Energy Storage Arbitrage in Day-Ahead Electricity Market Using Deep Reinforcement Learning

Tim Zonjee, Shahab Shariat Torbaghan
Environmental Technology, *Wageningen University & Research*, Wageningen, The Netherlands
tim.zonjee@wur.nl

***Abstract*—Large scale integration of renewable and distributed energy resources increases the need for flexibility on all levels of the energy value chain. Energy storage systems are considered as a major source of flexibility. They can help with maintaining a secure and reliable grid operation. The problem is that these technologies are capital intensive and therefore, there is a need for new algorithms that enable arbitrage while ensuring financial feasibility. To this end, in this research, we develop a constrained deep Q-learning based bidding algorithm to determine the optimal bidding strategy in the day-ahead electricity market. The proposed algorithm ensures compliance to energy storage system constraints. It takes imperfect, yet reasonably accurate, 24-hour-ahead price forecast data as an input and returns the optimal bidding strategy as output. The numerical results and the sensitivity analysis show that the proposed algorithm effectively contains the impact of price forecast uncertainty to guarantee financial feasibility.**

***Index Terms*—Energy Storage, Energy Arbitrage, Deep Reinforcement Learning, Deep Q-Network, Day-Ahead Electricity Market.**

I. Introduction

A. Background

Decarbonization is a major pillar of the ongoing transition in energy systems. Decarbonization via renewable energy resources with their varying and intermittent profile, has increased the uncertainty at all levels of the energy value chain [1]. The increasing uncertainties have introduced new challenges in maintaining the supply and demand balance and in ensuring the reliable and secure operation of the electricity network [2]. One way to manage the uncertainty is to increase flexibility in the energy system [3].

Grid connected energy storage systems (ESS) are seen as a valuable source of flexibility to manage these uncertainties [4]. The main challenge is to ensure financial feasibility of ESS operation to encourage owners to offer available capacity of their ESS when needed. One way to achieve this is by devising a bidding algorithm that ensures profitability of energy storage arbitrage in electricity markets. The challenge for a profit maximization algorithm as such, lies in determining the optimal bidding strategy despite being provided with uncertainty associated erroneous market price forecasts as an input.

B. Literature Review

The energy arbitrage problem can be characterized as a sequential decision-making problem, where an agent makes decisions on how to interact with the market and ESS environments [5]. One approach to solving such a problem is by mathematical optimization (MO). In [6], the authors describe a mixed-integer linear programming (MILP) approach to find the optimal bidding strategy in intraday (ID) markets under the assumption of perfect market price forecasts. In [7] and [8], the authors respectively propose a stochastic and robust formulation of the energy arbitrage problem in real-time (RT) markets to handle uncertainty in price forecasts.

Another approach to solving the energy arbitrage problem is by reinforcement learning (RL). RL is a machine learning approach based on trial-and-error learning, meaning that little prior information about the system is needed. In [9] and [10], the authors use RL to solve the energy arbitrage problem for the RT market by combining an historical price moving average with a Q-learning [11] and a proximal policy optimization [12] algorithm respectively. The authors of [13] propose a Deep Q Network (DQN) [14] based algorithm which exploits raw historical price forecast data for solving the energy arbitrage problem in day-ahead (DA) markets. More recent RL works extend the scope to maximizing profits from a combination of both energy arbitrage, and another revenue source simultaneously. In [15], this additional revenue source is obtained by simultaneously applying a DQN-based algorithm to maximize savings from self-utilization of renewable energy source supply technologies. In [16], a Q-learning agent is trained to simultaneously serve the grid operator by peak-load shifting, and in [17] a deep-deterministic policy gradient based algorithm [18] is applied to simultaneously participate in real-time frequency regulation services.

We distinguish two reasons why one would solve the energy arbitrage problem using an RL rather than an MO based approach. First, RL-based algorithms have the ability to learn how to deal with imperfect forecasts since the optimal decision-making strategy is determined directly from raw historical price forecasts. Second, RL enables offline learning, meaning that the computationally expensive learning of the optimal bidding strategy is detached from the quicker real-time decision-making process. MO algorithms on the other hand require online optimization for solving the decision-making problem as a whole. This is more computationally expensive, considering that the above-mentioned MO algorithms need to be provided with a multitude of uncertainty scenarios to handle price forecast uncertainty. This phenomenon becomes especially problematic when the decisions are to be revised in short periods before the market gate-closure-time.

In the above-mentioned literature on RL-based approaches,

2023 IEEE Belgrade PowerTech | 978-1-6654-8778-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/POWERTECH55446.2023.10202674

it is assumed that either DA forecasts are available one hour ahead, or that the agent requires more price forecasts than just the day ahead as an input. The former assumption does not fully acknowledge the complexity of the DA market. The latter assumption means a reliance on less accurate forecasts, since price forecasts become less accurate when the time horizon increases [19]. We argue that an agent would only require the 24 hour-ahead DA price forecasts as an input, to optimally solve the energy arbitrage problem.

Each of the above-mentioned RL works mentions application of safety constraints to the ESS energy capacity. One can do so by modifying the optimization criterion (soft constraint) or the exploration (learning) process (hard constraint) [20]. In [15] and [16], the authors modify the optimization criterion by punishing the agent for capacity constraint violation. In [9], [10] and [17], the exploration process is modified by optionally and iteratively changing the magnitude of the (dis)charging action to comply with the ESS energy capacity constraint. We argue that both approaches reduce the potential for the RL algorithm to converge to the optimal bidding strategy. Therefore, in this research, we formulate a hard constraint on the ESS energy capacity and solve the constrained problem using a constrained deep Q-network (CDQN) based algorithm.

Finally, it should be noted that grid-connected intermittent renewable energy sources are expected to increase. This development probably causes an increase in price volatility and forecasting complexity in various electricity markets. To analyze the performance of an RL-based algorithm in this context, we perform a global sensitivity analysis on the market price forecast uncertainty in this research. To our knowledge, such an analysis is still missing from existing literature.

C. Contributions

The contributions of the paper are as follows:

1) It introduces a CDQN bidding algorithm to solve an energy arbitrage problem. Taking imperfect point-wise price forecasts as an input, the algorithm learns to find the near-optimal bidding strategy to execute energy storage arbitrage in the DA electricity market. The proposed CDQN is a sequel to the DQN proposed in [13].
2) It reformulates the energy arbitrage problem by introducing a new state space consisting of the ESS's state of charge (SoC), a fixed 24-hour forecast, and an hour counter for the algorithm to extract the bidding hour. The combination of the latter two enables the use of accurate and realistic price forecasts.
3) It performs a global sensitivity analysis on the performance of the CDQN agent with respect to the market price forecast uncertainty.

D. Structure of the Paper

The remainder of this paper is organized as follows; Section II, presents the CDQN architecture and the problem formulation. Section III, evaluates the performance of the proposed methodology by applying it to a flow battery ESS and comparing it to the performance of a conventional MILP method. Section IV concludes the paper and proposes future research directions.

II. Methodology

In this section we first present a generalized introduction to the proposed CDQN algorithm. Second, we formulate the energy arbitrage problem as a Markov Decision Process (MDP). Last, we present the complete algorithmic implementation.

A. Constrained Deep Q Network Architecture

In this subsection we explain the background of (Deep) RL and the proposed CDQN for solving the optimal bidding problem.

1) Q-Learning: In RL, the goal of an agent is to maximize the cumulative reward from taking consecutive actions. This translates in the agent trying to learn the optimal policy π^* from any state $s_t \in \mathcal{S}$. Q-learning is an RL approach that works off-policy and is model-free. In Q-learning, the optimal policy for a given state is found by taking the action $a_t \in \mathcal{A}$ that has the maximum Q-value. The Q-value of a state-action pair is defined as:

$$Q(s_t, a_t) = E[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t, a_t] \tag{1}$$

where γ denotes the discount factor. r_t is the reward signal resulting from a specific state-action pair. To determine the true Q-values of all state-action pairs, the agent has to explore the full environment by iteratively taking actions from different states, resulting in differing rewards. In this research, we apply TD(0)-learning [21], meaning Q-values belonging to specific state-action pairs are updated each single iteration following:

$$\begin{aligned} Q_{new}(s_t, a_t) \;=\; & Q_{old}(s_t, a_t) + \\ & \alpha(r_t + \gamma \max_{\mathcal{A}} Q(s_{t+1}, \mathcal{A}) - Q_{old}(s_t, a_t)) \end{aligned} \tag{2}$$

where α is the learning rate. The new Q-value is then stored in a look-up table. Training a Q-learning agent takes a predefined number of episodes $e \in \mathcal{E}$, where each episode consists of $h \in \mathcal{H}^e$ iterations. In each iteration, the agent observes the state, and then selects an action following the epsilon greedy algorithm. The agent either explores the environment with probability ϵ by taking $a_t = rand(\mathcal{A})$ or exploits its knowledge about the action with the highest Q-value with probability $1 - \epsilon$ by taking $a_t = \arg\max_{\mathcal{A}} Q(s_t, \mathcal{A})$. To enable efficient training, ϵ generally decays over the episode number through $\epsilon = D^e$ with epsilon decay factor D such that the agent mainly explores the environment at the start of its training, and exploits its knowledge about a good policy more and more while further in training. [22].

2) Deep Q-Network: Q-learning is reliant on a look-up table that presents Q-values for each state-action pair. However the problem becomes intractable when the state or action spaces are high-dimensional or continuous. A solution to this problem is the Deep Q-Network in which the Q-values are approximated using a deep neural network:

$$Q(s_t, a_t; w) \approx Q(s_t, a_t) \tag{3}$$

where w denotes the DQN weight configuration. The goal in training the DQN-agent is set to find the 'true' Q-value by minimizing the difference between the old and new Q-value through stochastic gradient descent. Following (2), this is equivalent to minimizing the Mean Squared Error loss term, given by:

$$L(w) = (r_t \ + \ \gamma \max_{\mathcal{A}} Q(s_{t+1}, \mathcal{A}; w^-) \ - \ Q_{old}(s_t, a_t; w))^2 \tag{4}$$

where w^- denotes the target network weight configuration which is a copy of the DQN weight configuration, taken every U number of episodes. In a DQN-based algorithm, within a single step h, the decision-making process is decoupled from the process of updating the DQN. First, the step that has been decided upon, which we call an experience, is stored as a 4-dimensional array (s_t, a_t, r_t, s_{t+1}) in the experience replay memory (ERM) with size M. Next, a size X batch of experiences is randomly drawn from the ERM. This batch is then used to get the old and new Q-values, apply (4), and finally update the DQN weight configuration. Both the inclusion of a target network and training on a batch of experiences have been proven to improve the DQN training process. [14].

3) Constrained Deep Q-Network: To conform to the ESS energy capacity boundaries, we formulate a system constraint as given in (8). Constrained DQN (CDQN) is a version of DQN that ensures both compliance with system constraints at every time instant and the potential to converge to the optimal solution [23]. In the energy arbitrage problem, the permissible action space $\mathcal{A}'_t$ is dependent on time, and therefore should be shrunk to a subset of the original action space $\mathcal{A}'_t \in \mathcal{A}$. Practically, this changes several things for the above-mentioned theory: in action selection through exploration or exploitation $\mathcal{A} \rightarrow \mathcal{A}'_t$; in (2) and (4) $\mathcal{A} \rightarrow \mathcal{A}'_{t+1}$; and to enable execution of these equations, experiences should be stored as 5-dimensional arrays $(s_t, a_t, r_t, s_{t+1}, \mathcal{A}'_{t+1})$.

B. Problem Formulation

In our energy arbitrage problem, the CDQN agent should learn to maximize profit by observing the ESS and market environment, controlling the actions of the ESS to bid in the market environment, and by receiving reward signals in the form of the profit/loss resulting from the taken actions. We use this subsection to formulate the energy arbitrage problem as an MDP by characterizing the state space, action space and reward function. The problem formulation is generic and could be applied to any ESS with fixed energy capacity, power rating and (dis)charging efficiencies.

1) Assumptions: The energy market is assumed to be perfectly competitive. This implies that market participants are price-takers and that the market is sufficiently liquid. Furthermore, 24 hour-ahead price forecasts at different level of accuracy are assumed to be given as input to the CDQN agent. In section III-A we show how the market price forecasts are generated.

2) State Space: We define a continuous state space where the state at day d, hour t, is defined as:

$$s_{d,t} = (p^{f'}_{d,t}, c_t, SoC_{d,t}), \quad t \in \mathcal{T}^d, d \in \mathcal{D}. \tag{5}$$

where $\mathcal{D}$ and $\mathcal{T}^d$ are the sets of days and hours in a day respectively. $p^{f'}_{d,t} = (p^{f'}_{d,t=0}, p^{f'}_{d,t=1}, \ldots, p^{f'}_{d,t=22}, p^{f'}_{d,t=23})$ are the generated (imperfect) and scaled hourly DA market clearing price (MCP) forecasts for the next day d. First we limit the generated imperfect MCP forecasts by taking $p^f_{d,t} = min(p^f_{d,t}, 150)$ and $p^f_{d,t} = max(p^f_{d,t}, -50)$. Then we apply min-max scaling and obtain $p^{f'}_{d,t} \in [0, 1]$. $SoC_{d,t} \in [0, 1]$ is the SoC of the ESS at day d and the start of hour t. c_t is the hour counter that has been developed for the algorithm to extrapolate which of the 24 forecasted prices belongs to the state's bidding hour. The combination of the hour counter and price forecasts with a fixed 24-hour time horizon as inputs, enable the agent to find the optimal bidding strategy using the most accurate available price forecasts at each time instant.

3) Action Space: Next, we define a discrete action space:

$$\mathcal{A} = \{\overline{P^C}, 0, -\overline{P^D}\} \tag{6}$$

where $\overline{P^C}$ and $\overline{P^D}$ are the net maximum charging and discharging power rating of the ESS respectively. Taking the action with value '0' means for the ESS to stand by. After the CDQN algorithm determines an action $a_{d,t} \in \mathcal{A}$ for the full duration of hour t, the SoC reads as:

$$SoC_{d,t+1} = SoC_{d,t} + \frac{a_{d,t}}{\overline{E} - \underline{E}} \tag{7}$$

where $\overline{E}$ and $\underline{E}$ are the maximum and minimum energy capacity of the ESS respectively. The choice of $a_{d,t}$ is limited to the physical boundary of the ESS: $SoC \in [0, 1]$. Therefore:

$$-SoC_{d,t} \cdot (\overline{E} - \underline{E}) \leq a_{d,t} \leq (1 - SoC_{d,t}) \cdot (\overline{E} - \underline{E}) \tag{8}$$

Because our action space is discretized, this constraint might cause limitation of the SoC to a smaller range in practice. To resolve this problem, we assume $\overline{P^C} = \overline{P^D}$ in (6).

4) Reward Function: We define the reward function as a scaled version of the profit/loss made from a single action:

$$r_{d,t} = -\eta_{d,t} \cdot a^{'}_{d,t} \cdot p^{r'}_{d,t} \tag{9}$$

where

$$a^{'}_{d,t} = \begin{Bmatrix} \frac{a_{d,t}}{|a_{d,t}|}, & a_{d,t} & \neq 0 \\ a_{d,t}, & a_{d,t} & = 0 \end{Bmatrix} \tag{10a}$$

$$\eta_{d,t} = \begin{Bmatrix} \frac{1}{\eta^C_{d,t}}, & a_{d,t} & > 0 \\ \eta^D_{d,t}, & a_{d,t} & < 0 \end{Bmatrix}. \tag{10b}$$

$\eta^C_{d,t}$ and $\eta^D_{d,t}$ are the charging and discharging efficiency of the ESS at day d, hour t, respectively. $p^{r'}_{d,t}$ denotes the realized

market price on day d, hour t, scaled along the same method as $p^{f'}_{d,t}$. From the previously described scaling operations follows $r_{d,t} \in [-\frac{1}{\eta^C_{d,t}}, \eta^D_{d,t}]$. The reward being in this range should prevent problems with saturation and inefficient learning as described in [24], while simultaneously preserve a proportional reward allocation with respect to the charging and discharging actions. Hereby the cumulative reward is mathematically proportional to the profit/loss made from cumulative actions, thereby training the agent to learn the optimal bidding strategy.

Algorithm 1 depicts pseudocode of the CDQN architecture applied to the described energy arbitrage problem.

Algorithm 1 CDQN for Energy Storage Arbitrage in DA Market

Set Hyperparameters $\mathcal{E}, \mathcal{H}^e, D, X, \gamma, \alpha, MAPE, U, M$
Load Realized DA Market Price Data p^r
Compute the complete set of p^f from (13)
Initialize ESS Environment $\overline{E}, \underline{E}, \overline{P}, \underline{P}, \eta^C, \eta^D$
Initialize CDQN algorithm ω, ω^-, ERM
for $e \in \mathcal{E}$ **do**
  Compute $\epsilon = D^e$
  Draw $d = rand.choice(\mathcal{D})$
  Set $SoC_{d,0} = 0$
  Compute $s_{d,0}$ and $\mathcal{A}'_{d,0}$
  for $h \in \mathcal{H}^e$ **do**
    $t = h$
    Draw $r = rand.uniform(0, 1)$
    if $r > \epsilon$ **then**
      $a_{d,t} = \arg\max_{\mathcal{A}'_{d,t}} Q(s_{d,t}, \mathcal{A}'_{d,t})$
    else
      $a_{d,t} = rand.choice(\mathcal{A}'_{d,t})$
    end if
    Execute $a_{d,t}$
    Compute $r_{d,t}$ from (9) and (10)
    Compute $s_{d,t+1}$ from (5) and (7)
    Compute $\mathcal{A}'_{d,t+1}$ from (6) and (8)
    Store experience in ERM
    Draw X experiences from ERM
    for $x = 1 : X$ **do**
      Update the Q-value through (2)
    end for
    Update the CDQN through (4)
  end for
  if $e \mod U = 0$ **then**
    Update target network $w^- = w$
  end if
end for

III. Numerical Results

In this section we first explain the DA MCP forecast data preparation step. Second, we give a description of the case study, parameter settings and performance evaluation measures. Next, we establish the optimal algorithmic implementation and analyze the CDQN's performance for several ESS configurations. Finally, we analyze the CDQN's performance through applying a global sensitivity analysis on the forecast uncertainty and comparing the results to a benchmark optimization algorithm.

A. Data Preparation: Imperfect Price Forecast Generation

The imperfect DA MCP forecasts are generated by adding an error term to the realized historical DA MCP data. A popular way of evaluating the performance of a price forecasting tool is by using the Mean Absolute Percentage Error (MAPE) measure [25]:

$$MAPE_{out} = \frac{100\%}{n(\mathcal{D}) \cdot n(\mathcal{T}^d)} \cdot \sum_{d \in \mathcal{D}} \sum_{t \in \mathcal{T}^d} \frac{|p^r_{d,t} - p^f_{d,t}|}{p^r_{d,t}} \quad (11)$$

where $n()$ denotes the number of elements in a set. Representing an unspecified forecasting algorithm, we aim to generate hourly DA MCP forecasts at a desired accuracy evaluated by the MAPE. To do so, we assume that forecast error is represented by addition of a Gaussian noise signal to the historical hourly MCPs, that is $p^f_{d,t} = p^r_{d,t} + \mathcal{N}(0, \sigma)_{d,t}$. We use this relation to substitute $p^f_{d,t}$ in (11) and subsequently rewrite the equation to solve for $\mathcal{N}(0, \sigma)_{d,t}$. Substituting this solution in $p^f_{d,t} = p^r_{d,t} + \mathcal{N}(0, \sigma)_{d,t}$ results in:

$$p^f_{d,t} = p^r_{d,t} + p^r_{d,t} \cdot \frac{MAPE_{in}}{100\%} \cdot \mathcal{N}(0, \sqrt{\frac{\pi}{2}})_{d,t} \quad (12)$$

where $MAPE_{in} = MAPE_{out}$ for $n(\mathcal{D}) \to \infty$. In (12) the error term includes multiplication with the historical DA MCP itself causing unrealistic noise approximations when the historical DA MCP takes a near zero value. Arguably, it would be more realistic to uniformly distribute the added noise based on the average realized daily DA MCP. Therefore, we alter (12) to:

$$p^f_{d,t} = p^r_{d,t} + (\frac{1}{n(\mathcal{T}^d)} \sum_{t \in \mathcal{T}^d} p^r_{d,t}) \cdot \frac{MAPE_{in}}{100\%} \cdot \mathcal{N}(0, \sqrt{\frac{\pi}{2}})_{d,t} \quad (13)$$

This alteration causes a slight difference between the intended MAPE output measure and the MAPE input value: $MAPE_{out} = (1 \pm 0.1) \cdot MAPE_{in}$ for $n(\mathcal{D}) \to \infty$.

B. Case Study, Parameter Settings & Performance Evaluation

As a case study we study the 'Green Battery' (GB), a type of acid-base flow battery developed by the AquaBattery company [26]. In flow batteries, energy capacity and power rating can be independently scaled. Also, there are no standby power losses since the liquids are stored in separate storage tanks [27]. In a standard GB power module $\overline{P^D} = 0.3\ MW$. Due to a lack of related time-dependent data and/or models on the GB, we assume time-independent charging and discharging efficiency terms, where $\eta^C_{d,t} = 0.9$ and $\eta^D_{d,t} = 0.8$, as communicated by Aquabattery. We formulated our problem so that time dependent efficiency terms can be incorporated when data or model availability allows for it, as suggested in [13].

Without performing extensive tuning, the hyperparameters have been set as presented in Table I. Agents are trained for 10,000 episodes each consisting of a single bidding day. Most notable is that the discount factor has a high value such that later states heavily influence the action made in the current state, enabling determination of the optimal bidding strategy for a block of 24 consecutive bidding hours.

TABLE I: Summary of CDQN settings

Item	Value
No. nodes input layer	$25 + n(c_t)$
Type hidden layers	$Dense$
No. hidden layers	2
No. nodes hidden layers	64
No. nodes output layer	$n(\mathcal{A})$
Activation function	$ReLu$
Optimizer	$Adam$
Learning rate (α)	0.00025
Epsilon Decay Factor (D)	0.99953
Batch size (X)	64
Experience Replay Memory Size (M)	$240,000$
Target Network Update Frequency (U)	100
Set of Episodes ($\mathcal{E}$)	$0:10,000$
Set of steps within Episode ($\mathcal{H}^e = \mathcal{T}^d$)	$0:23$

The proposed approach is evaluated using historical French DA market clearing price data of 2019 and 2020 for training and testing respectively [28]. The CDQN agent is expected to determine the optimal bidding strategy using imperfect DA MCP forecasts obtained as is explained in Section III-A above. Therefore, the DA MCP forecasts will be generated with a MAPE value that is equal in both training and testing phase. Literature on DA MCP forecasting tools show that MAPE-values average around 5% [29]–[33], which we take as our initial MAPE-value. We measure the performance of an agent through the yearly accumulated profit from bidding in the 2020 French DA market and average the results from 10 consecutive runs with equal settings to reduce the impact of randomness in the CDQN training process. The algorithm is developed using python 3.8 and Keras [34] with TensorFlow back-end [35]. Training a single agent takes about 30 minutes while solving the bidding problem for a whole year takes a only a few seconds when executed on an 8-core CPU (2.9 GHz) and a 1024-core GPU (1.35 GHz).

C. Establishment of Algorithmic Implementation

In this subsection we briefly describe experiments to establish the hour counter and action space size implementation

1) Hour Counter Methodology: In establishing the state space observation of the CDQN agent, we have tested three hour counter types to represent $\mathcal{T}^d$. The Index hour counter: an array of length 24, where each number takes a value of 0 except for the number with its index equal to bidding hour t, which takes a value of 1. The Binary hour counter: an array of length 5 where the bidding hour t is expressed as a binary number. The Fraction hour counter: an array of length 1, where the bidding hour t is expressed as a fraction of $n(\mathcal{T}^d)$. Our analysis showed that the CDQN agent presented with the Index hour counter could accumulate most yearly profit. Therefore, this hour counter has been adopted for the upcoming experiments.

2) Action Space Size: As a second experiment, we test the influence of the action space size on the performance of the CDQN agent. We set $n(\mathcal{A}) = \{3, 5, 9, 15\}$ resulting in e.g. $\mathcal{A} = \{\overline{P^C}, 1/2\overline{P^C}, 0, -1/2\overline{P^D}, -\overline{P^D}\}$ for $n(\mathcal{A}) = 5$. Analysis of the results showed a small decrease in performance for increased action space size, and therefore we set $n(\mathcal{A}) = 3$ in further experiments.

D. Establishment of Algorithmic Performance

Next, we establish performance of the proposed CDQN algorithm by application of a global sensitivity analysis on the ESS power to energy (PE) ratio configuration. Additionally, this analysis gives insight in investment decisions regarding the optimal PE ratio for a potential investor. We set $\overline{E} = 3.6\ MWh$, $\underline{E} = 0\ MWh$ and $\overline{P^C} = \overline{P^D} = \{0.3, 0.6, 1.8, 3.6\}\ MW$ resulting in PE-ratios of 1:12, 1:6, 1:2 and 1:1 respectively. In Fig. 1 we observe the accumulated profit to be positively correlated with the PE-ratio. This is because a higher power rating enables a higher volume of electricity trade. The correlation is sublinear since ESSs with high PE-ratio configurations are limited in their number of consecutive (dis)charging actions, while price peaks and valleys often last for multiple consecutive hours.

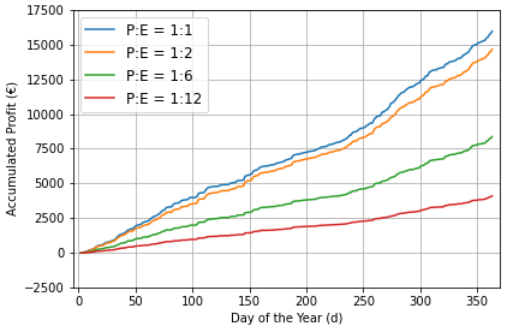


Fig. 1: Yearly average accumulated profit from energy arbitrage for the CDQN algorithm applied to different ESS power to energy ratio configurations.

E. Robustness to Price Forecast Error

The next step in our research is to put the performance of the CDQN algorithm into perspective by comparing to a benchmark algorithm. Therefore, we reformulate the optimization problem as a MILP with objective function:

$$\sum_{t=0}^{23} (\eta_t^D D_t \overline{P^D} - \frac{1}{\eta_t^C} C_t \overline{P^C}) * p_t^f \tag{14}$$

and constraints:

$$\begin{cases} SoC_{t+1} = SoC_t - D_t\overline{P^D} + C_t\overline{P^C} \\ 0 \leq SoC_t \leq 1 \\ SoC_0 = SoC_{23} = 0 \\ D_t, C_t \in \{0,1\} \\ 0 \leq D_t + C_t \leq 1 \end{cases} \tag{15}$$

where D_t and C_t are binary decision variables that are set to 1 when respectively discharging and charging, and 0 otherwise. The last constraint prevents simultaneous charging and discharging, while it leaves the option to do neither. We then solve this optimization problem using CPLEX [36] in Python for each day of the year.

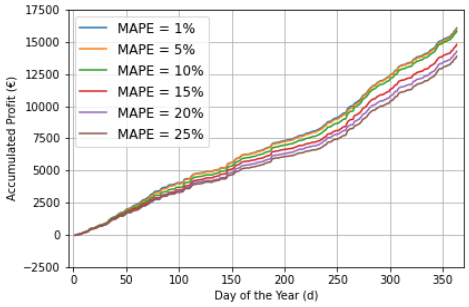
In addition we aim to examine the added value of the capabilities of the CDQN to learn how to deal with forecast error in comparison to the deterministic MILP algorithm. DA MCP forecast errors range between 2-25% MAPE for different tools [29]–[33]. Therefore, we apply a global sensitivity analysis on the forecast error where $MAPE = \{1, 5, 10, 15, 20, 25\}$ %. To examine whether the CDQN algorithm really learns to deal with forecast error, we distinguish a CDQN algorithm that is trained on the respective MAPE values that it is also tested against, and a CDQN algorithm that is trained on perfect price forecasts. We compare the performance of these two agents and the MILP algorithm and display the results of the P:E = 1:1 cases in Fig. 2.

From Figure 2a–c one observes a negative correlation between the performance and MAPE value. This is expected, since a higher MAPE hinders the informative information embedded in the DA forecast and therefore, complicates the decision-making process.

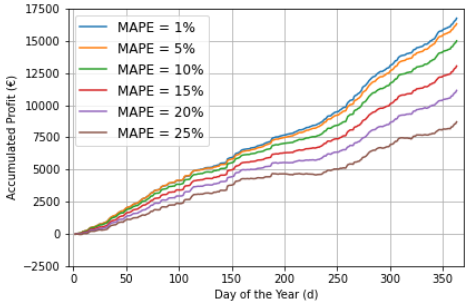
Comparing Figure 2a to Figure 2c, one also observes that the reference algorithm outperforms the DQN so long as $MAPE \leq 5\%$. However, for $MAPE \geq 10\%$, the DQN is superior to the MILP based reference algorithm. If we were to provide perfect DA MCP forecasts, the MILP algorithm would ensure maximization of profits. Since CDQN is learning-based, it would never solve with 100% accuracy. This explains why the MILP algorithm performs better in the scenarios with low forecast error. The fast decline in performance of the MILP algorithm is caused by the algorithm making a considerable number of bids that would, under the assumed perfect price forecasts, only result in a small extra profit. Increasing the MAPE value, increases the probability of these bids resulting in losses, and therefore, performance of the MILP algorithm decreases, even leading to a net yearly loss in the most severe cases.

Now comparing Figure 2b to Figure 2c where both algorithms use perfect price forecasts, one observes the CDQN outperforms the MILP algorithm for $MAPE \geq 10\%$. We argue that the CDQN tends to converge to a more conservative decision-maker, resulting in strategy in which the ESS charges only when DA MCPs are very low and discharges only when DA MCPs are very high.

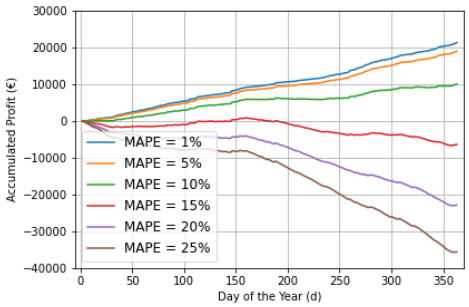
Finally, from Figure 2a and Figure 2b one observes that the CDQN algorithm performs better when trained using



(a) DQN trained with price forecast error



(b) DQN trained without price forecast error



(c) Reference Algorithm

Fig. 2: Global sensitivity analysis on price forecast error for 3 different algorithmic implementations: a) CDQN algorithm trained and tested on imperfect price forecasts with equal error terms, b) CQN algorithm trained on perfect price forecasts, while tested on imperfect price forecasts, and c) MILP algorithm tested on imperfect price forecasts.

erroneous price forecasts. This result implies that the CDQN algorithm has the ability to learn the error pattern in the provided price forecasts and subsequently uses that knowledge to find the optimal bidding strategy in a comparable scenario.

IV. Conclusion

In this paper we developed a novel DRL-based algorithm for energy storage arbitrage in the DA electricity market. The resulted CDQN algorithm takes a reasonably accurate and realistic 24-hour price forecast as input. The proposed algorithm is generic in that it can be applied to any ESS technology with fixed power and efficiency rating. Through global sensitivity analysis, we showed that the CDQN algorithm retrieves a net

profit from energy arbitrage for each of the tested forecast error cases. We also find that the CDQN algorithm performs near-optimal for low market price forecast errors and outperforms a benchmark optimization algorithm for higher errors. This demonstrates the added value of DQN-based algorithms in the problem set-up characterized by market environments with high price forecast errors. Ultimately, utilizing the proposed algorithms favorably serves the ESS, the energy market and grid participation, by offering more flexibility to the system despite the uncertain price forecasts provided as input to it.

The RT and ID markets are characterized by relatively high price volatility and forecast complexity. This paper showed that the proposed DQN algorithm is especially good at containing the impact(s) of imperfect price forecast. Therefore, a valuable future work might focus on expanding the energy arbitrage problem formulation to also include the RT and/or ID markets with the aim of determining a potentially more interesting business case for potential ESS investors.

REFERENCES

[1] J. Wiseman, "The great energy transition of the 21st century: The 2050 zero-carbon world oration," *Energy research & social science*, vol. 35, pp. 227–232, 2018.

[2] D. Azari, S. S. Torbaghan, H. Cappon, K. J. Keesman, H. Rijnaarts, and M. Gibescu, "Exploring the impact of data uncertainty on the performance of a demand response program," *Sustainable Energy, Grids and Networks*, vol. 20, p. 100262, 2019.

[3] S. S. Torbaghan, G. Suryanarayana, H. Höschle, R. D'hulst, F. Geth, C. Caerts, and D. Van Hertem, "Optimal flexibility dispatch problem using second-order cone relaxation of ac power flows," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 98–108, 2019.

[4] D. Azari, S. S. Torbaghan, M. Gibescu, and M. A. Van Der Meijden, "The impact of energy storage on long term transmission planning in the north sea region," in *2014 North American Power Symposium (NAPS)*. IEEE, 2014, pp. 1–6.

[5] W. B. Powell, "From reinforcement learning to optimal control: A unified framework for sequential decisions," in *Handbook of Reinforcement Learning and Control*. Springer, 2021, pp. 29–74.

[6] 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, 2018.

[7] D. Krishnamurthy, C. Uckun, Z. Zhou, P. R. Thimmapuram, and A. Botterud, "Energy storage arbitrage under day-ahead and real-time price uncertainty," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 84–93, 2017.

[8] A. Akbari-Dibavar, K. Zare, and S. Nojavan, "A hybrid stochastic-robust optimization approach for energy storage arbitrage in day-ahead and real-time markets," *Sustainable Cities and Society*, vol. 49, p. 101600, 2019.

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

[10] H. Xu, X. Li, X. Zhang, and J. Zhang, "Arbitrage of energy storage in electricity markets with deep reinforcement learning," *arXiv preprint arXiv:1904.12232*, 2019.

[11] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3, pp. 279–292, 1992.

[12] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[13] J. Cao, D. Harrold, Z. Fan, T. Morstyn, D. Healey, and K. Li, "Deep reinforcement learning-based energy storage arbitrage with accurate lithium-ion battery degradation model," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4513–4521, 2020.

[14] 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.

[15] D. J. Harrold, J. Cao, and Z. Fan, "Data-driven battery operation for energy arbitrage using rainbow deep reinforcement learning," *Energy*, vol. 238, p. 121958, 2022.

[16] G. Han, S. Lee, J. Lee, K. Lee, and J. Bae, "Deep-learning-and reinforcement-learning-based profitable strategy of a grid-level energy storage system for the smart grid," *Journal of Energy Storage*, vol. 41, p. 102868, 2021.

[17] Y. Miao, T. Chen, S. Bu, H. Liang, and Z. Han, "Co-optimizing battery storage for energy arbitrage and frequency regulation in real-time markets using deep reinforcement learning," *Energies*, vol. 14, no. 24, p. 8365, 2021.

[18] 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.

[19] M. N. Akhter, S. Mekhilef, H. Mokhlis, and N. Mohamed Shah, "Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques," *IET Renewable Power Generation*, vol. 13, no. 7, pp. 1009–1023, 2019.

[20] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[21] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.

[22] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[23] G. Kalweit, M. Huegle, M. Werling, and J. Boedecker, "Deep constrained q-learning," *arXiv preprint arXiv:2003.09398*, 2020.

[24] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[25] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *International journal of forecasting*, vol. 30, no. 4, pp. 1030–1081, 2014.

[26] R. Pärnamäe, L. Gurreri, J. Post, W. J. van Egmond, A. Culcasi, M. Saakes, J. Cen, E. Goosen, A. Tamburini, D. A. Vermaas *et al.*, "The acid–base flow battery: Sustainable energy storage via reversible water dissociation with bipolar membranes," *Membranes*, vol. 10, no. 12, p. 409, 2020.

[27] N. Tokuda, T. Kumamoto, T. Shigematsu, H. Deguchi, T. Ito, N. Yoshikawa, and T. Hara, "Development of a redox flow battery system," *SUMITOMO ELECTRIC TECHNICAL REVIEW-ENGLISH EDITION-*, pp. 88–94, 1998.

[28] ewoken, "Epex spot data," dataset available from https://ewoken.github.io/epex-spot-data/. [Online]. Available: https://github.com/ewoken/epex-spot-data

[29] N. Amjady and F. Keynia, "Day ahead price forecasting of electricity markets by a mixed data model and hybrid forecast method," *International Journal of Electrical Power & Energy Systems*, vol. 30, no. 9, pp. 533–546, 2008.

[30] S. K. Aggarwal, L. M. Saini, and A. Kumar, "Electricity price forecasting in deregulated markets: A review and evaluation," *International Journal of Electrical Power & Energy Systems*, vol. 31, no. 1, pp. 13–22, 2009.

[31] M. Shafie-Khah, M. P. Moghaddam, and M. Sheikh-El-Eslami, "Price forecasting of day-ahead electricity markets using a hybrid forecast method," *Energy Conversion and Management*, vol. 52, no. 5, pp. 2165–2169, 2011.

[32] I. P. Panapakidis and A. S. Dagoumas, "Day-ahead electricity price forecasting via the application of artificial neural network based models," *Applied Energy*, vol. 172, pp. 132–151, 2016.

[33] R. Angamuthu Chinnathambi, A. Mukherjee, M. Campion, H. Salehfar, T. M. Hansen, J. Lin, and P. Ranganathan, "A multi-stage price forecasting model for day-ahead electricity markets," *Forecasting*, vol. 1, no. 1, pp. 26–46, 2018.

[34] F. Chollet *et al.* (2015) Keras. [Online]. Available: https://github.com/fchollet/keras

[35] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[36] I. I. Cplex, "V12. 1: User's manual for cplex," *International Business Machines Corporation*, vol. 46, no. 53, p. 157, 2009.