Deep Reinforcement Learning Based Energy Storage Arbitrage With Accurate Lithium-ion Battery Degradation Model

Jun Cao, *Member, IEEE,* Dan Harrold, Zhong Fan, *Senior Member, IEEE,*
Thomas Morstyn, *Member, IEEE,* David Healey, and Kang Li

***Abstract*—Accurate estimation of battery degradation cost is one of the main barriers for battery participating on the energy arbitrage market. This paper addresses this problem by using a model-free deep reinforcement learning (DRL) method to optimize the battery energy arbitrage considering an accurate battery degradation model. Firstly, the control problem is formulated as a Markov Decision Process (MDP). Then a noisy network based deep reinforcement learning approach is proposed to learn an optimized control policy for storage charging/discharging strategy. To address the uncertainty of electricity price, a hybrid Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) model is adopted to predict the price for the next day. Finally, the proposed approach is tested on the the historical UK wholesale electricity market prices. The results compared with model based Mixed Integer Linear Programming (MILP) have demonstrated the effectiveness and performance of the proposed framework.**

***Index Terms*—Energy storage, Energy arbitrage, Battery degradation, Deep reinforcement learning, Noisy Networks**

I. Introduction

ENERGY storage systems can improve the flexibility of the power systems by providing various ancillary services to system operators, e.g. load shifting, frequency regulation, voltage support and grid stabilization [1]. Among these, energy arbitrage represents the largest profit opportunity for battery storage. In electricity markets, the storage can take advantage of the daily energy price fluctuations to buy the cheapest energy available during the period of low demand and sell it at the highest price in order to generate profits using energy arbitrage.

Extensive research has been conducted on the optimisation of energy storage arbitrage problem to maximise revenue. In [2] and [3], a mixed integer linear approach was developed to optimise the storage dispatch that can maximise the profits in real-time markets in the United States and Germany, respectively. In order to handle the uncertainty in electricity price, a scenario-based stochastic formulation was developed in [4] for battery energy arbitrage in both day-ahead and real-time market. The authors of [5] present a bidding mechanism based on two stage stochastic programming for a group of storage that participate in the day-ahead reserve market. Apart from the above stochastic optimization approaches, robust optimization is also widely used to handle uncertainty. In [6], a robust optimization based bidding strategy has shown an increasing probability of yielding better economic performance than a deterministic optimization based bidding strategy, when the forecast error in electricity price increases. In [7], an affinely adjustable robust bidding strategy for a solar power with a battery storage system was proposed to address the uncertainties of both PV solar power productions and electricity prices. However, the research in [2]-[7] did not consider a detailed model of battery degradation during the energy arbitrage process.

Battery degradation model is the key factor to energy arbitrage problem. Accurate calculation of degradation costs is crucial for obtaining realistic estimates of profitability. There is a growing literature examining the impact of battery degradation on energy arbitrage revenue [8], [9]. The impact of battery degradation on energy arbitrage revenue is studied in [8] and a novel battery operational cost model considering degradation cost based on depth of charge and discharge rate is developed in [9]. However, the degradation model used in [8], [9] is quite simplistic, which is not realistic to account for the degradation costs for energy arbitrage. There are already some independent research works on battery degradation model using either model based or data driven methods [10], [11], which can provide a precise degradation costs for different charging profiles. One of the main barriers of embedding this accurate model to energy arbitrage problem is that the calculation of degradation process is quite complicated and it is not straightforward to find a simple mathematical degradation model that can be included into the model-based energy arbitrage algorithm.

Recently, data-driven model-free approaches have made great progress in decision-making problems [12]. Many studies have focused on the application of Reinforcement Learning (RL), a model-free agent based AI algorithm, for smart grid, especially demand response. The authors of [13] present a comprehensive review on RL for demand response. The authors of [14] proposed a deep reinforcement learning based

This work is partly supported by the SEND project (Grant REF. 32R16P00706) funded by ERDF and BEIS, the Royal Society Research Grant (REF. RGS/R1/191395) and EnergyREV(EP/S031863/1).

J. Cao is with the School of Geography, Geology and the Environment, Keele University, UK ST5 5BG (corresponding author: jcao01@qub.ac.uk). Dan Harrold and Prof. Z. Fan are with the School of Computing and Mathematics, Keele University, UK ST5 5BG (email: z.fan@keele.ac.uk).

T. Morstyn is with the Department of Engineering Science, University of Oxford, Oxford OX1 2JD, U.K (thomas.morstyn@eng.ox.ac.uk).

Prof. David Healey is the managing director of smart grid solutions and also Professor in Practice at Keele University (email:d.l.healey@keele.ac.uk).

Prof. Kang Li is with the School of Electronics and Electrical Engineering, University of Leeds, Leeds, UK (e-mail: k.li1@leeds.ac.uk).

approach to optimize the EV charging scheduling. A Q learning based algorithm is proposed in [15] for energy arbitrage on the real-time market. Compared to model-based methods, the data-driven approaches show great advantages: 1) they have self-adaptability, model-free nature, and the ability to learn from historical data; 2) Deep reinforcement learning (DRL) can learn a good control policy, even under a very complex environment by using deep neural networks. This feature provides great potentials for DRL to learn a battery charging/discharging policy for energy arbitrage considering a complicated, precise battery degradation model.

The objective of energy arbitrage using battery storage is to maximise the profits. In current literature, three relatively simple assumptions in energy storage arbitrage remain the major obstacles for its adoption in industry: 1) perfect foresight about electricity market prices; 2) constant battery charging/discharging efficiency; 3) simple representation of battery degradation model. This paper aims to address all these issues by using a deep reinforcement learning method. The contribution of this paper is to propose a self-learning noisy network based deep reinforcement learning approach to learn the optimized control actions for battery storage under very complex environment (e.g. accurate battery degradation, nonlinear charging/discharging efficiency and price uncertainty).

The remainder of this paper is organized as follows. Section II introduces the environment model of the battery storage and battery degradation costs. The control problem is formulated as a Markov Decision Process in Section III. The deep reinforcement learning algorithm is introduced in Section IV. Section V presents case studies results and finally Section VI concludes the paper.

II. Environment Model

To improve the training process of the proposed DRL method, the battery and battery degradation are modeled based on a standardized set of environments in OpenAI Gym [16] in this section.

A. Battery Energy Storage Model

In this paper, a generalized mathematical model of energy storage system based on state of charge (SoC) to describe the battery behaviour, is defined as follows:

$$\mathrm{SoC}_{t+1} = \begin{cases} \mathrm{SoC}_t - \frac{1}{E_{ess}} \cdot \eta_t^{ch} \cdot \int_t^{t+1} P_{e,t} dt, P_{e,t} < 0 \\ \mathrm{SoC}_t - \frac{1}{E_{ess}} \cdot \frac{1}{\eta_t^{dis}} \cdot \int_t^{t+1} P_{e,t} dt, P_{e,t} > 0 \\ \mathrm{SoC}_t - \frac{1}{E_{ess}} \cdot \int_t^{t+1} P_{standby,t} dt, P_{e,t} = 0 \end{cases} \tag{1}$$

where SoC_t is the state of charge at time t; $P_{e,t}$ is the output power of battery ($P_{e,t} > 0$, when discharging and $P_{e,t} < 0$, when charging); $P_{standby,t}$ is the standby losses of battery; E_{ess} is the energy capacity of battery (kWs); η_t^{ch} and η_t^{dis} are the charging and discharging efficiencies respectively.

In the conventional battery energy storage model, the charging/discharging efficiency is usually assumed to be constant. However, the efficiency is actually a nonlinear function of battery SoC and battery charging/discharging power [17]. To calculate the efficiency of battery, a steady state equivalent

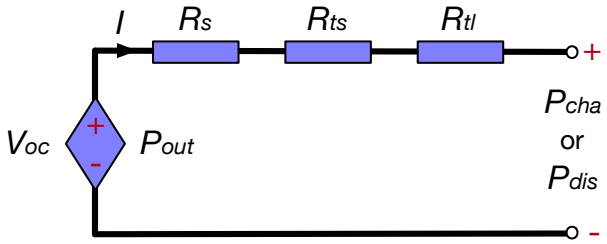

Fig. 1. Steady state battery equivalent circuit.

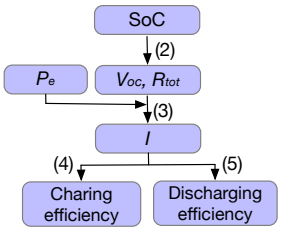

Fig. 2. The calculation process of discharging and charging efficiencies.

circuit model is adopted to represent the Li-ion battery, as shown in Fig. 1. The circuit consists of an open circuit voltage V_{oc} and three resistors (R_s, R_{ts}, R_{tl}) that represent different electrochemical processes: ohmic losses, charge tansfer and membrane diffusion. The open circuit voltage and three resistors are the nonlinear function of SoC, which can be expressed by [17]

$$\begin{cases} V_{oc} = a_0 e^{(-a_1 SoC)} + a_2 + a_3 SoC - a_4 SoC^2 + a_5 SoC^3 \\ R_s = b_0 e^{(-b_1 SoC)} + b_2 + b_3 SoC - b_4 SoC^2 + b_5 SoC^3 \\ R_{ts} = c_0 \cdot e^{-c_1 \cdot SoC} + c_2 \\ R_{tl} = d_0 \cdot e^{-d_1 \cdot SoC} + d_2 \\ R_{tot} = R_s + R_{ts} + R_{tl} \end{cases} \tag{2}$$

Then we can obtain the circuit current by solving the quadratic equation $P_e = I(V_{oc} - R_{tot} I)$ in Fig. 1:

$$I = \frac{V_{oc} - \sqrt{V_{oc}^2 - 4 \cdot R_{tot} \cdot P_e}}{2 \cdot R_{tot}} \tag{3}$$

The charging and discharging efficiencies of battery can be given by (4) and (5), respectively.

$$\eta^{ch} = \frac{V_{oc}}{V_{oc} - R_{tot} \cdot I} \tag{4}$$

$$\eta^{dis} = \frac{V_{oc} - R_{tot} \cdot I}{V_{oc}} \tag{5}$$

Fig. 2 shows the basic calculation process of charging and discharging efficiencies. For a particular SoC and charging/discharging power P_e, we can derive the efficiencies through the flowchart of Fig. 2. Fig. 3 shows the results of different charging/discharging efficiency corresponding to different SoC and charging/discharging rate (called C-rate, defined as the charge or discharge current divided by the battery's capacity). As seen in Fig. 3, the efficiency of a battery improves for higher SoC and lower C-rate.

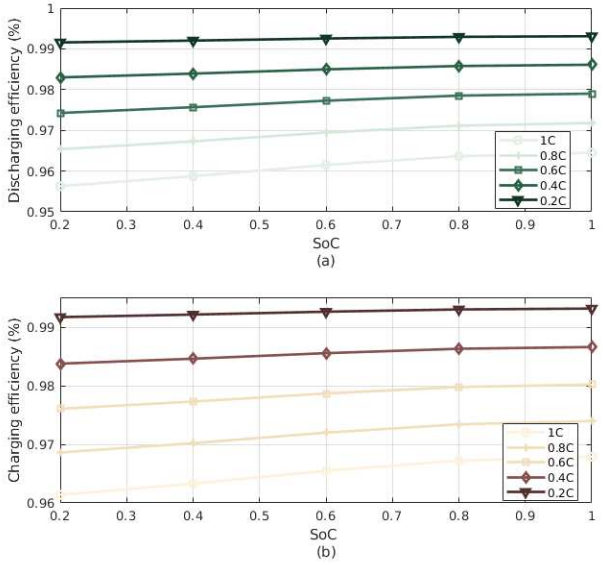

Fig. 3. Discharging and charging efficiencies.

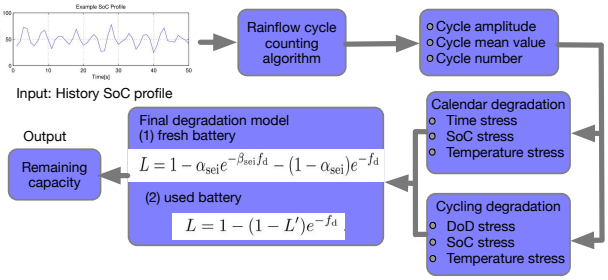

Fig. 4. Framework of the battery degradation assessment [11]

B. Battery Degradation Model

In the process of energy arbitrage, a key factor is the accurate estimation of the battery operating cost, which mainly stems from the battery degradation. An accurate battery degradation model as a function of the battery operation is needed to calculate the operation cost during the energy arbitrage.

The degradation process of battery is a nonlinear process with respect to time and cycle numbers, which is shown in Fig. 5. Basically, battery aging consists of two types of aging: (i) calendar aging and (ii) cyclic aging [18]. Calendar aging reflects the battery's inherent degradation over time, which is affected by the temperature and SoC. Cyclic aging is the capacity lost each time in the battery operation during charging and discharging and it depends on the depth of charge, discharging rate, ambient temperature, etc. [18].

A semi-empirical lithium-ion battery degradation model which can account for irregular cycling operations in [11] has been adopted to estimate the battery degradation costs. Fig. 4 shows the framework of calculation process. Firstly, a historical SoC profile is used as the input to the rainflow cycle-counting algorithm [19] and the output of the algorithm includes: 1) cycle amplitude; 2) cycle mean value; 3) cycle number; 4) cycle begin and end time. Then, both the calendar and cycling degradation results are combined to estimate the final remaining capacity of the battery. Fig. 6 shows the results

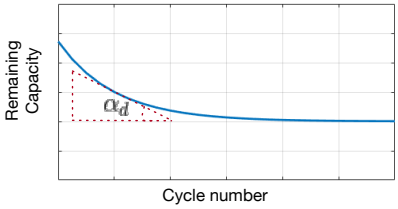

Fig. 5. General capacity degradation of lithium-ion battery [20]

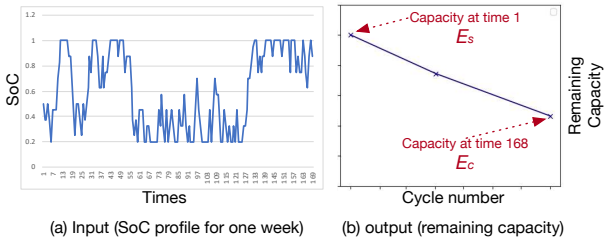

Fig. 6. The results of battery degradation using the framework in Fig. 4

of battery degradation using the framework in Fig. 4. The input of the algorithm is the random SoC profile for one week (168 hours), as shown in Fig. 6 (a). Fig. 6 (b) dispalys the final results, with the initial capacity E_s at time 1 and remaining capacity E_c at time 168.

However, this model in [11] can only estimate the degradation for a period of cycling operations, which will lead to a delayed reward to the reinforcement learning approach. It can hardly recognise which action (charging/discharging) is actually responsible for the high reward (degradation costs in this paper), as the rewards are delayed and accumulated.

Actually, the degradation process can be treated as a linear function to the cycle number during a short period of time (see Fig. 6(b)). To account for the immediate rewards in the learning process, a degradation coefficient α_d, which represents the slope of linear approximation of battery aging during a short period of time in Fig. 5, is proposed to estimate the reward for every charging or discharging control action. The coefficient α_d is updated based on the degradation results of the last training episode in reinforcement learning algorithm explained in Section IV. It is defined as:

$$\alpha_{d,j} = \frac{E_{s,j} - E_{c,j}}{\sum_{i=1}^{T} |P_{e,i}|} * C_B \tag{6}$$

where, $E_{s,j}$ and $E_{e,j}$ are the battery remaining capacity at the start and end point of the episode j. T is the time period per episode (168 hours in this paper). $P_{e,i}$ is the battery charging/discharging power at time i during the episode j. C_B is the battery cost per kWh.

III. Problem Formulation

A Markov Decision Process (MDP) model with discrete time step T_s is formulated in this section for the energy arbitrage problem. The time step T_s is chosen based on the time interval of the input data. In this paper, T_s is one hour according to the available data of the UK wholesale market. The whole sequential decision-making process of the MDP model for battery energy arbitrage is: given a state $s_t \in S$ at time step t which includes battery state of charge SoC_t

and next 24 hours electricity prices from forecasting, the agent selects an action (charging or discharging) from a action-space $a \in A(s)$ based on the policy π. The goal of the proposed algorithm is to find the optimal policy to maximise the reward (profit) in the energy arbitrage process. The MDP formulation for energy arbitrage is defined as:

1) State space: The state space at time instant t is defined as $s_t = (c_t, ..., c_{t+22}, c_{t+23}, SoC_t)$. where $c_t, ..., c_{t+22}, c_{t+23}$ is the predicted price for the next day. Using the predicted price signal is to make sure the agent knows whether the price signal is going up or down in order to make the best control action. The state transition of the battery SoC from state s_t to $s_t + 1$ is defined in (1).

2) Action space: The charging/discharging action space is discrete as $a = (-P_e^{max}, -0.5P_e^{max}, 0, 0.5P_e^{max}, P_e^{max})$, where P_e^{max} is the maximum charging/discharging power of the battery. The actual charging/discharging power is limited by (7) due to the limit of SoC.

$$\frac{(SoC_t - SoC_{max}) \cdot E_{ess}}{\eta_t \cdot T_s} \leq P_{e,t} \leq \frac{(SoC_t - SoC_{min}) \cdot E_{ess}}{\eta_t \cdot T_s} \tag{7}$$

where SoC_{max} and SoC_{min} are the maximum and minimum state of charge of the battery, respectively. η_t is the charging/discharging efficiency defined in (1).

3) Reward: The design of reward function is the key factor in the algorithm. The reward in the energy arbitrage problem should include not only the profit from the discharging action, but also the degradation costs of the control action. The immediate reward R_t at time step t is defined as follows:

$$R_t = c_t \cdot \frac{P_{e,t}}{P_e^{max}} - \alpha_d \cdot \frac{|P_{e,t}|}{P_e^{max}} \tag{8}$$

where $P_{e,t} \cdot c_t$ denotes the charging cost when $P_{e,t} < 0$ and discharging revenue when $P_{e,t} > 0$. $\alpha_d \cdot |P_{e,t}|$ represents the cost of battery degradation. α_d is updated every training episode. To improved the results and the speed of training, the reward scale technique suggested by [21] is adopted to clip the reward between -1 and 1.

The cumulative profits during the energy arbitrage are denoted as:

$$R_t^{cum} = \sum_t^T (P_{e,t} \cdot c_t - \alpha_d \cdot |P_{e,t}|) \tag{9}$$

R_t^{cum} is used as the only metric to evaluate the performance of different methods.

IV. Proposed Algorithm

A. Reinforcement Learning

In this section, the background of the RL is introduced.

1) Q-learning: Q-learning is a model-free reinforcement learning algorithm. The goal of Q-learning is to let the agent learn a best policy in a given state by exploring the environment [22]. The quality of the charging/discharging action a in a given state s is determined by the action-value function, denoted as $Q_\pi(s, a)$ for policy π, which is defined as:

$$Q_\pi(s, a) = \mathbb{E}_\pi \left[\sum_{k=0}^{k=K} \gamma^k \cdot R_{t+k} \mid s_t = s, a_t = a \right]. \tag{10}$$

where γ is the discount factor, and the policy π maps from the system states to the charging/discharging action.

By exploring the environment, the agent will iteratively update the action-value function $Q_\pi(s, a)$ using the following Bellman Equation:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[R_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right] \tag{11}$$

where α is the learning rate.

The iteration will continue until it converges to the best action-value function $Q_\pi^*(s, a)$. Then, the choosing action is determined by the ϵ-greedy policy, which at every timestep t, the agent selects the greedy action $a_t = argmax_a Q(s, a)$ with probability $1 - \epsilon$ and selects a random action to explore a better reward with probability ϵ. The $Q_\pi^*(s, a)$ is approximated by a look up table in Q-learning.

2) Deep Q-network (DQN): Q-learning is confronted with a difficult task when the state or action space are high-dimensional. One solution proposed by Google DeepMind [12] is to use a deep neural network to approximate the optimal action-value function $Q_\pi^*(s, a)$. The represented value function by DQN with weights ω is denoted as:

$$Q(s_t, a_t; \omega) \approx Q^*(s_t, a_t) \tag{12}$$

The objective of DQN is to minimise the Mean Squared Error (MSE) loss $L(\omega)$ between $Q(s, a)$ and TD (temporal difference) target by Stochastic Gradient Descent (SGD):

$$L(\omega) = (R_t + \gamma \max_a Q(s_{t+1}, a_{t+1}; \omega^-) - Q(s_t, a_t; \omega))^2 \tag{13}$$

where, the TD target is $y_t = R_t + \gamma \max_a Q(s_{t+1}, a_{t+1}; \omega^-)$. In (13), we actually use a separate network (target network) with a fixed parameter ω^- for estimating the TD target y_t and the parameters from DQN network ω are copied to update the target network ω^- periodically.

Some other improvements to the DQN includes: Double DQN (DDQN), Dueling DQN and Noisy Networks for Exploration, which will be introduced in the following parts.

3) DDQN: The standard DQN suffers from upward bias caused by $\max_a Q(s, a; \omega)$ in (13) [23]. DDQN mitigates the issue by using two separate networks to decouple the action selection from the target Q value generation.

In DDQN, we use the current DQN network ω to select what is the best action to take for the next state (the action with the highest Q value) and use the older target network ω^- to evaluate the target Q value of taking that action at the next state. The TD target of DDQN is defined as:

$$y_t^{DDQN} = R_t + \gamma Q(s_{t+1}, \underset{a_{t+1}}{\operatorname{argmax}} Q(s_{t+1}, a_{t+1}, \omega), \omega^-) \tag{14}$$

4) Dueling DQN: To further improve the DQN, the dueling DQN approximates the Q-function by decoupling the action-independent value function $V(s, v)$ and the advantage function $A(s, a, \omega)$ [24].

Instead of using a single stream of fully connected layers for Q-value estimation, the dueling network uses two streams of fully connected layers with parameters v and ω respectively. One stream is used to provide value function estimate given

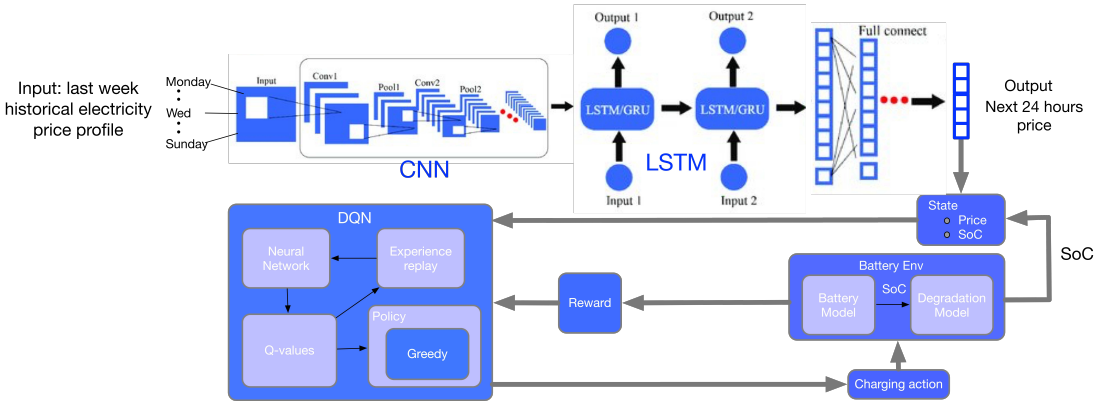

Fig. 7. The overall framework of the proposed approach. (The top part is the proposed prediction algorithm based on hybrid CNN and LSTM networks; the bottom part is the basic DQN approach. To improve the stability of training process, we use the experience replay mechanism [12] which stores the state transitions in a replay buffer and randomly sampled during the training).

a state, while the other stream is for estimating advantage function for each valid action. Finally, the two streams are combined in a way to produce and approximate the Q-function, which is denoted as follows:

$$Q(s,a) = V(s,v) + A(s,a,\omega) \tag{15}$$

5) Noisy network for Exploration

An alternative approach to exploration when using neural network to approximate the action-value function is Noisy Networks for Exploration [25] that replaces the linear layer with a noisy linear layer, which is defined as:

$$Y = (\mu^{\omega} + \sigma^{\omega} \odot \epsilon^{\omega})X + (\mu^{b} + \sigma^{b} \odot \epsilon^{b}) \tag{16}$$

where $\epsilon = [\epsilon^{\omega}, \epsilon^{b}]$ are randomly sampled, zero mean noise matrices with fixed statistics, and $\mu = [\mu^{\omega}, \mu^{b}]$ and $\sigma = [\sigma^{\omega}, \sigma^{b}]$ are the learning parameters of the network. In noisy network, instead of using an $\epsilon-$ greedy policy, the agent can act greedily according to a network using noisy linear layers.

B. Proposed algorithm

The overall framework of the proposed approach is shown in Fig. 7. The first part of the algorithm is forecasting the electricity price using hybrid CNN and LSTM network. Then the prices predicted concatenated with other features such as SoC are fed into the DRL to learn the optimal policy. The detailed explanation of these two parts are shown as follows:

1) Price forecasting: The goal of the proposed forecasting approach is to forecast the hourly market price of the next day (24 hours), by using the historical price data of the last one week (168 hours). There are three steps in the forecasting approach:

(i) Data pre-processing: the database used has some extreme high peaks which are caused by either the market failure or data errors. To reduce the impact of data outliers on prediction accuracy, all the values that are outside the range of 15% and 85% quantiles are replaced by the threshold values. Then, the price data are scaled to [0,1] values by using MinMaxScaler function in Python Sklearn [26].

(ii) Model architecture design: The proposed model uses a combined CNN and LSTM networks. LSTM network is well known for modelling the time series data [27] and has shown great advantages in load forecasting using smart meter data [28], [29]. The reason for adding a CNN layer prior to the LSTM network is to incorporate multiple features simultaneously (other features such as weather, generation) and reduce the temporal input dimension if only one feature is included. In this paper, only one feature is included in the input data which is the price. The CNN layer can reduce the temporal input dimension (from 1×168 to 7×24). Finally, a fully connected layer with 24 nodes is connected to the output. Each node is corresponding to every hour that predicted.

(iii) Training and accuracy assessment: The architecture designed in step (ii) will be tuned and trained. The final trained model will be used for prediction and the accuracy will be assessed using Mean Absolute Error (MAE).

2) NoisyNet-DDQN algorithm (NN-DDQN): The detailed algorithm for energy arbitrage using NN-DDQN is presented in Algorithm 1.

V. Case Study

In this section, we evaluate the proposed approach using actual UK wholesale market electricity price [30]. Electricity prices from Yeas 2015 and 2016 are used as the training and testing data, respectively.

We use five Lithium-ion batteries and each battery has the capacity 200kWh and the charging/discharging power is discretised to [-100kW, -50kW, 0, 50kW, 100kW]. The battery parameters for calculating efficiency are shown in Table I. The whole training takes about three and half hours on a Computer with GPU GTX 1080 Ti and CPU i7-7800X. Once the training is finished, the proposed approach takes about 5ms to output the control actions, which could be used in real time control. The algorithm is developed on Python and Keras, which is a high-level neural networks API [31].

A. Forecasting method evaluation

The price forecasting method proposed in Section IV-B is adopted to predict the electricity price and the model architecture developed in Keras is shown in Table II. The data

Algorithm 1 NN-DDQN for Energy Arbitrage

Initialize the ϵ set of random variables of the network;
Initialize the network and target network parameters;
Initialize the reply memory: D and the mini-batch size;
for Episode $e = 1$ **to** J **do**
  Observe state space $s_t = (c_{t-23}, c_{t-22}, ...c_t, SoC_t)$
  for $t = 1, \dots T$: **do**
    Sample zero mean noisy ϵ
    Select an action $a_t = argmax_a(Q(s, a))$
    Execute action a_t, receive reward r_t and next state s_{t+1}
    Store transition s_t, a_t, r_t, s_{t+1} in D
    Sample random mini-batch of transitions s_j, a_j, r_j, s_{j+1} from D
    Sample the noisy variable for the online and target network ϵ
    Estimate the target y_j
    $y_j = R_j + \gamma Q(s_{j+1}, \text{argmax}_{a'} Q(s_{j+1}, a', \omega), \omega^-)$
    Do a gradient descent with loss $y_j - Q(s_t, a_t, \omega))^2$
    Every C steps update $\omega' = \omega$
  end for
end for

TABLE I
BATTERY PARAMETERS IN (2) [17]

a_0	-0.852	a_1	63.867	a_2	3.6297	a_3	0.559
a_4	0.51	a_5	0.508	b_0	0.1463	b_1	30.27
b_2	0.1037	b_3	0.0584	b_4	0.1747	b_5	0.1288
c_0	0.1063	c_1	62.94	c_2	0.0437	d_0	-200
d_1	-138	d_2	300				

are randomly splitted using `train_test_split` function in Sklearn [26]. The input data spans a whole last week of electricity prices (168 hours) and these data are fed into the convolutional layer with a kernel size and a stride of 24, which results in a length 7 per feature map. The output is the electricity price prediction for the next day (24 hours).

Fig. 8 shows the forecasting results of one week during summer and winter seasons. We can clearly see that the model can learn not only the daily variations of prices, but also the week and seasonal patterns (more peaks values during winter). The forecasting accuracy MAE is 4.686 in this case.

B. Performance Evaluation of NN-DDQN

The performance of the proposed NN-DDQN is evaluated using the electricity prices at year 2016 in this section. To compare the effectiveness of the proposed approach, the proposed NN-DDQN is compared with other two DRL methods:

TABLE II
MODEL ARCHITECTURE IN KERAS

Layer type	Output shape	Param
Input Layer	(None, 168,1)	0
Conv1D	(None, 7, 128)	3200
LSTM	(None, 32)	20608
Dense	(None, 24)	792

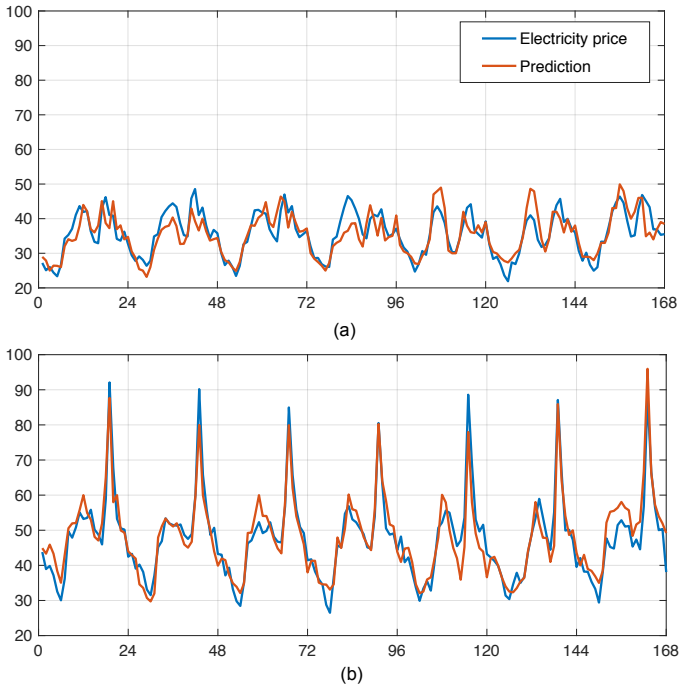

Fig. 8. Electricity price forecasting results during summer and winter season. ((a)Electricity forecasting results from 1st July to 7th July; (b)Electricity forecasting results from 1st Jan to 7th Jan)

TABLE III
SUMMARY OF DRL TRAINING SETTINGS

Item	Value
No. of hidden layers	3
No. of nodes in each layer	16
Activation function	ReLU
Learning rate	0.00025
Optimizer	Adam optimizer
batch size	32
Target model update	10000 steps

Vanilla DQN and Double dueling DQN, described in Section IV. All the training settings are summarized in Table III. The NN-DDQN is trained with 12000 episodes. The convergence process of the episode rewards over 12000 episodes for these three methods is illustrated in Fig.10. It can be observed that the NN-DDQN is more stable during the training process, compared with other two approaches. It can converge to the optimized reward which is around 6 at episode 2200. As the NN-DDQN keeps on choosing random actions with a small probability of epsilon 0.01, therefore the episode rewards keep fluctuating.

After training, the optimal weight parameters of NN-DDQN are used to control the charging/discharging actions of battery storage using the electricity price at year 2016. Fig. 9 shows the charging/discharging results over one week for different summer and winter price patterns. The electricity prices are illustrated with the green line and the SoC, charging/discharging actions are represented with blue and red bars respectively. The charging power (-100kW) and discharging power (100kW) are scaled to -1 and 1 to allow them draw on one figure. We can clearly see that the proposed approach can learn the optimized charging/discharging strategy (charging during low

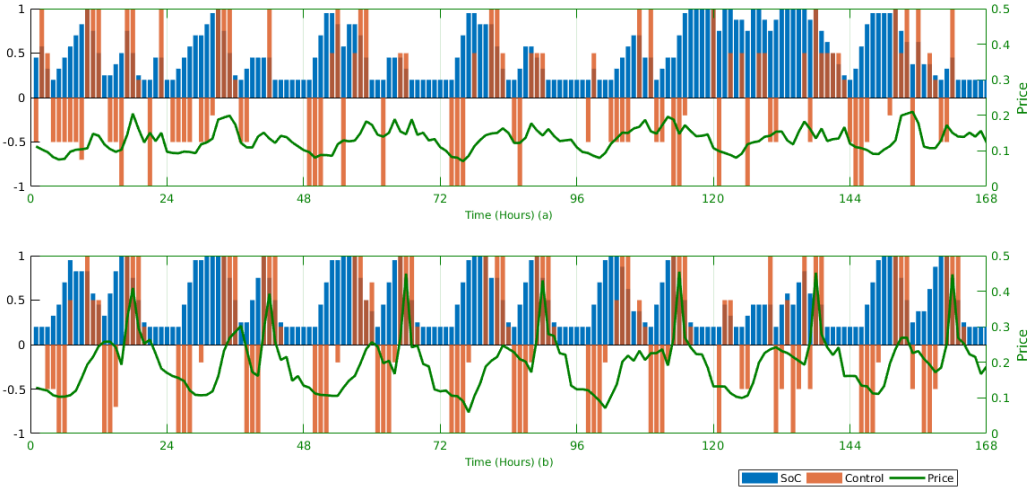

Fig. 9. The charging/discharging results over one week for summer (a) and winter (b). (Blue bar: SoC; Red bar: charging(-)/discharging(+) actions, the values are scaled from [-100kW, 100kW] to [-1, 1]; The green curve with the right axis represents the electricity prices)

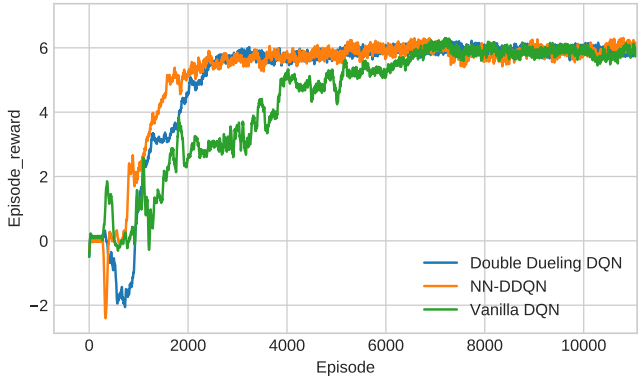

Fig. 10. The episode reward during the training process

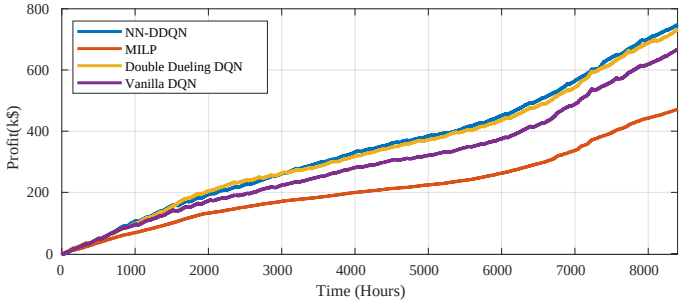

Fig. 11. Comparison results of cumulative profits with MILP

prices, and discharging during high prices) for battery not only in the summer period, but also during the winter when the variation of electricity price is quite high. We have found that the battery experiences roughly two cycles per day during the winter season and it can still make profits considering the higher price difference between the peak and valley electricity price in the whole sale market.

C. Comparison Results

1) Comparison results with model-based method: The proposed approach is compared with the mixed integer linear programming (MILP) shown in Appendix. The electricity price in MILP is predicted using the same prediction method illustrated in Section IV-B. The cumulative profits of the proposed methods and MILP method over the whole year of 2016 are presented in Fig. 11. We can observe that the proposed NN-DDQN improves the profits by 58.51% in comparison with the MILP method. In addition, the NN-DDQN shows better results in comparison with Vanilla DQN and Double Dueling DQN.

2) Comparison results without uncertainty (perfect forecasting of price): The proposed approach is compared with perfect forecasting of electricity price to show how forecasting algorithm influences the results. The cumulative profits of the proposed methods and perfect forecasting over the whole year of 2016 are presented in Fig. 12. We can observe that the perfect forecasting can improve the profits by 4.63% in comparison with the proposed NN-DDQN method. The reason of this small difference is that the proposed prediction method illustrated in Section IV-B can already predict the electricity price accurately as shown in Fig. 8.

3) Comparison results without accurate degradation model: The proposed approach is compared to the model that does not consider the battery degradation which means $\alpha_{d,j} = 0$ in (6). The cumulative profits of the proposed methods and the model without battery degradation over the whole year of 2016 are presented in Fig. 12. We can observe that the model without considering degradation can influence the profits by 5.13% in comparison with the proposed NN-DDQN method.

4) Comparison results with different hyperparameters of training: The proposed approach is compared with different hyperparameters of training shown in Table III. The number of hidden layers in Table III is changed to 4 in the comparison. The cumulative profits of the proposed methods and different hyperparameters over the whole year of 2016 are presented in Fig. 12. We can observe that fine-tuned hyperparameters in NN-DDQN can improve the profits.

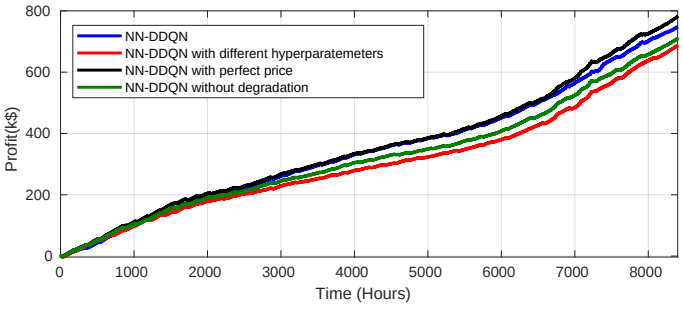

Fig. 12. Comparison results of cumulative profits

VI. Conclusions

In this paper, we have proposed a charging/discharging strategy for energy storage participating in the energy arbitrary based on DRL methods, which is a model-free approach, and can learn any complex system models. We use DRL methods to address three challenges in energy storage arbitrage: nonlinear efficiency of battery charging/discharging, accurate battery degradation model and electricity price uncertainty. In the DRL, a combined CNN and LSTM hybrid network is proposed to predict the electricity prices. Then a NN-DDQN is implemented to learn the optimal control policy of battery considering the price uncertainty and battery degradation. Experimental results using actual electricity prices have demonstrated the effectiveness of the proposed methods.

References

[1] X. Luo, J. Wang, M. Dooner, and J. Clarke, "Overview of current development in electrical energy storage technologies and the application potential in power system operation," *Applied Energy*, vol. 137, pp. 511 – 536, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261914010290

[2] H. Khani and M. R. D. Zadeh, "Real-Time Optimal Dispatch and Economic Viability of Cryogenic Energy Storage Exploiting Arbitrage Opportunities in an Electricity Market," *IEEE Tran. on Smart Grid*, vol. 6, no. 1, pp. 391–401, JAN 2015.

[3] D. Metz and J. T. Saraiva, "Use of battery storage systems for price arbitrage operations in the 15-and 60-min German intraday markets," *Electric Power Systems Research*, vol. 160, pp. 27–36, JUL 2018.

[4] D. Krishnamurthy, C. Uckun, Z. Zhou, P. R. Thimmapuram, and A. Botterud, "Energy Storage Arbitrage Under Day-Ahead and Real-Time Price Uncertainty," *IEEE Tran. on Power Systems*, vol. 33, no. 1, pp. 84–93, JAN 2018.

[5] H. Akhavan-Hejazi and H. Mohsenian-Rad, "Optimal operation of independent storage systems in energy and reserve markets with high wind penetration," *IEEE Tran. on Smart Grid*, vol. 5, no. 2, pp. 1088–1097, March 2014.

[6] A. A. Thatte, L. Xie, D. E. Viassolo, and S. Singh, "Risk Measure Based Robust Bidding Strategy for Arbitrage Using a Wind Farm and Energy Storage," *IEEE Tran. on Smart Grid*, vol. 4, no. 4, pp. 2191–2199, DEC 2013.

[7] A. Attarha, N. Amjady, and S. Dehghan, "Affinely Adjustable Robust Bidding Strategy for a Solar Plant Paired With a Battery Storage," *IEEE Tran. on Smart Grid*, vol. 10, no. 3, pp. 2629–2640, MAY 2019.

[8] F. Wankmller, P. R. Thimmapuram, K. G. Gallagher, and A. Botterud, "Impact of battery degradation on energy arbitrage revenue of grid-level energy storage," *Journal of Energy Storage*, vol. 10, pp. 56 – 66, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352152X16303231

[9] N. Padmanabhan, M. Ahmed, and K. Bhattacharya, "Battery energy storage systems in energy and reserve markets," *IEEE Tran. on Power Systems*, pp. 1–1, 2019.

[10] T. Ashwin, Y. M. Chung, and J. Wang, "Capacity fade modelling of lithium-ion battery under cyclic loading conditions," *Journal of Power Sources*, vol. 328, pp. 586 – 598, 2016.

[11] B. Xu, A. Oudalov, A. Ulbig, G. Andersson, and D. S. Kirschen, "Modeling of lithium-ion battery degradation for cell life assessment," *IEEE Tran. on Smart Grid*, vol. 9, no. 2, pp. 1131–1140, March 2018.

[12] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, p. 529, Feb 2015.

[13] J. R. Vazquez-Canteli and Z. Nagy, "Reinforcement learning for demand response: A review of algorithms and modeling techniques," *Applied Energy*, vol. 235, pp. 1072–1089, FEB 2019.

[14] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time ev charging scheduling based on deep reinforcement learning," *IEEE Tran. on Smart Grid*, vol. 10, no. 5, pp. 5246–5257, Sep. 2019.

[15] H. J. Wang and B. Zhang, "Energy storage arbitrage in real-time markets via reinforcement learning," *IEEE Power Energy Society General Meeting (PESGM)*, pp. 1–5, 2018.

[16] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.

[17] T. Morstyn, B. Hredzak, R. Aguilera, and V. Agelidis, "Model predictive control for distributed microgrid battery energy storage systems," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 3, pp. 1107–1114, May 2018.

[18] J. Vetter, P. Novk, M. Wagner, C. Veit, K.-C. Mller, J. Besenhard, M. Winter, M. Wohlfahrt-Mehrens, C. Vogler, and A. Hammouche, "Ageing mechanisms in lithium-ion batteries," *Journal of Power Sources*, vol. 147, no. 1, pp. 269 – 281, 2005.

[19] S. Downing and D. Socie, "Simple rainflow counting algorithms," *International Journal of Fatigue*, vol. 4, no. 1, pp. 31 – 40, 1982.

[20] R. Spotnitz, "Simulation of capacity fade in lithium-ion batteries," *Journal of Power Sources*, vol. 113, no. 1, pp. 72 – 80, 2003.

[21] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/viewPaper/16669

[22] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2018.

[23] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," *CoRR*, vol. abs/1509.06461, 2015. [Online]. Available: http://arxiv.org/abs/1509.06461

[24] Z. Wang, N. de Freitas, and M. Lanctot, "Dueling network architectures for deep reinforcement learning," *CoRR*, vol. abs/1511.06581, 2015. [Online]. Available: http://arxiv.org/abs/1511.06581

[25] M. Fortunato, M. G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin, C. Blundell, and S. Legg, "Noisy networks for exploration," *CoRR*, vol. abs/1706.10295, 2017. [Online]. Available: http://arxiv.org/abs/1706.10295

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[28] 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 Tran. on Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan 2019.

[29] Y. Wang, D. Gan, M. Sun, N. Zhang, Z. Lu, and C. Kang, "Probabilistic individual load forecasting using pinball loss guided lstm," *Applied Energy*, vol. 235, pp. 10 – 20, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261918316465

[30] (2017) The changing price of wholesale uk electricity over more than a decade. [Online]. Available: https://www.ice.org.uk/knowledge-and-resources/briefing-sheet/the-changing-price-of-wholesale-uk-electricity

[31] (2019) Keras: The python deep learning library. [Online]. Available: https://keras.io/

[32] IBM. (2019) IBM ILOG CPLEX optimization studio. [Online]. Available: www.cplex.com

VII. Appendix

The energy arbitrage problem is formulated as a MILP and solved using CPLEX [32] in Python. The objective of the

MILP is:

$$\sum_{k=1}^{N}((P_{c,k}-P_{d,k})\cdot c_k-\alpha_d(P_{c,k}+P_{d,k})) \tag{17}$$

constraints:

$$\begin{cases} SoC_k = SoC_{k-1}-\eta_d P_{d,k}\cdot u_{d,k}+\eta_c P_{c,k}\cdot u_{c,k} \\ 0\le P_{c,k}\le u_{c,k}P_c^{\max} \\ 0\le P_{d,k}\le u_{d,k}P_d^{\max} \\ 0\le u_{c,k}+u_{d,k}\le 1 \\ u_{c,k},\, u_{d,k}\in\{0,1\} \\ 0.2\le SoC_k\le 1 \\ SoC_0 = 0.5 \end{cases} \tag{18}$$

The state of charge of a storage unit, denoted by SoC_k, at time step k depends on its state of charge in the previous time step $k-1$ and the current charge power $P_{c,k}$ or discharge power $P_{d,k}$. Losses of the battery are represented by charging/discharging efficiencies η_c and η_d respectively.

c_k is the predicted prices using the method in Section IV-B. $u_{d,k}$ is a binary variable with $u_{d,k}=1$ if the battery is discharging and $u_{d,k}=0$ if the battery is charging. The binary variables $u_{d,k}$ and $u_{c,k}$ prevent the model from using the charge and discharge efficiencies of the storage units to dump energy by simultaneously charging and discharging the battery.