumulated charging cost are 36.92 $ and 24.65 $, respectively. We can observe that the performance is sensitive to the discounted factor γ. γ should be close to 1 such that the learned policy is farsighted.

Since GPU is used to train the neural network, the batch size is generally determined by the following criteria [51]:

- It is common for power of 2 batch size to offer better runtime. Typical power of 2 batch sizes range from 32 to 256.
- Since all examples in the batch will be processed in parallel in the GPU, the amount of memory usage will scale with the batch size. For many hardware setups, this is the limiting factor in batch size.

Bengio states that batch size $= 32$ is a good default value [52]. In our model, the batch size is set as 32 accordingly.

During the training process, the number of training epochs can be determined by early stopping [52]. Specifically, we set the number of epochs to a large number and stop the training when the cumulative rewards does not improve.

In our proposed approach, the probability ε is annealed linearly from 1.0 to 0.1, and fixed at 0.1 thereafter. This kind of design is consistent with that in [23] and is commonly used by reinforcement learning researchers to balance the trade-off between exploration and exploitation [38].

V. Conclusion

In this paper, we formulate the EV charging/discharging scheduling problem as a MDP with unknown transition probability from the user's perspective. In the problem formulation, the randomness of both the electricity price and the commuting behavior are considered. We propose a deep RL based approach to determine an optimal strategy for this real-time scheduling problem. The proposed approach is a model-free approach which does not need any system model information. In the proposed approach, a representation network is applied to extract features from the electricity prices. Then, a Q network is implemented to approximate the optimal action-value function based on these features. Finally, the action that maximizes this function is selected for the scheduling problem. The parameters of these two networks are updated by gradient descent during the training process. Experimental results show that the proposed approach outperforms various benchmark solutions. In addition, the proposed approach can adaptively adjust to different users' preferences towards the cost-saving objective and the range anxiety reducing objective.

References

[1] A. S. A. Awad, M. F. Shaaban, T. H. M. EL-Fouly, E. F. El-Saadany, and M. M. A. Salama, "Optimal resource allocation and charging prices for benefit maximization in smart pev-parking lots," *IEEE Transactions on Sustainable Energy*, vol. 8, no. 3, pp. 906–915, July 2017.

[2] D. Dallinger, J. Link, and M. Bttner, "Smart grid agent: Plug-in electric vehicle," *IEEE Transactions on Sustainable Energy*, vol. 5, no. 3, pp. 710–717, July 2014.

[3] The egallon: How much cheaper is it to drive on electricity? [Online]. Available: https://energy.gov/articles/egallon-how-much-cheaper-it-drive-electricity

[4] 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 Transactions on Smart Grid*, vol. 6, no. 6, pp. 2714–2724, Nov 2015.

[5] Maigha and M. L. Crow, "Cost-constrained dynamic optimal electric vehicle charging," *IEEE Transactions on Sustainable Energy*, vol. 8, no. 2, pp. 716–724, April 2017.

[6] J. R. Pillai and B. Bak-Jensen, "Integration of vehicle-to-grid in the western danish power system," *IEEE Transactions on Sustainable Energy*, vol. 2, no. 1, pp. 12–19, Jan 2011.

[7] 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 Generation, Transmission Distribution*, vol. 8, no. 6, pp. 1007–1016, June 2014.

[8] J. Zhao, C. Wan, Z. Xu, and J. Wang, "Risk-based day-ahead scheduling of electric vehicle aggregator using information gap decision theory," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1609–1618, July 2017.

[9] M. G. Vayá and G. Andersson, "Optimal bidding strategy of a plug-in electric vehicle aggregator in day-ahead electricity markets under uncertainty," *IEEE Transactions on Power Systems*, vol. 30, no. 5, pp. 2375–2385, Sept 2015.

[10] ——, "Self scheduling of plug-in electric vehicle aggregator to provide balancing services for wind power," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 2, pp. 886–899, April 2016.

[11] M. R. Sarker, H. Pandi, and M. A. Ortega-Vazquez, "Optimal operation and services scheduling for an electric vehicle battery swapping station," *IEEE Transactions on Power Systems*, vol. 30, no. 2, pp. 901–910, March 2015.

[12] D. Wu, H. Zeng, C. Lu, and B. Boulet, "Two-stage energy management for office buildings with workplace ev charging and renewable energy," *IEEE Transactions on Transportation Electrification*, vol. 3, no. 1, pp. 225–237, March 2017.

[13] Y. Guo, J. Xiong, S. Xu, and W. Su, "Two-stage economic operation of microgrid-like electric vehicle parking deck," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1703–1712, May 2016.

[14] I. Momber, A. Siddiqui, T. G. S. Romn, and L. Sder, "Risk averse scheduling by a pev aggregator under uncertainty," *IEEE Transactions on Power Systems*, vol. 30, no. 2, pp. 882–891, March 2015.

[15] S. I. Vagropoulos and A. G. Bakirtzis, "Optimal bidding strategy for electric vehicle aggregators in electricity markets," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4031–4041, Nov 2013.

[16] 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 Transactions on Sustainable Energy*, vol. 7, no. 1, pp. 374–385, Jan 2016.

[17] N. G. Paterakis, O. Erdin, A. G. Bakirtzis, and J. P. S. Catalo, "Optimal household appliances scheduling under day-ahead pricing and load-shaping demand response strategies," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 6, pp. 1509–1519, Dec 2015.

[18] O. Erdinc, N. G. Paterakis, T. D. P. Mendes, A. G. Bakirtzis, and J. P. S. Catalo, "Smart household operation considering bi-directional ev and ess utilization by real-time pricing-based dr," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1281–1291, May 2015.

[19] L. Yao, W. H. Lim, and T. S. Tsai, "A real-time charging scheme for demand response in electric vehicle parking station," *IEEE Transactions on Smart Grid*, vol. 8, no. 1, pp. 52–62, Jan 2017.

[20] G. Binetti, A. Davoudi, D. Naso, B. Turchiano, and F. L. Lewis, "Scalable real-time electric vehicles charging with discrete charging rates," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2211–2220, Sept 2015.

[21] Y. T. Liao and C. N. Lu, "Dispatch of ev charging station energy resources for sustainable mobility," *IEEE Transactions on Transportation Electrification*, vol. 1, no. 1, pp. 86–93, June 2015.

[22] Q. Huang, Q. S. Jia, Z. Qiu, X. Guan, and G. Deconinck, "Matching ev charging load with uncertain wind: A simulation-based policy improvement approach," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1425–1433, May 2015.

[23] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[24] Z. Wen, D. ONeill, and H. Maei, "Optimal demand response using device-based reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2312–2324, Sept 2015.

[25] 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 Transactions on Smart Grid*, vol. 6, no. 4, pp. 1795–1805, July 2015.

[26] A. Chi, J. Lundn, and V. Koivunen, "Reinforcement learning-based plug-in electric vehicle charging with forecasted price," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 3674–3684, May 2017.

[27] S. Bahrami, V. W. S. Wong, and J. Huang, "An online learning algorithm for demand response in smart grid," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2017.

[28] F. Ruelens, B. J. Claessens, S. Vandael, B. D. Schutter, R. Babuska, and R. Belmans, "Residential demand response of thermostatically controlled loads using batch reinforcement learning," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–11, 2017.

[29] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems (MCSS)*, vol. 2, no. 4, pp. 303–314, 1989.

[30] W. T. Miller, R. S. Sutton, and P. J. Werbos, *A Menu of Designs for Reinforcement Learning Over Time*. MIT Press, 1995, pp. 67–95.

[31] P. Werbos, J. Si, A. Barto, W. Powell, and D. Wunsch, "ADP: Goals, opportunities and principles," *Handbook of learning and approximate dynamic programming*, pp. 3–44, 2004.

[32] H. He, Z. Ni, and J. Fu, "A three-network architecture for on-line learning and optimization based on adaptive dynamic programming," *Neurocomputing*, vol. 78, no. 1, pp. 3–13, 2012.

[33] B. Sun, Z. Huang, X. Tan, and D. H. K. Tsang, "Optimal scheduling for electric vehicle charging with discrete charging levels in distribution grid," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 624–634, March 2018.

[34] C. L. Floch, E. C. Kara, and S. Moura, "Pde modeling and control of electric vehicle fleets for ancillary services: A discrete charging case," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 573–581, March 2018.

[35] G. Binetti, A. Davoudi, D. Naso, B. Turchiano, and F. L. Lewis, "Scalable real-time electric vehicles charging with discrete charging rates," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2211–2220, Sept 2015.

[36] Grid-connected renewable energy systems. [Online]. Available: https://energy.gov/energysaver/grid-connected-renewable-energy-systems

[37] R. Bellman, *Dynamic programming*. Princeton Univeristy Press, John Wiley & Sons, 1958.

[38] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[39] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct 1990.

[40] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[41] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on lstm recurrent neural network," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2017.

[42] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, "Short-term residential load forecasting based on resident behaviour learning," *IEEE Transactions on Power Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[43] K. Greff, R. K. Srivastava, J. Koutnk, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, Oct 2017.

[44] T. de Bruin, J. Kober, K. Tuyls, and R. Babuška, "The importance of experience replay database composition in deep reinforcement learning," in *Deep Reinforcement Learning Workshop, NIPS*, 2015.

[45] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[46] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural networks: Tricks of the trade*. Springer Berlin Heidelberg, 2012, pp. 437–478.

[47] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*. Springer Berlin Heidelberg, 2012, pp. 9–48.

[48] California independent system operator. [Online]. Available: http://oasis.caiso.com/mrioasis/logon.do

[49] P. Kou, D. Liang, L. Gao, and F. Gao, "Stochastic coordination of plug-in electric vehicles and wind turbines in microgrid: A model predictive control approach," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1537–1551, May 2016.

[50] D. Ernst, P. Geurts, and L. Wehenkel, "Tree-based batch mode reinforcement learning," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 503–556, 2005.

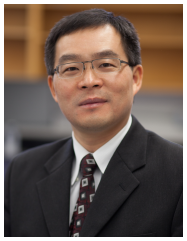
[51] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[52] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 437–478.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2018.2879572, IEEE Transactions on Smart Grid

Zhiqiang Wan (S'16) received his B.S. degree from the Harbin Institute of Technology, Harbin, China, in 2012. He received his M.S. degree in the School of Electrical and Electronics Engineering, Huazhong University of Science and Technology (HUST), Wuhan, China, in 2015. He is presently working towards his Ph.D. degree in the School of Electrical, Computer and Biomedical Engineering, University of Rhode Island (URI), Kingston, Rhode Island, USA. His current research interests include deep learning, smart grid, and deep reinforcement learning.

Hepeng Li (M'15) received his B.S. degree in information and computing science and the M.S. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2009 and 2012, respectively. He is an assistant professor in Institute of Automation, Chinese Academy of Sciences, Shenyang, China. He worked as a visiting scholar at University of Rhode Island in the U.S. from 2017 to 2018. His research interests include smart gird, microgrids, convex optimization, neural networks and deep learning.



Haibo He (SM'11-F'18) received the B.S. and M.S. degrees in electrical engineering from the Huazhong University of Science and Technology, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from Ohio University in 2006. He is currently the Robert Haas Endowed Chair Professor at the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island. His current research interests include computational intelligence, machine learning, data mining, and various applications. He has published one sole-author research book (Wiley), edited one book (Wiley-IEEE) and six conference proceedings (Springer), and authored and co-authored more than 300 peer-reviewed journal and conference papers. He was the general chair of the IEEE Symposium Series on Computational Intelligence (SSCI 2014). He received the IEEE International Conference on Communications Best Paper Award (2014), IEEE CIS Outstanding Early Career Award (2014), and National Science Foundation CAREER Award (2011). He is currently the Editor-in-Chief of the IEEE Transactions on Neural Networks and Learning Systems.

Danil Prokhorov (SM'02) started his research career in Russia. He studied system engineering which included courses in math, physics, mechatronics and computer technologies, as well as aerospace and robotics. He received his M.S. with Honors in St. Petersburg, Russia, in 1992. After receiving Ph.D. in 1997, he joined the staff of Ford Scientific Research Laboratory, Dearborn, Michigan. While at Ford he pursued machine learning research focusing on neural networks with applications to system modeling, powertrain control, diagnostics and optimization. He has been involved in research and planning for various intelligent technologies, such as highly automated vehicles, AI and other futuristic systems at Toyota Tech Center (TTC), Ann Arbor, MI since 2005. Since 2011 he is in charge of future research department in Toyota Motor North America R & D. He has been serving as a panel expert for NSF, DOE, ARPA, Senior and Associate Editor of several scientific journals for over 20 years. He has been involved with several professional societies including IEEE Intelligent Transportation Systems (ITS) and IEEE Computational Intelligence (CI), as well as International Neural Network Society (INNS) as its former Board member, President and recently elected Fellow. He has authored lots of publications and patents. Having shown feasibility of autonomous driving and personal flying mobility, his department continues research of complex multi-disciplinary problems while exploring opportunities for the next big thing.