# Battery Scheduling Algorithm for Energy Arbitrage via Reinforcement Learning

Alban Puech, Guruprerana Shabadi

***Abstract*—As the share of renewable energy sources increases in the electricity market, new solutions are needed to build a flexible and reliable grid that helps at aligning consumption with production. Energy arbitrage with battery storage systems supports renewable energy integration into the grid by shifting demand and increasing the overall utilization of power production systems. In this project, we propose an optimized scheduling algorithm for energy arbitrage via Reinforcement Learning. The algorithm optimally schedules the charge and discharge operations associated with the most profitable trading strategy. We compare our algorithm to a baseline based on a linear programming formulation and report similar performances. We finally illustrate the pitfalls of relying on a reinforcement learning approach.**

## I. Introduction

In IEA's pathway to net-zero CO2 emissions by 2050 [1], electricity accounts for 50 percent of the energy use in 2050, and 70 percent of it is produced by a mix of wind and solar PV. This increased dependence on variable energy sources (VES) calls for important investments into grid-scale storage solutions to balance their variable and non-dispatchable aspects. Those solutions are expected to help in increasing the grid flexibility, its capacity to handle sudden and large changes in the power supply or demand, and its reliability [2]. While Pumped-storage hydroelectricity (PHS) currently represents 90 % of the world's storage [3], battery system storage (BSS) are expected to see the largest market growth and can play a similar role [4]. BSS can have many different use cases, from replacing backup emergency power units in hospitals to being used as frequency response systems. [5] lists some of the applications of BSS. Energy arbitrage is one of the many ways that BSS that can help in accommodating the change in demand and production. BSSs are charged when the price of electricity is low and discharged when the price is high, generally during consumption peaks. Energy arbitrage effectively assists the integration of renewable energy to the grid and allows higher utilization of the VES by shifting the demand to better align it to the important production periods. The profitability of energy arbitrage with batteries depends on factors such as battery cost, the price difference between low-demand and high-demand periods, and current battery technologies are often shown to be too expensive for energy arbitrage to be profitable on its own in some electricity markets [6]. However, intraday price variability represents an opportunity for asset owners to stack additional revenue streams obtained from arbitrage to their existing ones. It provides additional incentives for operating and investing in grid-scale BSS. In addition to increasing the return-over-investments of BSS, energy arbitrage has numerous positive societal impacts. [7] shows that it increases the return to renewable production and reduces CO2 emissions. Energy arbitrage requires the generation of a charging schedule. Charging schedules are necessary to place the selling and buying orders and to plan the operations on the battery.

Electricity prices show large intraday variations, mainly driven by consumption and production. Those prices are not known in advance, and the main challenge in energy arbitrage is to deal with this uncertainty. Existing approaches in the literature using linear programming (LP) [8], [9], [10], [11], [12] assume perfect forecast of the prices, and do not deal with the price variation uncertainty. This motivates us to train a reinforcement learning (RL) agent that is able to predict optimal charging schedule for a battery in order to maximize profit through energy arbitrage without knowledge of the future hourly prices. The main challenge with this approach is to construct the ideal set of input for the algorithm to extract relevant information and patterns from the prices to make profitable decisions.

In this project, we propose a Partially Observable Markov Decision Process model and a simulation environment of the battery and the electricity market that uses real-life electricity prices. We further develop a reinforcement-learning trained agent that optimally schedules operations on the battery. We compare our approach to two baseline strategies based on linear programming. We find that the state-of-the-art reinforcement learning algorithm trained with our environment is able to achieve 75 percent of the maximal obtainable profit. However, we observe that the simple linear programming strategy with forecasted input prices attains comparable performance. We believe that this approach performs as well as the RL agent due to the high seasonality component of the hourly prices. However, in the context of increased complexity in the operational and availability requirements of real-world applications, the RL approach looks more promising.

## II. Background

### A. Decision process

A Markov Decision Process (MDP) is a model for sequential decision-making in a fully observable stochastic environment which satisfies the Markov property at each time step. Formally, we can define an MPC as a four-tuple $(S, A, P, R)$ where $S$ is the state space, $A$ is the set of actions $P$ represents the transition probabilities, and $R$ is the reward function. At each time step $t$, an agent acting on an MDP environment is given a state $S_t \in S$ and chooses an action $A_t \in A$. The

environment then transitions to the next state $S_{t+1} \in S$ with probability given by $P(S_{t+1} \mid S_t, A_t)$ and gives the reward $R(A_t, S_t, S_{t+1})$ to the agent.

A Partially Observable Markov Decision Process (POMDP) is a generalization of the conventional MDP model where the state is not fully-observable. It is defined by a six-tuple $(S, A, P, R, O, \Omega)$ where the four first components are the same as in an MDP model, with an additional observation space $O$, and an observation model $\Omega$. While the dynamics of the underlying environment remains the same as in an MDP, the agent acting on a POMDP in a given state $S_t \in S$ only receives an observation $O_t \in O$ with the probability $\Omega(O_t \mid S_t)$ which partially describes the state $S_t$.

The goal of an agent acting on an MDP or a POMDP is to build an optimal policy $\pi$ that chooses actions at each time step and maximizes the expected cumulative reward:

$$\text{Maximize: } \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$$

Where $r_t$ is the reward received at time $t$ and $\gamma \in [0, 1]$ is the discount factor describing the preference of the agent for immediate rewards rather than future rewards.

### B. Reinforcement learning

Reinforcement learning is a paradigm for learning optimal policies to control an agent in an environment described by an MDP or POMDP. The agent learns by trial and errors via repeated guided experiments without knowledge of the underlying transition probabilities and reward distribution. RL algorithms allow learning a policy $\pi$ which is used to choose an action $a_t = \text{argmax}_a \pi(a \mid S_t)$ in a state $S_t$.

### C. Policy gradient and actor-critic methods

Policy gradient methods directly update the policy parameters using gradient-based optimization, contrary to traditional deep reinforcement learning methods like deep Q-learning, which estimate $Q(S, A)$ values. Actor-critic methods are a type of policy-gradient methods that use the value function approximated by a neural network as a baseline estimate in the computation of the advantage function $\widehat{A}$, which is then used in the computation of the gradient. Proximal Policy Optimization [13] is a state-of-the-art policy-gradient based method which we use to train our RL agent.

## III. Approach

### A. Environment

Our environment consists of a battery for which we would like to build an agent that controls its charging (buying electricity) and discharging (selling electricity) in order to maximize the profits which depend on the variations in the price of electricity. These electricity prices also form a component of our environment. However, before constructing a model for our environment, we make the following reasonable assumptions:

1) Constant charge and discharge rates.
2) No self-discharge: the battery does not lose charge when idle.
3) No capacity fading and no charge efficiency decrease during the lifetime of the battery (we do not lose any energy while charging).
4) The electricity prices change every hour which means that each time step within our environment represents an interval of one hour.
5) The battery has a capacity of 100 kWh.

We then formulate the environment composed of the battery and the electricity prices as a POMDP $(S, A, P, R, O, \Omega)$ whose components are described in the following sections.

*1) State:* The price of electricity in the market is determined by complex processes and is influenced by several variables representing production and demand. As a result, we are unable to gather data of all these variables which influence the evolution of the prices. This implies that at any given time, we are unable to fully observe the state and motivates the use of a POMDP to model our environment.

Although it is not possible to observe all the variables of the state space that (along with the action) determine the next state, there are three variables of the state that can be determined. At every time step $t$ (hour of the day), the *partial* state of our environment is defined by $(SOC_t, C_t, p_t)$, where $C_t$ denotes the cumulative profit until time $t$, $SOC_t$ the state of charge of the battery at time $t$ as a ratio between 0 and 1, and $p_t$ the electricity price at time $t$.

*2) Action:* [14] shows that the optimal actions assuming a constant maximum energy increment and that the capacity is a multiple of the maximum energy increment are full rate charge, full rate discharge and hold. At each time step $t$, the action is chosen from the following set:

$$\mathcal{A} = \{D, H, C\}$$

where $D$ stands for discharge at maximum rate for the entire hour, $H$ represents *hold* which means that we do not charge or discharge for the next hour, and lastly $C$ represents charging at maximum rate.

*3) Transitions:* Since we cannot observe all the variables of the state, we cannot describe the transition probabilities that determine the next price $p_{t+1}$. However, the transitions of the $SOC$ and $C$ variables are deterministic. When an action $A_t$ is taken based on the partial state $S_t = (SOC_t, C_t, p_t)$, the next values of these variables at time $t + 1$ are computed as:

$$SOC_{t+1} = \begin{cases} \max(SOC_t - 0.5, 0) & A_t = D \\ SOC_t & A_t = H \\ \min(SOC_t + 0.5, 1) & A_t = C \end{cases}$$

$$C_{t+1} = C_t - p_t \times (SOC_{t+1} - SOC_t)$$

*4) Reward:* Our reward function is inspired from [15]. In order to define the reward, we first define at each step $t$, the valuation $V_t$ given by

$$V_t = C_t + p_t \times SOC_t$$

Intuitively, the valuation represents the current value of the energy stored in the battery, along with the cumulative profits

made so far. Then, the reward $R_{t+1}$ is given by the change in valuation as

$$R_{t+1} = V_{t+1} - V_t$$

Observe that the sum of rewards directly corresponds to the cumulative profit made so far. Moreover, compared to using the positive and negative cash flow at each step as reward as proposed by [16], this reward function allows to continuously assign rewards that effectively represent the incremental profit of each action (including hold).

*5) Observation State:* Due to the unobservability of all the state variables which determine the future prices, we rely on additional variables which we add to the observation state of the POMDP in order to aid the agent in taking the appropriate actions for maximizing arbitrage, i.e., the agent needs to know the trend in the price variations to take well-informed decisions.

Intuitively, at every time step $t$, a profitable action consists in buying electricity when $p_t$ reaches a local minimum, and selling it when $p_t$ reaches a local maximum. It is thus insufficient to rely only on the current price to determine the best action to take. For this reason, we include $p_{t-1}$, $p_{t-2}$, $\ldots$, $p_{t-k}$ within the observation state $O_t$.

Similarly, knowing the prices for the next hours, namely, $p_{t+1}, p_{t+2}, \ldots$ is of crucial importance to, for example, assess the profitability of a buying action. As we do not know them, we rely on estimates $h_{t+1}, h_{t+2}, \ldots, h_{t+23}$, computed as the prices at each hour of the day averaged over the last $l$ days.

Note that including the full sequence of 24 hourly estimates allows the model to compare the prices observed during the current day with the historical estimates. According to how far the observed prices are from the historical estimates, we hope the algorithm to rely more or less on those estimated future price for the decision-making.

### B. Baseline model

Along with our reinforcement learning approach, we also build two baselines that formulate the problem as a linear optimization one.

The first one outputs the optimal charge schedule using the true future prices. We refer to this model as LP-OPTIM.

$$\begin{aligned} &\text{Given } SOC_0, p_t \quad \forall t \in \{0, 1, .., 23\} \\ &\text{Maximize: } -\sum_{t=0}^{23} p_t \times (SOC_{t+1} - SOCt) \\ &\text{Subject to: } \mid SOC[t-1] - SOC[t] \mid \quad \leq 0.5 \quad \forall t \in \{1, .., 24\} \end{aligned}$$

Note that the profit obtained using this baseline corresponds to the maximum obtainable profits given the assumptions stated above.

The second baseline (LP-PRED) corresponds to the same linear optimization formulation solved using $h_1, h_2, \ldots, h_{23}$ as estimates of $p_1, p_2, \ldots, p_{23}$

### C. Performance metric of a model

Given a sequence of hourly prices $p_0, \ldots, p_T$, an agent $M$ outputs a charging schedule for the battery over the time horizon $[0, T]$, and a corresponding profit $P(M)$. Let $P(\text{LP-OPTIM})$ be the maximum possible profit achieved by the LP-OPTIM model. Then we define the performance of our model to be the ratio:

$$\Xi(M) = \frac{P(M)}{P(\text{LP-OPTIM})}$$

Our goal then becomes to maximize this performance metric.

We also use the following metrics to compare the model to our baselines:

$$\begin{aligned} \#\text{Cycles} &= \sum_{t=0}^{T-1} \frac{\mid SOC_{t+1} - SOC_t \mid}{2} \\ \#\text{Action switches} &= \sum_{t=1}^{T-1} \mathbb{1}_{(SOC_t - SOC_{t-1}) \neq (SOCt+1 - SOC_t)} \end{aligned}$$

We would like our agent to minimize the number of cycles and action switches to reduce strain on the battery.

## IV. Results and Discussion

### A. Data and Training

We use the data available from the European Hourly Day-Ahead Electricity Price dataset [17]. For training and evaluation, we use the pricing data from the years 2020-2022 in Germany. Data from 2021 is used for training, 2022 for testing and 2020 as the evaluation set. The data is available in the form of hourly prices per Megawatt-hour (MWh). Fig.1 displays the evaluation, train and test data used for the RL algorithm.

Our environment is implemented as an Open AI Gym environment. We use the implementation of Proximal Policy Optimization (PPO) [13] algorithm from StableBaselines3 [18]. During the training process, the model is evaluated on the evaluation dataset every 10,000 steps and the one achieving the highest cumulative reward is used. We stop the training when we observe that the performance on the evaluation data starts declining, as shown in Fig.2.

### B. Results

Fig.3 shows the daily profit distribution over the test set obtained using the LP-OPTIM, the LP-PRED and the RL strategies (top to bottom). The RL algorithm is able to replicate closely the profit distribution from the LP-PRED algorithm. There does not seem to be any particular situation or days when the algorithm is performing erratically. It is able to generalize to variations in prices and to price ranges that are not seen in the training set: This is surprising because, as seen in Fig.1, the prices seen in the training and the evaluation sets show different variations than in the test dataset. Stopping the training based on results achieved on a set showing more similar price variations to the test set would have probably given better results.

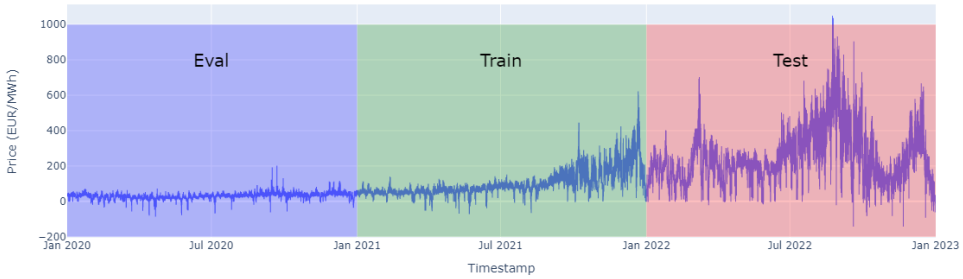


Fig. 1: Day-Ahead hourly electricity prices in the training (covering the year 2021), evaluation (2020), and test datasets (2022)


Fig. 2: Evolution of the cumulative reward per episode on the training and evaluation set during training. The peak on the evaluation set was reached after 50000 iterations.

Fig.4 shows the relative difference in profit between RL and LP-OPTIM. We see that from September to end of December 2022, the prices show large-scale variations across several weeks. In this period, the relative difference in profit between the RL algorithm and LP-OPTIM is higher compared to the rest of the year. This might be linked to the training set not showing such large-scale trends. There are some days when the RL algorithm gives higher profit than LP-OPTIM (Fig.4 shows negative relative difference). This is due to the fact that LP-OPTIM optimizes the schedule on a daily basis, ending the day with a fully discharged battery, while the RL algorithm does not have this restriction.

### *C. Analysis of the RL algorithm*

In Table I, we document the performance of the two baseline algorithms (LP-OPTIM and LP-PRED) along with the RL algorithm. The average daily profits obtained using LP-PRED and RL are similar (respectively 80 percent and 75 percent of the LP-OPTIM profits). On the other hand, the number of charging-discharging cycles performed following the schedule obtained with the RL-algorithm is significantly lower than with LP-PRED (16 percent lower) or LP-OPTIM (32 percent lower). The same observation can be made on the number of action switches metric. This indicates that the RL algorithm is more conservative in its approach to arbitrage and has a bias towards holding actions when the price variations are small. This is shown in Fig.5 where LP-OPTIM exploits every price spread, while the RL strategy only charges and discharges the battery on particularly large local extrema.

While the RL algorithm is able to generate reasonable profits, the model is in practice much harder to develop and deploy in real life settings: it requires training, fine-tuning, supervision, and maintenance while the baselines can be easily implemented without requiring large computational resources and data pipeline.

Furthermore, we do not have any guarantees on how the RL algorithm performs. It only recognizes patterns from the training in order to generate a sensible charging schedule that generates profit. On the other hand, LP-PRED performs optimally given the price predictions. This means that its performance only depends on the accuracy of the predictions. While the schedule generated by LP-PRED is the result of a linear optimization (optimal schedule for the given price predictions), the RL algorithm is unable to replicate such a process because it only defines its internal rules based on past experiences. It is difficult to expect the RL algorithm to output an optimal schedule for unseen price trends.

We were expecting the RL algorithm to better cope with the price uncertainty, given that it can adapt the control decisions based on some underlying latent variables deduced from the

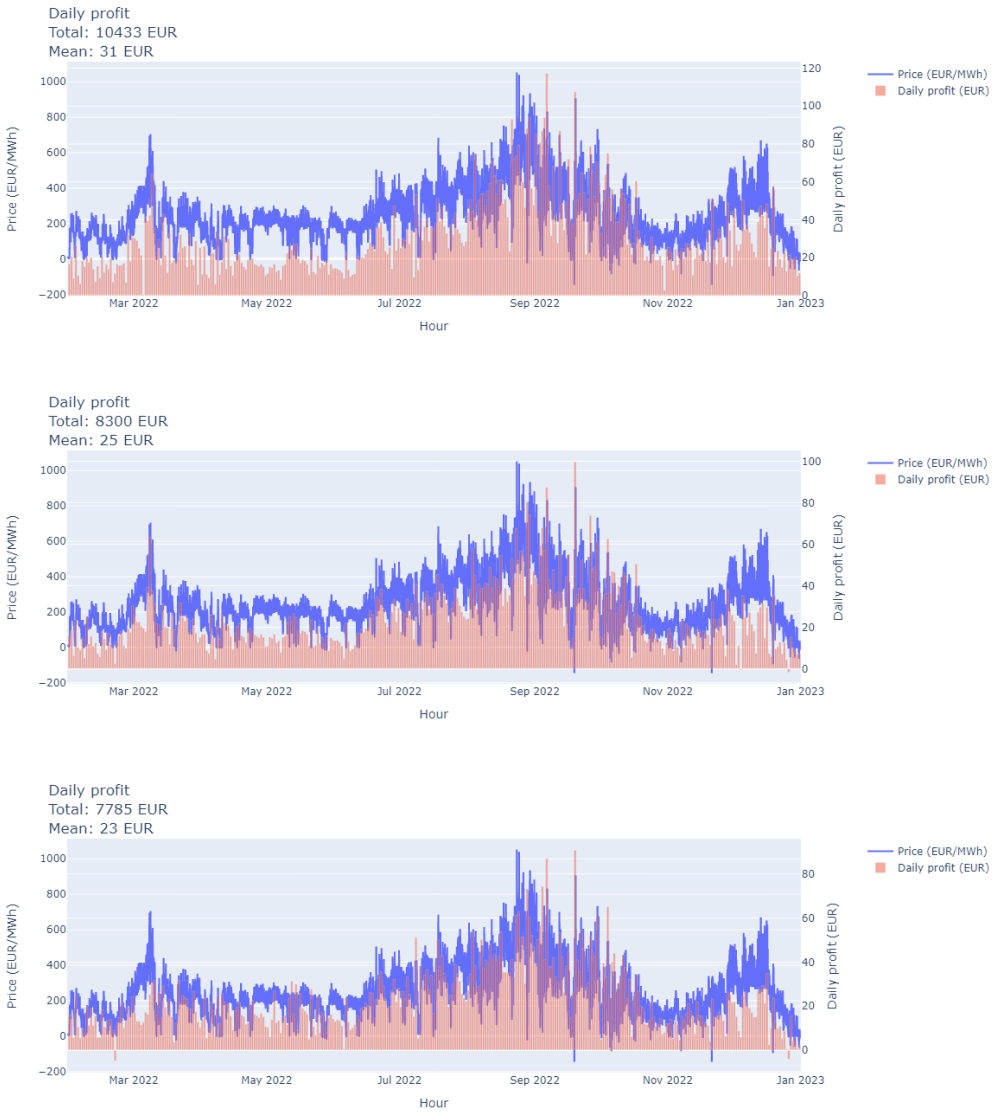


Fig. 3: Daily profit distribution following the schedule given by LP-OPTIM (top), LP-PRED (middle), and RL (bottom)

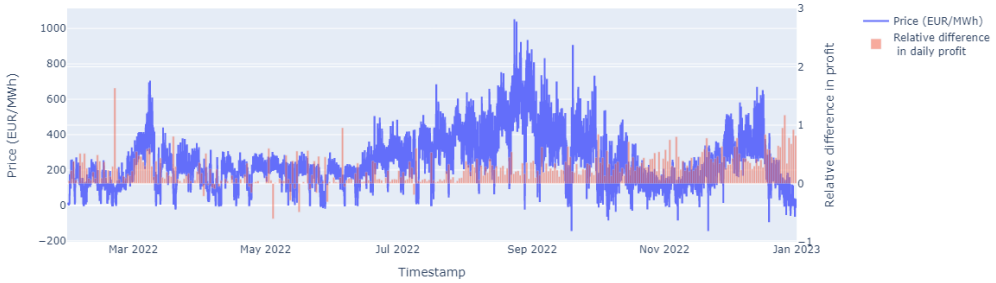


Fig. 4: Relative difference in daily profit between LP-OPTIM and RL

| Alg. | Avg. Daily Profit | Ξ | #Cycles | #Action switches | #Days neg. profits |
|---|---|---|---|---|---|
| **LP-OPTIM** | 31.05 | 1 | 957.25 | 4197 | 0 |
| **LP-PRED** | 24.70 | 0.80 | 779.5 | 3324 | 2 |
| **RL-PPO** | 23.17 | 0.75 | 654.5 | 2611 | 2 |

TABLE I: Evaluation of the different metrics of the baseline and the PPO algorithm on the test dataset

electricity prices. However, the strong performance of the LP-PRED baseline indicates that the price variations have a large seasonality component and do not show high stochasticity.

### *D. Transfer performance*

We try our RL agent trained on the 2021 German prices on the 2022 French electricity prices.

Fig.6 shows the daily profit distribution over the test set obtained using the LP-OPTIM, the LP-PRED and the RL strategies (top to bottom). Again, RL and LP-PRED show similar daily profit distribution. The cumulative profit obtained using RL is actually closer to the one obtained with LP-PRED than it was the case for Germany. The algorithm is able to generalize well to other price distributions from other countries.

Another interesting observation is that both RL and LP-PRED model are able to take advantage of the large price surge of April 4th. Recall that LP-PRED does not rely on the future prices, so that the decision to buy before the surge and to sell during the surge is not directly motivated by observation of this price peak. However, the price surge happened at an hour that showed large prices in the historical data used for the computation of the schedule, so that the algorithm was planning on selling the electricity at this time regardless of the existence of the peak. The RL algorithm does not have access to the current time. However, it was still able to track the price change, as shown in Fig.7. Indeed, the algorithm emptied the initially fully charged battery for the entire duration of the price peak.

This example highlights the robustness of the RL algorithm that is able to output relevant schedule in extreme price situations that have not been seen during training.

## V. Conclusions

We have implemented, using proximal policy optimization algorithm, a reinforcement learning agent that controls the charging and discharging schedule of a battery and is able to generate 75% of the maximum attainable profits (under full knowledge of the prices). We compared our RL agent to baseline models LP-PRED and LP-OPTIM which are based on linear programming. We also discuss the limitations of the RL agent which utility in a real-world environment is undermined by the exceptional performance of the simple LP-PRED algorithm. However, we claim that in the case of higher stochasticity in the electricity prices and complex real-world constraints, reinforcement learning can yield better control policies at a smaller computational cost.

## References


[1] Iea. Net zero by 2050 – analysis, May 2021.

[2] Claudia Pavarini. Battery storage is (almost) ready to play the flexibility game – analysis, Nov 2018.

[3] Mike McWilliams. 6.08 - pumped storage hydropower. In Trevor M. Letcher, editor, *Comprehensive Renewable Energy (Second Edition)*, pages 147–175. Elsevier, Oxford, second edition edition, 2022.

[4] Max Schoenfisch and Amrita Dasgupta. Grid-scale storage – analysis, Sep 2022.

[5] Vinayak Sharma, Andres Cortes, and Umit Cali. Use of forecasting in energy storage applications: A review. *IEEE Access*, 9:114690–114704, 2021.

[6] Xinjing Zhang, Chao (Chris) Qin, Eric Loth, Yujie Xu, Xuezhi Zhou, and Haisheng Chen. Arbitrage analysis for different energy storage technologies and strategies. *Energy Reports*, 7:8198–8206, 2021.

[7] Omer Karaduman. Economics of grid-scale energy storage. *Job market paper*, 2020.

[8] Dheepak Krishnamurthy, Canan Uckun, Zhi Zhou, Prakash R. Thimmapuram, and Audun Botterud. Energy storage arbitrage under day-ahead and real-time price uncertainty. *IEEE Transactions on Power Systems*, 33:84–93, 2018.

[9] Md Umar Hashmi, Arpan Mukhopadhyay, Ana Bušić, and Jocelyne Elias. Optimal control of storage under time varying electricity prices. In *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 134–140, 2017.

[10] Peter M. van de Ven, Nidhi Hegde, Laurent Massoulié, and Theodoros Salonidis. Optimal control of end-user energy storage. *IEEE Transactions on Smart Grid*, 4(2):789–797, 2013.

[11] Yixing Xu and Chanan Singh. Adequacy and economy analysis of distribution systems integrated with electric energy storage and renewable energy resources. *IEEE Transactions on Power Systems*, 27(4):2332–2341, 2012.

[12] Alessandro Di Giorgio, Francesco Liberati, Andrea Lanna, Antonio Pietrabissa, and Francesco Delli Priscoli. Model predictive control of energy storage systems for power tracking and shaving in distribution grids. *IEEE Transactions on Sustainable Energy*, 8(2):496–504, 2017.

[13] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[14] E Abramova and D W Bunn. Optimal daily trading of battery operations using arbitrage spreads. *Energies*, 14(16):e4931, August 2021.

[15] Taylan Kabbani and Ekrem Duman. Deep reinforcement learning approach for trading automation in the stock market. *IEEE Access*, 10:93564–93574, 2022.

[16] Hao Wang and Baosen Zhang. Energy storage arbitrage in real-time markets via reinforcement learning. pages 1–5, 08 2018.

[17] Ember Climate. European wholesale electricity price data, Jan 2023.

[18] Implementation of the ppo algorithm within the stablebaselines3 package.

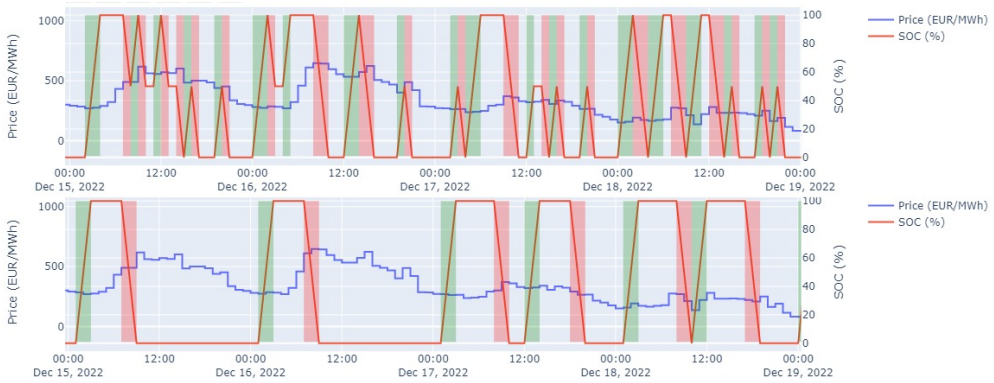


Fig. 5: Charging schedule obtained using LP-OPTIM (top), and using RL (bottom)


Fig. 6: Daily profit distribution following the schedule given by LP-OPTIM (top), LP-PRED (middle), and RL (bottom)

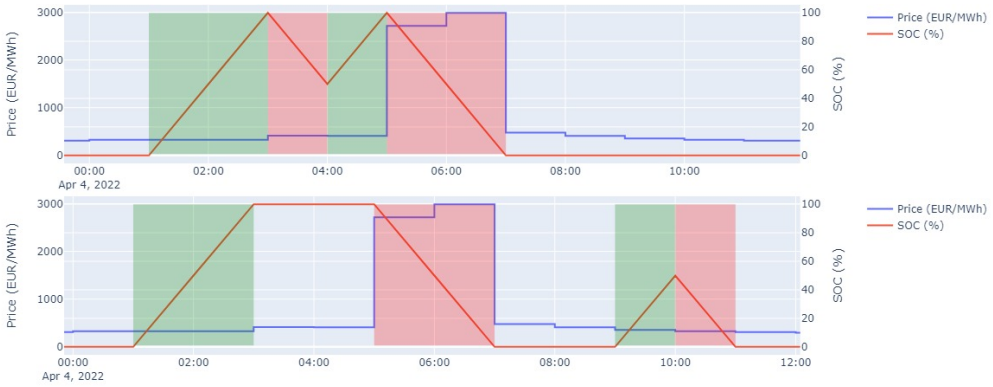


Fig. 7: Charging schedule obtained using LP-PRED (top), and using RL (bottom)