

# ENHANCING BATTERY STORAGE ENERGY ARBITRAGE WITH DEEP REINFORCEMENT LEARNING AND TIME-SERIES FORECASTING

Manuel Sage<sup>1</sup>, Joshua Campbell<sup>1</sup>, Yaoyao Fiona Zhao<sup>1,\*</sup>,

<sup>1</sup>McGill University, Montreal, Canada

## ABSTRACT

Energy arbitrage is one of the most profitable sources of income for battery operators, generating revenues by buying and selling electricity at different prices. Forecasting these revenues is challenging due to the inherent uncertainty of electricity prices. Deep reinforcement learning (DRL) emerged in recent years as a promising tool, able to cope with uncertainty by training on large quantities of historical data. *However, without access to future electricity prices, DRL agents can only react to the currently observed price and not learn to plan battery dispatch. Therefore, in this study, we combine DRL with time-series forecasting methods from deep learning to enhance the performance on energy arbitrage.* We conduct a case study using price data from Alberta, Canada that is characterized by irregular price spikes and highly non-stationary. This data is challenging to forecast even when state-of-the-art deep learning models consisting of convolutional layers, recurrent layers, and attention modules are deployed. Our results show that energy arbitrage with DRL-enabled battery control still significantly benefits from these imperfect predictions, but only if predictors for several horizons are combined. Grouping multiple predictions for the next 24-hour window, accumulated rewards increased by 60% for deep Q-networks (DQN) compared to the experiments without forecasts. We hypothesize that multiple predictors, despite their imperfections, convey useful information regarding the future development of electricity prices through a “majority vote” principle, enabling the DRL agent to learn more profitable control policies.

**Keywords:** reinforcement learning, deep learning, time-series forecasting, battery energy storage, energy arbitrage

## 1. INTRODUCTION

### 1.1 Background

Energy arbitrage (EA) describes the practice of buying electricity when prices are low and selling it when prices are high. When battery energy storages (BESs) are involved, charging and discharging occur at low and high power prices, respectively [1, 2]. The purpose of EA with BESs is to reduce costs or increase revenues for the battery operators. The growing incorporation of

intermittent renewable energy into electricity grids comes with an increasing demand for flexible energy storages. At the same time, intermittent renewables can increase price volatility [3, 4]. This favors EA with batteries, which has become the largest profit opportunity for BES operators according to Ref. [1].

Both planning new BES projects and controlling existing BESs are challenging due to the stochastic nature of electricity prices. A control strategy is required that dispatches the BES accordingly to maximize performance. Traditional tools for optimizing BES dispatch include programming methods such as mixed-integer linear programming [5, 6]. These white-box approaches can compute optimal solutions over small time horizons, but quickly become intractable for larger problems and time spans. Besides, they also require access to the dynamics of the environment and future states [7, 8]. Thus, the obtained solutions represent the ideal case of perfect future knowledge and might not be achievable in practice. Another popular category of optimizers for EA with batteries are heuristic methods, such as genetic algorithms and similar evolutionary approaches [9, 10]. Heuristic methods only require access to a sample model which facilitates the implementation of non-linearities such as battery degradation or charging efficiencies. However, heuristic methods suffer from poor convergence properties and sample inefficiency, which causes high compute cost [7, 11]. Like white-box approaches, applications of heuristic models are limited to shorter time periods, affecting their ability to handle uncertainty.

Reinforcement learning (RL) has recently gathered interest in the energy systems domain as an alternative. RL is a subset of machine learning characterized by an agent that learns by interacting with its environment. By training on large amounts of historical data, RL can learn control policies while considering uncertainty. Recent advances in deep reinforcement learning (DRL), which combines RL frameworks with deep learning, have led to more powerful, stable, and sample-efficient algorithms [12, 13]. RL only requires a sample model and no future information about uncertain variables. However, in the case of EA, providing the agent with information about future electricity prices might be beneficial. In this study, we therefore evaluate the combination of two deep learning methods: deep reinforcement learning and time-series forecasting with deep neural networks.

\*Corresponding author: yaoyao.zhao@mcgill.ca

Our study centers around the possible performance gains of this combination for energy arbitrage when dealing with messy, real-world data. We conduct a case study in Alberta, Canada that is characterized by great electricity price fluctuations and the absence of cyclic patterns. Comparing different predictors and two DRL algorithms, namely deep Q-networks (DQN) and proximal policy optimization (PPO), we show that time-series forecasting can significantly boost RL performance despite high forecasting errors. Our results indicate that performance highly depends on the forecasted horizon and the number of forecasters available, with multiple forecasters leading to better performances.

## 1.2 Related Work

Historically, statistical, non-machine learning models such as autoregressive integrated moving average (ARIMA) techniques have been widely used for time-series forecasting [14, 15]. Reference [15] used statistical and microeconomic models to predict price spikes in the Alberta electricity market from 2002-2015. However, in recent years deep learning methods have become popular. Neural networks with recurrent and convolutional architectures have demonstrated superiority over simpler statistical methods [16–18]. For example, Ref. [19] showed that a properly tuned Long Short-Term Memory (LSTM) model, a type of recurrent network, performed best when predicting hour-ahead and day-ahead electricity prices in New South Wales compared to other time-series forecasting techniques. More recently, Ref. [20] used LSTM models combined with an attention mechanism to achieve a mean absolute error (MAE) reduction of 7.09% when forecasting electricity prices in the Denmark compared to the standard LSTM implementation. Reference [21] proposed a Convolutional Neural Network (CNN) combined with attention-based LSTM layers to form a hybrid model. This architecture achieved a 2.20% improvement in normalized root-mean-squared error (RMSE) when predicting photovoltaic power one hour into the future compared to a regular LSTM model.

Typically, for time-series forecasting of environmental and energy variables, many input time-series are used to provide the models with additional information. This is balanced with the amount of available data, the quality of the data, and the need to prevent dilution. For example, Ref. [14] performed extensive feature selection for predicting day-ahead electricity price and concluded that hourly price, demand, wind power generation, wind speed, and ambient temperature were the most important factors for their case study of the Iberian electricity market and gave their models the best performance. Real-world electricity price data can be irregular which makes time-series forecasting more inherently difficult. Reference [22] showed that smoothing the data as a preprocessing step can improve model performance when forecasting variables in the energy sector.

In the related work applying RL to EA with batteries, EA is either regarded as the sole task of the battery [1, 2, 23], or combined with other battery services such as frequency regulation, demand response, load following, and improved utilization of renewable energies [24–27]. Reference [2] conducted a simple study in which tabular Q-learning is applied to pure EA. Here, the RL agent makes decisions solely based on the current electricity price and the state of charge (SOC) of the battery. A similar

approach, also with tabular Q-learning, was studied in Ref. [23].

Reference [27] compared tabular Q-learning, Q-learning with linear function approximation, and Sarsa to particle swarm optimization on a combined frequency regulation and EA task. In the conducted case study, Q-learning with function approximation performed best. Reference [26] applied deep deterministic policy gradients (DDPG) to a joint load following and EA environment, where EA helps to reduce the cost of power supply. DDPG scored only slightly worse compared to an oracle based on MPC with access to perfect future information, but only required a fraction of the compute cost. Reference [24] compared several DRL models including PPO, DDPG, and double DQN on a dispatch task combining frequency regulation, improved renewable energy utilization, and EA. This comparison is interesting as a value-based RL algorithm with discrete action choices (double DQN) is compared to three actor-critic methods with continuous actions. In the experiments, PPO performed best regarding both reward maximization and sample efficiency.

In a few studies, RL has been deployed along with forecasts of uncertain variables to improve the performance on EA. Reference [23] used LSTM cells to forecast electricity prices and demands for the next hour, which were then used in an EA environment controlled by tabular Q-learning. Reference [1] predicted the next 24 hours of electricity prices using a CNN-LSTM hybrid given the last 168 hours. The forecasts were then added to the state space of a noisy-net DQN model on a pure EA task. Reference [25] compared various variants of DQN to DDPG on a task combining EA, load following, and renewable energy control. The authors used artificial neural networks to forecast the next hour's values for solar and wind power generation, demand, and electricity price. On the best performing model, rainbow DQN, these forecasts helped to increase rewards by 14%.

## 1.3 Contributions

Our study differs from the existing work by its explicit focus on the possible performance gains on EA when combining time-series forecasting and DRL. We carefully benchmark different predictors, RL models, and forecasting horizons. Unlike the reviewed work, we conduct a case study on challenging electricity price data that is non-stationary and lacks obvious cyclic behavior. The data in Ref. [1, 23, 25] shows clear patterns facilitating predictions. For example, the next hour price forecaster in Ref. [25] achieved a low mean absolute percentage error (MAPE) of 11.6%, compared to 28.2% in our case study (see section 5.1). The contributions of this study can be summarized as follows:

1. We formulate an EA task for a grid-connected BESs considering charge and discharge efficiencies as well as battery degradation. To this environment, we apply a combined deep learning framework consisting of DRL for battery control and time-series forecasting for predictions on future electricity prices.
2. Conducting a case study on challenging price data from Alberta, Canada, we investigate how – despite high forecasting errors – DRL models can improve decision making when provided with price predictions. For this purpose, we

benchmark different forecasting architectures, DRL models, and forecasting horizons.

## 2. PROBLEM FORMULATION

In a recent study currently under review, we have compared the performance of various DRL algorithms and experiment design choices on the same system [28]. In the present study, we extend this methodology with time-series forecasts and assess the effect on DRL performance.

### 2.1 System Description

We define a simple environment of a grid connected BES aiming to capitalize on the volatility of electricity prices. This setup, where performance largely depends on electricity prices, is most suitable to identify the influence of time-series forecasting on RL performance. We assume that the battery is a price-taker that can purchase and sell electricity at market prices without influencing prices. The objective is to maximize the revenues of energy arbitrage over the optimization period:

$$\max R_{Total} = \sum_{t=0}^T R_{grid,t} - C_{degr,t} \quad (1)$$

subject to

$$R_{grid,t} = c_{w,t} \times P_{B,t} \times \Delta t \quad (2)$$

$$SOC_t = SOC_{t-1}(1 - \sigma) - \eta \frac{P_{B,t} \Delta t}{C_{max}} \quad (3)$$

$$C_{degr,t} = \frac{|(1 - SOC_t)^{k_p} - (1 - SOC_{t-1})^{k_p}|}{2 \times N_{100}^{fail}} \times C_{inv} \quad (4)$$

$$P_B^{min} \leq P_{B,t} \leq P_B^{max} \quad (5)$$

$$SOC^{min} \leq SOC_t \leq SOC^{max} \quad (6)$$

where

$t$  = time index

$R_{grid}$  = revenues/cost from grid interaction

$C_{degr}$  = cost of BES degradation

$c_w$  = wholesale price of electricity

$P_B$  = BES charging ( $P_B < 0$ ) / discharging power ( $P_B > 0$ )

$SOC$  = BES state of charge

$\sigma$  = BES self-discharge

$\eta$  = BES charge and discharge efficiency

$C_{max}$  = BES capacity

$k_p$  = Peukert constant

$N_{100}^{fail}$  = number of full BES cycles until failure

$C_{inv}$  = investment cost of BES

$P_B^{min}$  = BES charge limit

$P_B^{max}$  = BES discharge limit

$SOC^{min}$  = minimum allowable SOC

$SOC^{max}$  = maximum allowable SOC

The total revenue at each time-step is composed of the revenue for grid interaction and the cost of battery degradation (Eq. 1). The grid revenue is computed with the current time-step's electricity price and charging or discharging power of the battery (Eq. 2). Both grid revenue and battery power are positive when the battery is being discharged and negative when it is charged. We model the BES as a black box whose SOC changes depending on the applied charging or discharging power (Eq. 3).  $\eta$  is the charging or discharging efficiency and  $\eta = \eta_{dc} > 1$  in the case of discharging and  $\eta = \eta_{ch} < 1$  in the case of charging. To model battery degradation (Eq. 4), we resort to a depth of discharge approach for cyclic battery ageing as presented in Ref. [27]. Equations (5) and (6) are constraints limiting the charging/discharging power of the BES and the SOC, respectively.

### 2.2 Markov Decision Process

To apply RL to this environment, we formulate a Markov Decision Process (MDP) consisting of state space  $\mathcal{S}$ , action space  $\mathcal{A}$ , transition function  $\mathcal{P}$ , reward function  $\mathcal{R}$ , and discount factor  $\gamma$ . Together, these elements form the tuple  $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$  [29].

**2.2.1 State Space.** A state  $s_t \in \mathcal{S}$  contains the information available to the RL agent at the current time step. After taking an action, the environment transition to the next state  $s_{t+1} \in \mathcal{S}$  according to the transition function  $\mathcal{P}$ . In the basic experiments, i.e. without price forecasts, each state consists of the current battery SOC and electricity price:

$$s_t = (SOC_t, c_{w,t}) \quad (7)$$

In the experiments with price forecasts, one or more predictions, denoted  $p$ , are added to the state space:

$$s_t = (SOC_t, c_{w,t}, p_{t+1}, \dots, p_{t+24}) \quad (8)$$

**2.2.2 Action Space.** The action space for the EA task is one dimensional and continuous. It reaches from the BES charge limit to the BES discharge limit:  $a_t \in [P_B^{min}, P_B^{max}]$ . To avoid the violation of constraints (5) and (6), we implement a hard-coded safety layer that corrects the RL agent's actions if necessary:

$$a_{c,t} = \begin{cases} \min(a_t, P_B^{max}, \frac{(SOC_t - SOC^{min})C_{max}}{\Delta t}) & , a_t \geq 0 \\ \max(a_t, P_B^{min}, \frac{(SOC_t - SOC^{max})C_{max}}{\Delta t}) & , a_t < 0 \end{cases} \quad (9)$$

where  $a_{c,t}$  is the corrected action. For example, the safety layer prevents the agent from discharging the battery below  $SOC^{min}$ . We have used a similar mechanism in our previous work [30].

**2.2.3 Reward Function.** The instantaneous reward at each time step ( $R_t \in \mathcal{R}$ ) is  $R_t = R_{grid,t} - C_{degr,t}$  (see Eq. 1). The RL agent seeks to maximize the return, which is the sum of discounted rewards:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=t+1}^T \gamma^{k-t-1} R_k \quad (10)$$

where  $\gamma$  is the discount factor and  $\gamma \in [0, 1]$  [29]. We treat  $\gamma$  as a hyperparameter that we tune in our experiments.

### 3. METHODOLOGY

#### 3.1 Reinforcement Learning

In this section we focus on the description of the DQN algorithm [12], which performed best in our experiments. DQN and other value-iteration methods such as Q-learning and SARSA have dominated the reviewed related work and are popular in RL applications to battery dispatch in general [31]. Value-iteration methods iteratively learn state values,  $V(s)$ , that quantify the value of being in a state, or state-action values,  $Q(s, a)$ , that quantify the value of being in a state and taking an action. In Q-learning, the Q-values are updated at each time-step using the Bellman equation:

$$Q(s, a) = Q(s, a) + \alpha[r + \gamma \max_{a'} (Q(s', a')) - Q(s, a)] \quad (11)$$

where  $\alpha$  is the learning rate and  $r$  the immediate reward.  $s'$  and  $a'$  denote the next state and next action, respectively. This recursive equation allows the agent to update a Q-value based on the maximum possible Q-value in the next state [29].

In DQN, the Q-values are learned using neural networks as function approximators and denoted as  $Q(s, a; \theta)$ , where  $\theta$  are the weights of the neural network. The key concepts of DQN are:

- **Epsilon-greedy exploration:** The tradeoff between exploration and exploitation in DQN is managed through the  $\epsilon$ -greedy strategy:

$$a = \begin{cases} \arg \max_a Q(s, a; \theta), & \text{with probability } 1 - \epsilon \\ \text{random action}, & \text{with probability } \epsilon \end{cases} \quad (12)$$

$\epsilon$  is often chosen to encourage exploration early during training and then annealed to favor exploitation. In our experiments, we treat  $\epsilon$  and its behavior during training as hyperparameter that we tune to maximize performance.

- **Experience replay:** DQN stores the experiences made by interacting with the environment in the form of (*state, action, reward, next state*)-tuples in a replay buffer. At training time, a batch of interactions is then sampled from the buffer to compute the update. By decorrelating experiences, this strategy improves stability. At the same time, using a single experience for multiple updates improves sample efficiency.
- **Target network:** DQN keeps a copy of the Q-network, called target network, that is updated periodically with the weights of the Q-network. The target network is used to compute the maximum possible next state-action value in the loss function:

$$L(\theta) = \mathbb{E}_{s, a, r, s'} [(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta))^2] \quad (13)$$

where  $\theta^-$  are the parameters of the target network. The loss is then differentiated with respect to the weights and used for gradient descent. The use of a target network has shown to reduce the overestimation bias and increase stability.

A detailed explanation of these concepts and pseudocode of the DQN algorithm can be found in Ref. [12]. Besides DQN, we also run experiments using PPO, an on-policy, actor-critic algorithm.

For details on this model, the reader is referred to Ref. [13]. DQN requires the discretization of the action space. After conducting preliminary experiments, we decide to divide the action space into three discrete actions:  $a_t \in [P_B^{min}, 0, P_B^{max}]$ , representing maximum charge, idle, and maximum discharge. PPO can handle continuous action spaces and is applied to the unmodified action space introduced in section 2.2.2.

#### 3.2 Time-Series Forecasting and Forecast Integration

To predict electricity prices, we experiment with four different types of deep learning architectures that have demonstrated superior performance in the reviewed related work:

- CNN [32]
- LSTM [33]
- CNN-LSTM hybrids, with convolutional layer(s) being followed by LSTM-layer(s).
- CNN-LSTM hybrids with attention modules [34] after the LSTM layer(s).

All four model types end with one or more fully connected layers. The exact architectures, most prominently model depth, layer width, and activation function, are tuned to maximize performance. For convolutional layers, we tune kernel size and stride. Since we focus on point forecasts, each model is trained to forecast a single future electricity price and has thus one node in the output layer. This increases the number of models that must be tuned and trained compared to training one model to predict every horizon simultaneously, but it guarantees that the best performing model can be obtained for each horizon.

The electricity data in the conducted case study has an hourly resolution. Based on the reviewed related work and preliminary experiments, we decide to train the forecasters to predict electricity prices in 1h, 2h, 3h, 6h, 12h, 18h, and 24h. These forecast horizons were identified to be the most useful for the RL agents (based on experiments supplying the agents with the future ground truth as the forecasts), and they require less computational resources than training every horizon 1-24h. For each of these seven horizons and four model types we conduct a separate hyperparameter tuning. The tuning includes experiments with different input features and window sizes. Besides the electricity price itself, we experiment with adding the electricity demand, ambient temperature, solar irradiance, wind speed, ambient pressure, and relative humidity. We further test for time-related features such as the hour of the day, the week of the year and the month of the year. The sequential data is processed into feature-label pairs using a "sliding window" approach, where the feature array is the combination of the input features for a sequence of data points in the past up to the current point and the label is the future electricity price for the horizon being trained. The length of the feature array is thereby the window size.

As a data preprocessing technique, we normalize all inputs using min-max scaling to a  $[-1, 1]$  range. We also experiment with smoothing the spiky and irregular price data, by converting the raw data to an exponentially weighted moving average of itself, as done in [22]. In addition to the hyperparameters mentioned above,



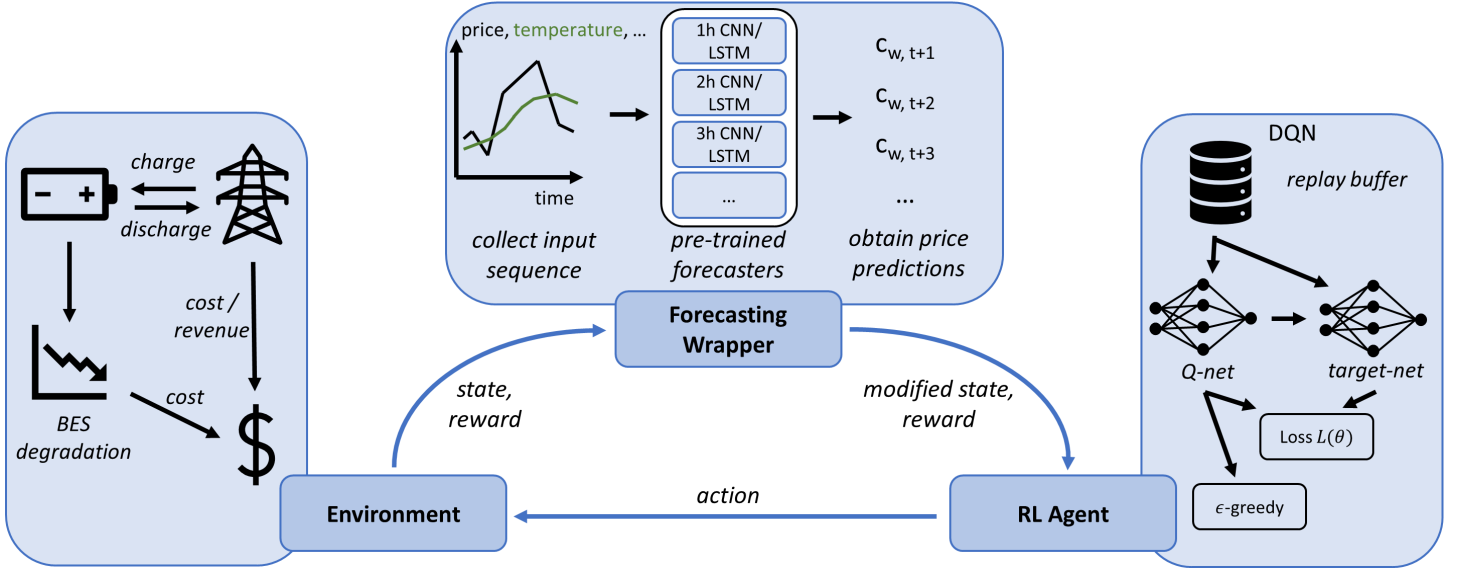


FIGURE 1: Schematic showing the integration of time-series forecasters into the agent-environment framework of reinforcement learning.

TABLE 1: Parameters of the case study

Parameter	Value
battery capacity, $C_{max}$	10 MWh
$SOC^{min} / SOC^{max}$	0.2 / 0.8
$p_B^{min} / p_B^{max}$	-2.5 / 2.5 MW
charge/ discharge eff., $\eta$	0.92 / $\frac{1}{0.92}$
self-discharge, $\sigma$	0
Peukert constant, $k_p$	1.14 [35]
cycles to failure, $N_{100}^{fail}$	6,000 [36]
investment cost, $C_{inv}$	300,000 \$/MWh [37]

we tune batch sizes, learning rates and learning rate schedulers. We use early stopping as a regularization technique, Adam as optimizer, and the RMSE as loss function.

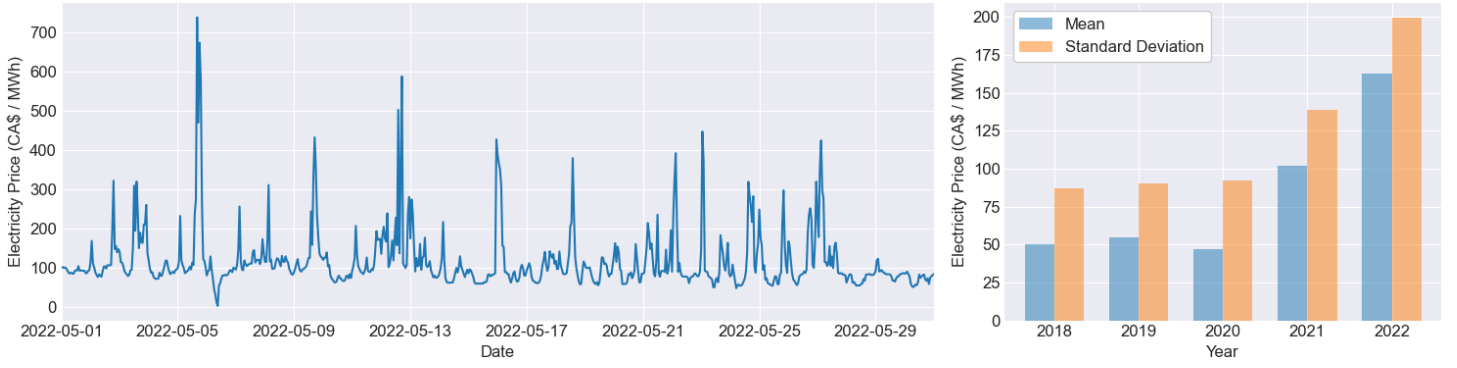
The integration of price forecasts into the RL framework is visualized in Fig. 1. We use a training and validation set composed of data recorded prior to the optimization period of the RL agent. This ensures that the price data encountered by the RL agent was not seen by the forecasters beforehand. Once the best forecasting models over the seven horizons are identified, these are saved and loaded into the forecasting wrapper. The wrapper sits between the environment and the agent, tracks the progress of the environment, and processes the input data for each forecaster. Then, the forecasters make predictions which are added to the original state and passed to the RL agent. Ideally, the RL agent makes better informed decisions with this additional information.

#### 4. CASE STUDY

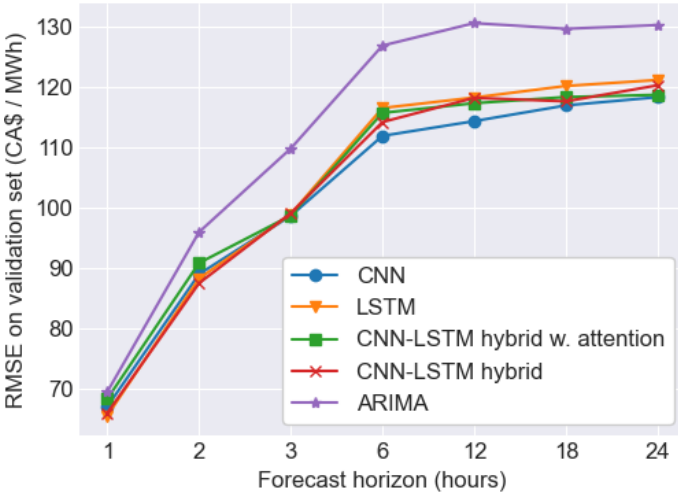
The location for the case study is in Alberta Canada, where the BES is connected to the utility grid and participating in price arbitrage. The parameters characterizing the BES are listed in Table 1 and assume the operation of a Lithium-Ion battery. We obtain electricity data from the Alberta Electric System Operator

[38] and climate data from the ERA5 reanalysis model [39]. Both data types are downloaded in hourly resolution for the years 2018 to 2022. The first 3.5 years from January 2018 to June 2021 form the training set for the price forecasting models. The following six months between July 2021 and December 2022 are the validation set that we use to tune hyperparameters and identify the best forecasters. The year of 2022 serves as test set for the forecasters and optimization time frame for the RL models. For RL, we define the entire year, i.e. 8760 hours, as one episode. Figure 2 provides additional information on the electricity data. The left part of the figure shows the challenging nature of electricity prices in Alberta, taking May 2022 as an example. Price spikes frequently occur, but at different times and with different magnitudes. The bar chart on the right highlights another challenge: the price data is highly non-stationary. Mean and standard deviation of electricity prices were relatively constant from 2018 to 2020. These three years form the majority of the training set. 2021 and especially 2022 show higher mean prices and more volatility, aggravating the forecasting task significantly. Our experiments therefore aim to tackle the challenge of improving energy arbitrage performance with price forecasts despite these challenges.

We utilize a variety of benchmarks for a fair assessment of the tested models. For time-series forecasting, we tune and train an ARIMA model. Additionally, a persistence model, naïvely assuming the forecasted electricity price is identical to the current price, serves as lower bound. Our lower bound for RL is the cross-entropy method (CEM), an evolutionary algorithm popular for continuous control tasks [40]. As upper bound, a genetic algorithm (GA) with access to environment dynamics and perfect future knowledge is used. In a model-predictive control (MPC) framework, the GA optimizes the sequence of battery control actions over a fixed horizon. Then, the first action of the obtained sequence is executed in the environment and the horizon shifts by one time-step in the future. This iterative process with receding horizon is repeated until the end of the episode is reached. Due to



**FIGURE 2: (Left) A one month sample of electricity prices in Alberta from May 2022. (Right) Mean and standard deviation of electricity prices in Alberta for the five years of data collected.**



**FIGURE 3: RMSE of tested models on the validation set for different forecast horizons.**

the full access to the environment dynamics, the MPC-GA model can be seen as an oracle whose performance is only limited by the available compute resources.

The codebase for this case study was written in Python 3.12 and is available along with the dataset at GitHub<sup>1</sup>. PyTorch was used as deep learning library for the time-series forecasters. Gymnasium and stable-baselines3 were used for the RL environment and agents, respectively. Both forecasters and DRL models were tuned using Optuna.

## 5. RESULTS AND DISCUSSION

### 5.1 Forecasting of Electricity Prices

In Fig. 3 we compare the performance on the validation set of the four deep learning models and ARIMA over different horizons after tuning. The persistence model scored significantly worse and was omitted in the figure. The forecasting error for all models increased with increasing forecast horizons. The four deep learning models performed similarly on all horizons, whereas the

ARIMA model received higher errors. For 3h to 24h forecasts, the CNN was found to perform best. For 1h and 2h forecasts, the LSTM and CNN-LSTM hybrid performed best by small margins, respectively. We chose these models for the further experiments with RL. When scored on the test set, their performance metrics ranked from 100 – 175 for RMSE and 28% - 68% for MAPE. These errors are significant when put into perspective with the electricity price statistics in Fig. 2 and the errors reported in the related work [23, 25]. A sample of the predictive performance for selected horizons is provided in the top of Fig. 5. All three horizons plotted struggle with price spikes. 1h predictions lag behind spikes, while 12h and 24h predictions often miss spikes or predict spikes where they don't occur and tend to estimate the magnitude of the spikes incorrectly.

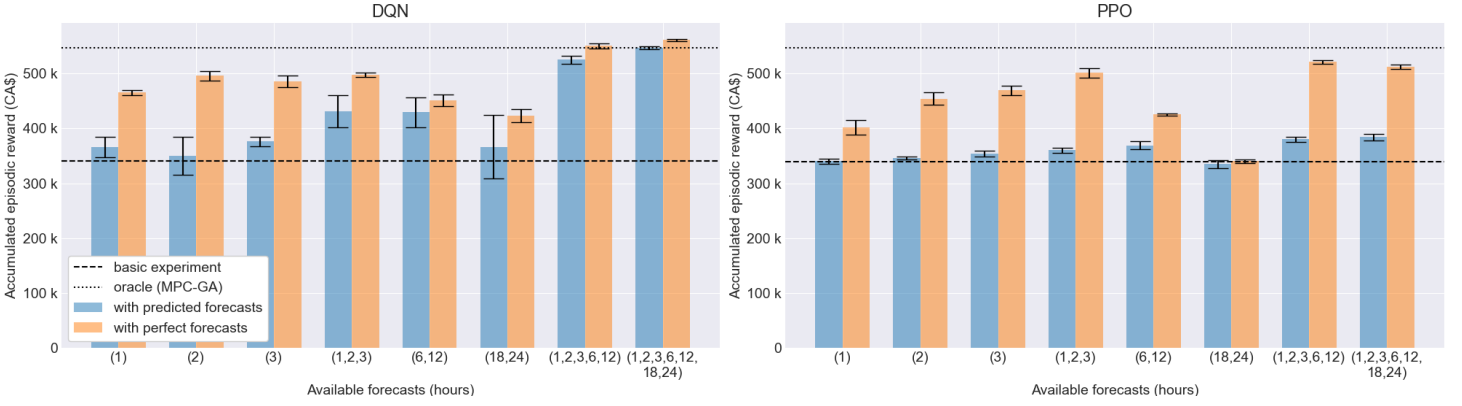
We further noticed that simple architectures, with small and few layers, performed better than more complex structures. The best performing CNNs were all characterized by small kernel sizes (1-3) and strides (1-2), bringing them closer to regular, fully connected layers. Recurrent cells, the attention mechanism, and more complex convolutions all did not improve or even worsened results. Our interpretation of this is that the spiky and irregular price data caused higher-capacity models to learn patterns that were coincidental and non-predictive. The non-stationarity of the data, especially when comparing training, validation, and test sets, further aggravated the task, slightly favoring smaller models with less tendency to overfit.

Experiments with smoothened input data did not improve performance for any of the smoothing coefficients and horizons tested. Regarding input variables, model performance improved when adding the electricity demand of Alberta and hour of the day. The hour of the day was thereby encoded using a sine and a cosine wave to better convey the cyclic nature of the data [41].

### 5.2 Energy Arbitrage with Price Forecasts

We begin the analysis of the results of DRL on EA by looking at the basic experiments (no forecasts) and those with perfect forecasts (ground truth is provided). In the basic experiments (see dashed lines in Fig. 4), DQN and PPO score nearly identical and accumulate around CA\$340,000 in rewards throughout the episode. For the experiments with forecasts, we group the horizons into three ranges: short-term (1,2,3h), middle term (6,12h)

<sup>1</sup>[https://github.com/masa2203/battery\\_arbitrage\\_with\\_drl](https://github.com/masa2203/battery_arbitrage_with_drl)



**FIGURE 4: DQN (left) and PPO (right) performance with different forecasting horizons for perfect (ground truth) and predicted forecasts after 50 episodes of training. All results were averaged over five independent runs. The error bars show  $\pm$  one standard deviation.**

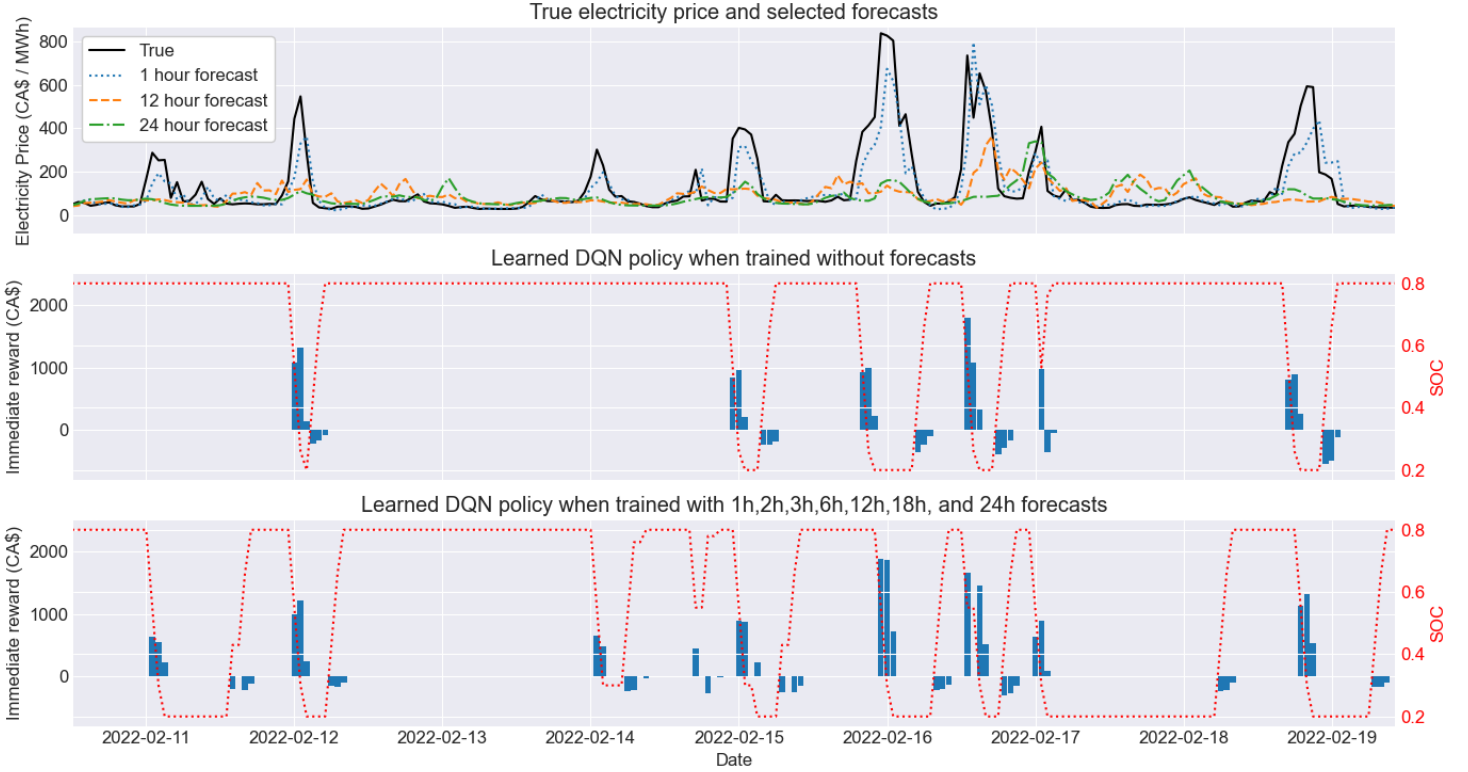
and long-term (18,24h). When provided with perfect forecasts (orange bars in Fig. 4), both DQN and PPO profit the most from short-term forecasts. We therefore subdivided the short-term forecast into its components and found that DQN profited the most from perfect 2h forecasts, and PPO from perfect 3h forecasts. While still enabling performance gains, the 1h forecast was the least useful short-term forecast. This finding is interesting as some of the related work solely relied on forecasting the next hour, possibly missing out further improvements. Our observation is underlined by the experiments combining short and middle-term forecasts and those with all forecasts. DQN and PPO further improved, with PPO scoring best with 1h, 2h, 3h, 6h, and 12h perfect forecasts and DQN with all seven horizons. In these two experiments, DQN outperformed even the oracle (MPC-GA). Comparing DQN and PPO on the experiments with perfect forecasts in Fig. 4, DQN achieved higher rewards on all combinations except for the short-term range. For both models, the stronger performances with 2h, 3h, and combined forecasts indicate that the agent requires time to react to changing prices. This is barely possible if only the next hour price is forecasted.

As a next step, we analyze the results of PPO and DQN with predicted forecasts (blue bars in Fig. 4). Naturally, the high prediction errors caused these experiments to score worse compared to perfect forecasts. For PPO, only moderate improvements compared to the basic experiment were achieved. Short-term and middle-term forecasts increased rewards by 6% and 9%, respectively. Long-term forecasts slightly decreased rewards. The best experiment, a combination of all seven forecasts, yielded an improvement of 14%. DQN overall responded better to predicted forecasts but showed increased instability across independent runs for single forecasts and ranges. However, the combination of multiple forecast ranges resulted in high and stable rewards. These experiments also visibly narrowed the gap between predicted and perfect forecasts. With access to all seven forecasters, DQN reached an accumulated reward of CA\$547,000 or 60% more than without forecasters (basic experiment). This represents a remarkable increase when considering the high forecasting errors.

To interpret these results, we compare the learned policies of basic DQN and DQN with access to all predictions. Figure 5 shows a nine-day sample from both policies below true

and predicted electricity prices for selected horizons during the same time interval. A striking difference between both policies is the increased activity for the agent with access to forecasts. Throughout the entire one-year episode, 2056 charging or discharging activities were recorded for the agent with access to forecasts, compared to only 1179 for the agent without forecasts. PPO charged or discharged the battery 1175 times without forecasts and only 1491 times when provided with forecasts. This reveals that the performance difference between the two DRL models is due to the greater increase in battery cycles for DQN. The increase in activity does shorten battery life, but it is factored in the reward function by increased degradation cost (see Eq. 4). Furthermore, after a discharge, the agent with forecasts tended to wait longer before recharging and thereby purchased electricity at lower prices. This behavior is well visible around February 19th in Fig. 5. Similarly, the agent with forecasts exploited high electricity prices during price spikes better and discharged the battery mostly at the peak. The agent without forecasts discharged the battery a few hours earlier when prices were increasing but still lower, for example on February 16th. For the entire episode, access to forecasts increased sales by CA\$350,000 whereas purchases increased by only CA\$105,000.

Only the combination of forecasters for multiple horizons significantly improved results. We assume that this is due to a “majority vote” mechanism, in which the presence of multiple forecasts helps the agent to recognize the right tendency. With fewer forecasts available, due to high forecasting errors, the “majority vote” becomes less accurate leading to suboptimal policies. Figure 6 shows the training progress of DQN and PPO in the basic experiments and with all forecasts. All RL experiments outperformed the CEM. While due to a different exploration behavior PPO is more sample efficient than DQN, it did not improve much with access to forecasts. On the other side, DQN benefited from forecasts not only regarding rewards but also in terms of stability. A possible explanation for the poor performance of PPO is that one of its key strengths, the fine-grained control of continuous action spaces, is not required for price-based arbitrage but makes learning a policy more difficult. The simpler DQN, assigning values to state-action pairs, might be better suited to recognize long-term dependencies by utilizing predictions. As charging the



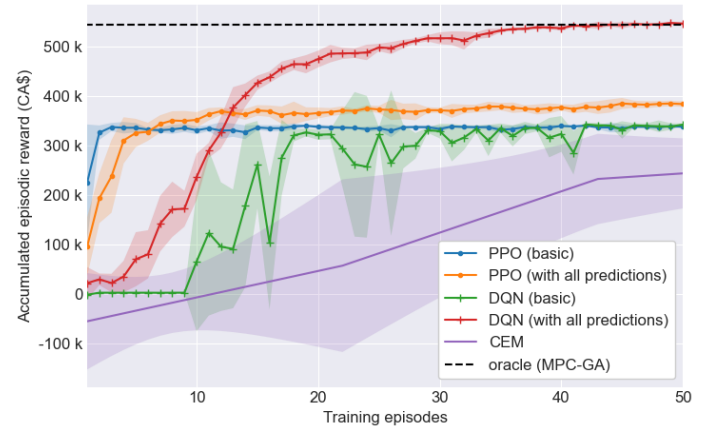
**FIGURE 5: A nine day sample of (Top) true electricity prices and predictions for selected horizons with the best deep learning models found, (Middle) the battery SOC and rewards for a trained DQN agent without access to forecasts, (Bottom) for a trained DQN agent with access to all seven predictions.**

battery is linked with negative rewards, and the corresponding profits can occur many time-steps later, this characteristic might explain the success of DQN.

Finally, it is important to note that the oracle does not provide an optimal solution and that its performance highly depends on the allocated computational resources. Conducting repeated optimizations over receding horizons, the quality of the solution depends on parameters such as the population size and number of iterations of the genetic algorithm. Table 2 compares the compute times and rewards of the models from Fig. 6 on the same system (64-bit Windows 11 Pro, Intel Core i9-12900K CPU, 64GB RAM, and NVIDIA GeForce RTX 3080 GPU). The MPC-GA algorithm in this case study required almost 12 hours of runtime, while the best performing DQN with price predictions completed training after 2 hours on the same system. Table 2 further shows that the performance gains from adding predictions come with higher compute cost due to larger state spaces and repeated querying of the pre-trained predictors during DRL training.

## 6. CONCLUSIONS

This study showed that time-series forecasting can substantially improve the performance of RL on price-based arbitrage with batteries. Despite high forecasting errors, rewards increased by 60% in our case study when DQN was supplied with predictions on future electricity prices. The key takeaway from our experiments is that these performance gains are only possible if multiple forecasts for different horizons are combined. Fu-



**FIGURE 6: Training progress of tested models with and without forecasts. All results were averaged over five independent runs. The shaded areas show  $\pm$  one standard deviation.**

ture work will have to investigate if our findings generalize to other locations and battery dispatch tasks with more uncertain variables. Besides, updating the weights of the forecasters with unseen data during agent-environment interaction could further improve results in the future.



**TABLE 2: Average compute times and rewards of the models shown in Fig. 6**

Model	Compute time (s)	Acc. episodic reward (CA\$)
PPO (basic)	603	339k
PPO (with predictions)	4,620	384k
DQN (basic)	1,529	341k
DQN (with predictions)	7,341	547k
CEM	2,490	241k
Oracle (MPC-GA)	42,210	546k

## ACKNOWLEDGMENTS

This work was supported by the McGill Engineering Doctoral Award (MEDA).

## REFERENCES

- [1] Cao, Jun, Harrold, Dan, Fan, Zhong, Morstyn, Thomas, Healey, David and Li, Kang. “Deep Reinforcement Learning-Based Energy Storage Arbitrage With Accurate Lithium-Ion Battery Degradation Model.” *IEEE Transactions on Smart Grid* Vol. 11 No. 5 (2020): pp. 4513–4521. DOI [10.1109/TSG.2020.2986333](https://doi.org/10.1109/TSG.2020.2986333).
- [2] Wang, Hao and Zhang, Baosen. “Energy Storage Arbitrage in Real-Time Markets via Reinforcement Learning.” *2018 IEEE Power & Energy Society General Meeting (PESGM)*: pp. 1–5. 2018. DOI [10.1109/PESGM.2018.8586321](https://doi.org/10.1109/PESGM.2018.8586321).
- [3] Rintamäki, Tuomas, Siddiqui, Afzal S. and Salo, Ahti. “Does renewable energy generation decrease the volatility of electricity prices? An analysis of Denmark and Germany.” *Energy Economics* Vol. 62 (2017): pp. 270–282. DOI [10.1016/j.eneco.2016.12.019](https://doi.org/10.1016/j.eneco.2016.12.019).
- [4] Ciarreta, Aitor, Pizarro-Irizar, Cristina and Zarraga, Ainhoa. “Renewable energy regulation and structural breaks: An empirical analysis of Spanish electricity price volatility.” *Energy Economics* Vol. 88 (2020): p. 104749. DOI [10.1016/j.eneco.2020.104749](https://doi.org/10.1016/j.eneco.2020.104749).
- [5] Biggins, F.A.V., Homan, S., Ejeh, J.O. and Brown, S. “To trade or not to trade: Simultaneously optimising battery storage for arbitrage and ancillary services.” *Journal of Energy Storage* Vol. 50 (2022): p. 104234. DOI [10.1016/j.est.2022.104234](https://doi.org/10.1016/j.est.2022.104234).
- [6] Metz, Dennis and Saraiva, João Tomé. “Use of battery storage systems for price arbitrage operations in the 15- and 60-min German intraday markets.” *Electric Power Systems Research* Vol. 160 (2018): pp. 27–36. DOI [10.1016/j.epsr.2018.01.020](https://doi.org/10.1016/j.epsr.2018.01.020).
- [7] Zhang, Zidong, Zhang, Dongxia and Qiu, Robert C. “Deep reinforcement learning for power system applications: An overview.” *CSEE Journal of Power and Energy Systems* Vol. 6 No. 1 (2020): pp. 213–225. DOI [10.17775/CSEE-JPES.2019.00920](https://doi.org/10.17775/CSEE-JPES.2019.00920).
- [8] Meng, Fanyi, Bai, Yang and Jin, Jingliang. “An advanced real-time dispatching strategy for a distributed energy system based on the reinforcement learning algorithm.” *Renewable Energy* Vol. 178 (2021): pp. 13–24. DOI [10.1016/j.renene.2021.06.032](https://doi.org/10.1016/j.renene.2021.06.032).
- [9] Lujano-Rojas, Juan M., Yusta, José M., Domínguez-Navarro, José A., Osório, Gerardo J., Shafie-khah, Miadreza, Wang, Fei and Catalão, João P. S. “Combining Genetic and Gravitational Search Algorithms for the Optimal Management of Battery Energy Storage Systems in Real-Time Pricing Markets.” *2020 IEEE Industry Applications Society Annual Meeting*: pp. 1–7. 2020. DOI [10.1109/IAS44978.2020.9334858](https://doi.org/10.1109/IAS44978.2020.9334858).
- [10] Feng, Lu, Zhang, Xinjing, Li, Chengyuan, Li, Xiaoyu, Li, Bin, Ding, Jie, Zhang, Chao, Qiu, Han, Xu, Yujie and Chen, Haisheng. “Optimization analysis of energy storage application based on electricity price arbitrage and ancillary services.” *Journal of Energy Storage* Vol. 55 (2022): p. 105508. DOI [10.1016/j.est.2022.105508](https://doi.org/10.1016/j.est.2022.105508).
- [11] Zhang, Bin, Hu, Weihao, Cao, Di, Huang, Qi, Chen, Zhe and Blaabjerg, Frede. “Deep reinforcement learning–based approach for optimizing energy conversion in integrated electrical and heating system with renewable energy.” *Energy Conversion and Management* Vol. 202 (2019): p. 112199. DOI [10.1016/j.enconman.2019.112199](https://doi.org/10.1016/j.enconman.2019.112199).
- [12] Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A, Veness, Joel, Bellemare, Marc G, Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K, Ostrovski, Georg et al. “Human-level control through deep reinforcement learning.” *nature* Vol. 518 No. 7540 (2015): pp. 529–533. DOI [10.1038/nature14236](https://doi.org/10.1038/nature14236).
- [13] Schulman, John, Wolski, Filip, Dhariwal, Prafulla, Radford, Alec and Klimov, Oleg. “Proximal policy optimization algorithms.” *arXiv preprint arXiv:1707.06347* (2017).
- [14] Bissing, Daniel, Klein, Michael T., Chinnathambi, Radhakrishnan Angamuthu, Selvaraj, Daisy Flora and Ranganathan, Prakash. “A Hybrid Regression Model for Day-Ahead Energy Price Forecasting.” *IEEE Access* Vol. 7 (2019): pp. 36833–36842. DOI [10.1109/ACCESS.2019.2904432](https://doi.org/10.1109/ACCESS.2019.2904432).
- [15] Castañeda-Leyva, Netzahualcōyotl, Hernández-Ramos, Hugo, Pérez-Hernández, Leonel Ramón and Rodríguez-Narciso, Silvia. “Prediction of electricity prices for non-regulated markets based on a power transformed mean reverting process.” *Applied Stochastic Models in Business and Industry* Vol. 38 No. 4 (2022): pp. 677–694. DOI [10.1002/asmb.2681](https://doi.org/10.1002/asmb.2681).
- [16] He, Hui, Hu, Ran, Zhang, Yaning, Zhang, Ying and Jiao, Runhai. “A Power Forecasting Approach for PV Plant based on Irradiance Index and LSTM.” *2018 37th Chinese Control Conference (CCC)*: pp. 9404–9409. 2018. DOI [10.23919/ChiCC.2018.8483960](https://doi.org/10.23919/ChiCC.2018.8483960).
- [17] Dehghani, Adnan, Moazam, Hamza Mohammad Zakir Hiyat, Mortazavizadeh, Fatemehsadat, Ranjbar, Vahid, Mirzaei, Majid, Mortezaei, Saber, Ng, Jing Lin and Dehghani, Amin. “Comparative evaluation of LSTM, CNN, and ConvLSTM for hourly short-term streamflow forecasting using deep learning approaches.” *Ecological Informatics* Vol. 75 (2023): p. 102119. DOI [10.1016/j.ecoinf.2023.102119](https://doi.org/10.1016/j.ecoinf.2023.102119).
- [18] Farsi, Behnam, Amayri, Manar, Bouguila, Nizar and Eicker, Ursula. “On Short-Term Load Forecasting Using Machine

- Learning Techniques and a Novel Parallel Deep LSTM-CNN Approach.” *IEEE Access* Vol. 9 (2021): pp. 31191–31212. DOI [10.1109/ACCESS.2021.3060290](https://doi.org/10.1109/ACCESS.2021.3060290).
- [19] Peng, Lu, Liu, Shan, Liu, Rui and Wang, Lin. “Effective long short-term memory with differential evolution algorithm for electricity price prediction.” *Energy* Vol. 162 (2018): pp. 1301–1314. DOI [10.1016/j.energy.2018.05.052](https://doi.org/10.1016/j.energy.2018.05.052).
- [20] Meng, Anbo, Wang, Peng, Zhai, Guangsong, Zeng, Cong, Chen, Shun, Yang, Xiaoyi and Yin, Hao. “Electricity price forecasting with high penetration of renewable energy using attention-based LSTM network trained by crisscross optimization.” *Energy* Vol. 254 (2022): p. 124212. DOI [10.1016/j.energy.2022.124212](https://doi.org/10.1016/j.energy.2022.124212).
- [21] Qu, Jiaqi, Qian, Zheng and Pei, Yan. “Day-ahead hourly photovoltaic power forecasting using attention-based CNN-LSTM neural network embedded with multiple relevant and target variables prediction pattern.” *Energy* Vol. 232 (2021): p. 120996. DOI [10.1016/j.energy.2021.120996](https://doi.org/10.1016/j.energy.2021.120996).
- [22] Livieris, Ioannis E., Stavroyiannis, Stavros, Iliadis, Lazaros and Pintelas, Panagiotis. “Smoothing and stationarity enforcement framework for deep learning time-series forecasting.” *Neural Computing and Applications* Vol. 33 No. 20 (2021): pp. 14021–14035. DOI [10.1007/s00521-021-06043-1](https://doi.org/10.1007/s00521-021-06043-1).
- [23] Han, Gwangwoo, Lee, Sanghun, Lee, Jaemyung, Lee, Kangyong and Bae, Joongmyeon. “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 (2021): p. 102868. DOI [10.1016/j.est.2021.102868](https://doi.org/10.1016/j.est.2021.102868).
- [24] Huang, Bin and Wang, Jianhui. “Deep-Reinforcement-Learning-Based Capacity Scheduling for PV-Battery Storage System.” *IEEE Transactions on Smart Grid* Vol. 12 No. 3 (2021): pp. 2272–2283. DOI [10.1109/TSG.2020.3047890](https://doi.org/10.1109/TSG.2020.3047890).
- [25] Harrold, Daniel J.B., Cao, Jun and Fan, Zhong. “Data-driven battery operation for energy arbitrage using rainbow deep reinforcement learning.” *Energy* Vol. 238 (2022): p. 121958. DOI [10.1016/j.energy.2021.121958](https://doi.org/10.1016/j.energy.2021.121958).
- [26] da Silva André, Joel, Stai, Eleni, Stanojev, Ognjen and Hug, Gabriela. “Battery control with lookahead constraints in distribution grids using reinforcement learning.” *Electric Power Systems Research* Vol. 211 (2022): p. 108551. DOI [10.1016/j.epsr.2022.108551](https://doi.org/10.1016/j.epsr.2022.108551).
- [27] Dong, Yi, Dong, Zhen, Zhao, Tianqiao and Ding, Zhengtao. “A Strategic Day-ahead bidding strategy and operation for battery energy storage system by reinforcement learning.” *Electric Power Systems Research* Vol. 196 (2021): p. 107229. DOI [10.1016/j.epsr.2021.107229](https://doi.org/10.1016/j.epsr.2021.107229).
- [28] Sage, Manuel and Zhao, Yaoyao Fiona. “Deep Reinforcement Learning for Economic Battery Dispatch: A Comprehensive Comparison of Algorithms and Experiment Design Choices.” (2024). Manuscript under review.
- [29] Sutton, Richard S and Barto, Andrew G. *Reinforcement learning: An introduction*. MIT press (2018).
- [30] Sage, Manuel, Staniszewski, Martin and Zhao, Yaoyao Fiona. “Economic Battery Storage Dispatch with Deep Reinforcement Learning from Rule-Based Demonstrations.” *2023 International Conference on Control, Automation and Diagnosis (ICCAD)*: pp. 1–6. 2023. DOI [10.1109/ICCAD57653.2023.10152299](https://doi.org/10.1109/ICCAD57653.2023.10152299).
- [31] Subramanya, Rakshith, Sierla, Seppo A. and Vyatkin, Valeriy. “Exploiting Battery Storages With Reinforcement Learning: A Review for Energy Professionals.” *IEEE Access* Vol. 10 (2022): pp. 54484–54506. DOI [10.1109/ACCESS.2022.3176446](https://doi.org/10.1109/ACCESS.2022.3176446).
- [32] Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P. “Gradient-based learning applied to document recognition.” *Proceedings of the IEEE* Vol. 86 No. 11 (1998): pp. 2278–2324. DOI [10.1109/5.726791](https://doi.org/10.1109/5.726791).
- [33] Hochreiter, Sepp and Schmidhuber, Jürgen. “Long Short-term Memory.” *Neural computation* Vol. 9 (1997): pp. 1735–80. DOI [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- [34] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz and Polosukhin, Illia. “Attention is all you need.” *Advances in neural information processing systems* Vol. 30 (2017).
- [35] Tran, Duong and Khambadkone, Ashwin M. “Energy Management for Lifetime Extension of Energy Storage System in Micro-Grid Applications.” *IEEE Transactions on Smart Grid* Vol. 4 No. 3 (2013): pp. 1289–1296. DOI [10.1109/TSG.2013.2272835](https://doi.org/10.1109/TSG.2013.2272835).
- [36] Cheng, Yu-Shan, Liu, Yi-Hua, Hesse, Holger C., Naumann, Maik, Truong, Cong Nam and Jossen, Andreas. “A PSO-Optimized Fuzzy Logic Control-Based Charging Method for Individual Household Battery Storage Systems within a Community.” *Energies* Vol. 11 No. 2 (2018). DOI [10.3390/en11020469](https://doi.org/10.3390/en11020469).
- [37] Cole, Wesley and Karmakar, Akash. “Cost Projections for Utility-Scale Battery Storage: 2023 Update.” *National Renewable Energy Lab (NREL), Golden, CO (United States)* (2023) DOI [10.2172/1984976](https://doi.org/10.2172/1984976).
- [38] Alberta Electric System Operator. “Market and system reporting.” (2023). <https://www.aeso.ca/market/market-and-system-reporting/>, visisted on 2023-05-27.
- [39] Hersbach, Hans, Bell, Bill, Berrisford, Paul, Hirahara, Shoji, Horányi, András, Muñoz-Sabater, Joaquín, Nicolas, Julien, Peubey, Carole, Radu, Raluca, Schepers, Dinand et al. “The ERA5 global reanalysis.” *Quarterly Journal of the Royal Meteorological Society* Vol. 146 No. 730 (2020): pp. 1999–2049. DOI [10.1002/qj.3803](https://doi.org/10.1002/qj.3803).
- [40] Szita, István and Lörincz, András. “Learning Tetris Using the Noisy Cross-Entropy Method.” *Neural Computation* Vol. 18 No. 12 (2006): pp. 2936–2941. DOI [10.1162/neco.2006.18.12.2936](https://doi.org/10.1162/neco.2006.18.12.2936).
- [41] Huang, Junhui, Algahtani, Mohammed and Kaewunruen, Sakdirat. “Energy Forecasting in a Public Building: A Benchmarking Analysis on Long Short-Term Memory (LSTM), Support Vector Regression (SVR), and Extreme Gradient Boosting (XGBoost) Networks.” *Applied Sciences* Vol. 12 No. 19 (2022). DOI [10.3390/app12199788](https://doi.org/10.3390/app12199788).