# A Scenario- and Reinforcement Learning-Based BESS Control Strategy for the Frequency Restoration Reserve Market

Jobse, Lennart W.

*Award date:*
2022

PO Box 513
5600 MB Eindhoven
The Netherlands
tue.nl

DEPARTMENT OF MECHANICAL ENGINEERING

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

MASTER THESIS

# A Scenario- and Reinforcement Learning-Based BESS Control Strategy for the Frequency Restoration Reserve Market

L.W. Jobse
1510894
10-03-2022

DEPARTMENT OF MECHANICAL ENGINEERING

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Title: A Scenario- and Reinforcement Learning-Based BESS Control Strategy for the Frequency Restoration Reserve Market
Name: Lennart Jobse
IDNR: 1510894
Section: Electrical Energy Systems
Department: Mechanical Engineering
Program: Sustainable Energy Technology
Thesis supervisor: dr. N. Paterakis
External supervisor: B. Stappers
Date: 23/04/2022

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

# Declaration concerning the TU/e Code of Scientific Conduct for the Master's thesis

I have read the TU/e Code of Scientific Conduct[i].

I hereby declare that my Master's thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct

Date 4/23/22

Name Lennart Jobse

ID-number 1510894

Signature

*Insert this document in your Master Thesis report (2nd page) and submit it on Sharepoint*

[i] See: http://www.tue.nl/en/university/about-the-university/integrity/scientific-integrity/
The Netherlands Code of Conduct for Academic Practice of the VSNU can be found here also.
More information about scientific integrity is published on the websites of TU/e and VSNU

# A Scenario- and Reinforcement Learning-Based BESS Control Strategy for the Frequency Restoration Reserve Market

L.W. Jobse
*Electrical Energy Systems*
*MSc. Sustainable Energy Technology*
*This report was made in accordance with the TU/e Code of Scientific Conduct for the Master thesis*
Supervised by dr. N. Paterakis
L.w.jobse@student.tue.nl
15108942

***Abstract*—Dealing with uncertainty is a key component in the control of Battery Energy Storage Systems (BESSs) on the electricity markets. This study proposes the incorporation of scenario sets as a forecasting component in a reinforcement learning model to optimize a control strategy for a BESS. The proposed model employs proximal policy optimization as its learning algorithm. The scenario sets are created using a class-based method with a long short-term memory neural network as the generative component. The profitability of this method has been evaluated on the Dutch frequency reserve restoration market. Test results indicate that the incorporation of scenario sets increase the performance of the model.**

***Index Terms*—Reinforcement Learning, Scenario Generation, BESS Control Strategy, Frequency Restoration Reserve Market**

## I. Introduction

Managing uncertainty in the electrical power system is becoming a key point of interest for participants in the energy sector. Several disruptive trends have moved this sector to an inflection point. One of these trends is the increase in Renewable Energy Sources (RESs) in the energy mix, making a larger share of energy generation weather-dependent. The intermittent nature of RESs creates uncertainty in supply, requiring more real-time balancing of demand and supply to ensure reliability and a high level of power quality [1]. Another important aspect is the increase in the geographical distribution of electricity generation, in contrast to the conventional centralized electricity grid that is currently in use. This trend demands electricity grids capable of facilitating the distributed generation, requiring advanced control systems at every level of the electricity grid. It is expected that the incorporation of Distributed Energy Systems (DESs), i.e. a combination of generation, storage, energy control, and monitoring solutions, is going to be part of these revisions [2]. However, these developments create more challenges for the operators of the grid as the operating status of DESs is often unknown [3]. Considering these trends, it can be assumed that the uncertainty in the energy sector is going to increase, complicating the reliable operation of the grid [4].

Deviations in frequency are one of the main concerns in the operation of the grid and are caused by mismatches in demand and supply [5]. The widespread application of Battery Energy Storage Systems (BESSs), an essential component of DESs, in the electricity grid has been proposed to decrease the number of frequency deviations [6], [7]. This development is largely hampered by the high capital costs and the uncertainty in the long-term performance of BESSs [8]. Therefore, the identification of a profitable control strategy for a BESS, is key in accelerating the penetration of BESSs into the grid.

In the Netherlands, frequency restoration is controlled by the utilization of a Frequency Restoration Reserve Market (FRRM). This market is focused on trading frequency restoration services. The participation of BESSs in these markets compares favourably to conventional methods, due to the fast discharging and negligible ramping times [9]. The FRRM represents the mismatch between demand and supply, and thus reflects the uncertainty within the electricity grid. As a result, the prices on the FRRM are erratic, complicating the identification of price trends. Developing a profitable control strategy for the participation of a BESS in the FRRM is a challenging problem.

## II. Related Research

A variety of optimization methods have been investigated to create a profitable control strategy for BESSs. Conventionally, the control of the BESS was based on a set of rules. In [10] a storage system is charged and discharged when the State of Charge (SoC) is low and high respectively while minimizing the degradation of the BESS. More recently, genetic and evolutionary algorithms have been implemented to aid in the control of BESSs. These types of algorithms are used for their capability to find good solutions with a relatively high computational efficiency. More information is presented in [11]–[14]. Nowadays, most algorithms used in the optimization of BESS control models are based on Reinforcement Learning (RL). A variety of studies focusing on mitigating overvoltages [15], demand response [16], and the implementation of 5G [17], all utilize RL to control BESS systems. The BESS control problem can be defined as a discrete stochastic control process, allowing it to be modelled as a Markov Decision Process (MDP) [18]. These types of problems are often addressed with RL models. A substantial amount of research is conducted in the optimization of BESSs to support frequency control [19], [20]. Most of these studies attempt to accommodate participation in multiple activities, as shown in [21], [22], where energy arbitrage is combined with frequency control. Most methods employing RL utilize a Multilayer Perceptron (MLP) as the representation of the RL agent, a framework called deep RL [18], [23]. In these cases, the MLP, a type of feed-forward neural network, is used to solve the high dimensional states of the MDP. Some studies utilize a stand-alone Recurrent Neural Networks (RNNs) to generate point forecasts that are presented to the RL in making optimal decisions [24], [25]. In these hybrid models, an RNN is leveraged due to their proficiency in understanding long-term dependencies [26].

The accurate forecasting of the prices of the FRRM or similar markets has been attempted using several methods. In [27] the state transition probabilities of the restoration volumes are generated. Subsequently, using historical data, the imbalance price is inferred from these probabilities. A simpler method was utilized in [28], where a Holt-Winters model is applied to a real-time electricity market. This model employs exponential smoothing based on an average, a trend, and a cyclical variation in a time series. In the last 10 years, more and more studies have utilized Long Short-Term Memory (LSTM) neural networks, an improved version of an RNNs, to forecast loads, generation, and prices in the electricity sector [29], [30]. However, these methods generate point forecasts that are represented by a single summary statistic, often lacking the characterization of uncertainty required for decision-making [31]. Therefore, a scenario set is used to explicitly model the uncertainty in prices on the electricity markets [32]. One method proposed in [33] utilizes an LSTM based network to successfully generate scenarios for the prices on the FRRM. This method utilized class allocation, distribution sampling, and scenario reduction to generate sets of scenarios that adequately represent the uncertainty within the FRRM price.

## III. Objectives and Motivation

The problem central to this work is the development of a profitable control strategy for a BESS to participate in the FRRM. To address this problem, this study attempts to achieve the following objective: *The optimization of a BESS control strategy on the FRRM in terms of profit by an RL model supported by LSTM-generated scenarios.*

The motivation for this study can be derived from two observations. First, a multitude of studies has been performed to optimize the operation of a BESS in terms of profit in an electricity market utilizing a form of RL. Although this aspect is well researched, it has been mostly applied to optimize the profit gained by the utilization of RES by acting as a buffer to the electricity market. In these studies, RL is implemented to cope with the uncertainty in both the intermittent power generation of RESs and the electricity market. Relatively little research is focused on the implementation of RL to maximize profit with a stand-alone BESS participating in the FRRM. In this manner, RL can be utilized to only cope with the uncertainty on the FRRM and the operational limits of the BESS.

Second, the incorporation of point forecasts in RL models to increase the performance is relatively common in the academic world. However, the implementation of scenario sets to achieve the same goal is underexposed. In the case of extremely uncertain and erratic time series, such as the prices on the FRRM, scenarios provide more support in decision-making than point forecasts. Therefore, the combination of RL and forecast scenarios appears to be promising to increase the performance of the control strategy of a BESS.

## IV. Background

This section covers the subjects of interest in the context of this study. It serves as a framework for the proposed methodology by covering the workings of the FRRM and its participants, BESSs, and RL and RNN models.

### A. System of Balancing Markets

Real-time balancing of demand and supply is required to prevent large frequency deviations. In the Netherlands, these are avoided by ensuring capacity to provide downward or upward regulating electricity [34]. This process is facilitated via the FRRM by the Dutch Transmission System Operator (TSO), TenneT B.V.. The participants of the FRRM are defined as Balance Responsible Parties (BRPs) and Balance Servicing Parties (BSPs). BRPs are financially responsible and are obliged to provide an *E-programme* on a daily basis, which corresponds to their net position for the next 24 hours. Utilizing the difference in E-programmes and the real-time demand and supply, the TSO determines the imbalance of each BRP. If an imbalance is observed, three types of reserves are available to be used to solve the imbalance. More information about the types of reserves is given in [35].

Balancing is performed by the TSO, by acting as an artificial market participant that facilitates the sale or purchase of electricity between BRPs and BSPs. If a BRP experiences a deviation from their E-programme, they are required to buy balancing power from or sell balancing power to another BSP. The price of the transaction, the *imbalance price*, is unknown upfront. Parallel to this, the BSPs can participate by placing *imbalance bids* to deliver or purchase an amount of balancing power for a certain price beforehand.

The determination of the imbalance price is based on the price component of the bids placed by the BSP, and the trend in balancing over a *settlement period* of 15 minutes. All price and power components of the imbalance bids in a settlement period are aggregated on an imbalance bid ladder. Based on the total imbalance power needed throughout a settlement period, the imbalance price for buying is the price of the highest activated bid to buy balancing power. Similarly, the imbalance price for selling is the price of the lowest activated bid to sell balancing power. A complete overview of the imbalance pricing mechanism is given in [36].

### *B. Battery Energy Storage System*

In terms of energy storage systems, BESSs are capable of responding fast, with relatively high reliability, and with a relatively easy charging method. All make it well suited for participation in the FRRM compared to pumped or thermal storage [19], [37]. Implementation of BESSs for frequency regulation had been achieved in island power systems over 25 years ago, but implementation in the grid has regained more attention recently [10], [38]. A BESS can be leveraged as a buffer within a BRP's portfolio to limit exposure to balancing costs, or by directly participating on the FRRM to maximize profit. More information about the first application can be found in [7], [18], [39], [40], but is not further investigated in this study. Direct participation in the FRRM can both be performed as a BSP, by placing imbalance bids in advance, or as a BRP, by purposefully deviating from the E-programme. The latter method is investigated in this study and works by providing an empty E-programme for the BESS, creating the possibility to create an artificial imbalance based on developments on the FRRM.

Charging and discharging of a BESS system leads to battery cell degradation, and can thus be defined as a cost from an economic viewpoint [7], [8], [10]. The main degradation mechanism considered in this study is the effect of *depth of discharge*. This parameter is the inverse of the SoC and is key in determining degradation cost in relation to optimal scheduling [8].

### *C. Reinforcement Learning*

Most novel strategies applied to solve the aforementioned problem use RL models. Control of a BESS, combined with the erratic nature of the FRRM, while being constrained by time-dependent battery characteristics, can be defined as a finite MDP. The MDP is composed of state $S$, action $A$, transition probability distribution $P$, reward $R$, and discount factor $y$ (1). Each time-step the agent receives an observation in the form state $S_t$. Subsequently an action is taken and a policy function $\pi(s_t) : P \rightarrow A$ is determined. This function designates the distribution over actions with respect to each state. The transition function $P(s_t, a_t, s_{t+1}) : S \times A \times S$ represents the reinforcement component of the algorithm.

$$M = \{S, A, P, R, y\} \tag{1}$$

RL algorithms are capable of directly optimizing a policy on historical or simulated data while being more adaptable than conventional methods [18]. As a result, a policy by an RL algorithm can be applied in various environments without the need for retraining. These algorithms can be defined as having two characters: the agent and the environment. At each time-step, the state is updated, observed by, and possibly altered by the agent, shown in Fig. 1. The action proposed by the agent, the current, and the new state of the environment, are used to provide the agent with a reward for that action (2). The reward function is critical in guiding the RL learning process and must represent the objective of the agent. At the same time, it must provide space for both exploration and exploitation for the agent.

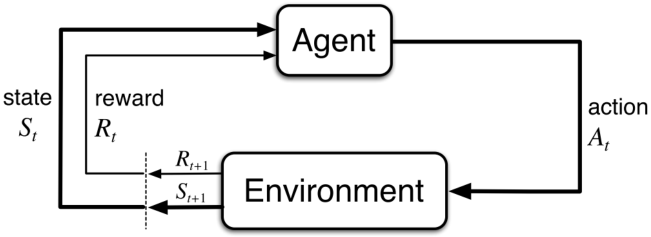


Fig. 1. Schematic of a Reinforcement Learning algorithm [41]

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

The goal of an RL algorithm is to develop an optimal policy $\pi^*$ that maps an action to a state. The general anatomy of an RL algorithm is composed of:

- Sample generation
- Estimation or returns
- Improvement of the policy $\pi$
- Repetition of above steps to find the optimal policy $\pi^*$

In terms of training, there are two main approaches to the learning process of an RL agent: one approach is Q-learning, which employs an objective function, often a Bellman equation, to learn an approximator $Q_\theta(s, a)$ with respect to the optimal action-value function $Q^*(s, a)$. Optimization is performed off-policy as the optimal policy is determined independently of the actions taken by the agent. Another approach is policy gradient optimization, in which a specific representation of the policy $\pi_\theta(a|s)$ is employed. In this stochastic policy, the probability of an action is connected to a given state. Optimization is performed on-policy as updates are carried out based on data gathered while acting based on the most recent policy. Policy gradient methods contain a stochastic gradient ascent algorithm in combination with an estimator of the policy gradient [42]:

$$\hat{g} = \hat{E}_t\big[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{A}_t\big] \tag{3}$$

Where the expectation $\hat{E}_t$ denotes the empirical average over a predetermined batch size. Advantage $\hat{A}_t$ is an estimator of the advantage at $t$ and defined as the discounted sum of rewards compared to a baseline, while $\pi_\theta$ represents a stochastic policy. Implementations of policy gradient methods use an objective function with a gradient that is the policy gradient estimator [42]:

$$L^{PG}(\theta) = \hat{E}_t[\log \pi_\theta(a_t|s_t)\hat{A}_t] \tag{4}$$

The main problem with policy gradient methods is the large, uncontrolled policy updates, destabilizing the training process.

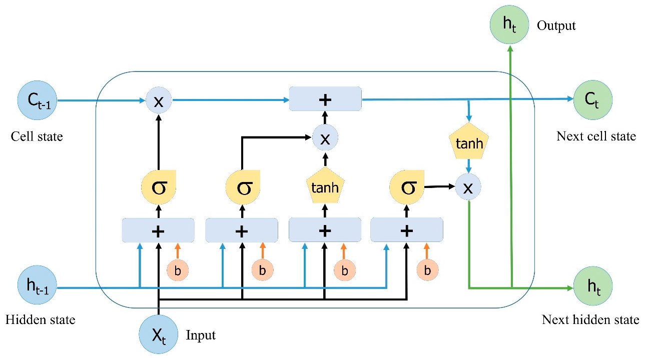


Fig. 2. Schematic overview of the workings of an LSTM cell [44]

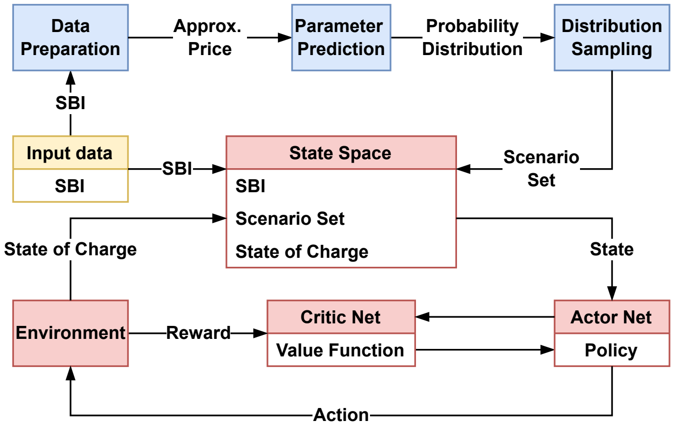


Fig. 3. Schematic overview of the proposed methodology

Therefore, some methods in this family perform optimization by gradient ascent to maximize a performance objective. The A2C and A3C policy algorithms are among them but are not covered in this study. Other methods maximize an approximation of the performance objective. One of the most promising of this family is the Proximal Policy Optimization (PPO) algorithm. PPO is a simplified and more stable version of Trust Region Policy Optimization (TRPO), an algorithm that focuses on limiting parameter updates that substantially change policies. This is performed by constraining the size of the update in policy at each iteration, restricting the exploration of the agent within safe limits while making full use of the available data [18], [42].

### D. *Long Short-Term Memory Units*

RNNs are known to perform well in learning sequential or time-varying patterns. This characteristic is the result of the back-propagation process, where hidden states are updated based on previous iterations (5). LSTM networks have been developed as an improvement over RNN to cope with the exploding gradient problem [43].

$$h_t = \sigma(Wx_t + Uh_{t-1} + b) \tag{5}$$

The main feature of the LSTM unit is the capability to filter information based on usefulness. This 'forgetting' characteristic partly combats the vanishing and exploding gradients during training. A schematic overview of the LSTM is given in Fig. 2. The input $x_t$ and the hidden state $h_{t-1}$ are used to add or remove information from the cell state $C_{t-1}$. This is used to generate the next cell state $C_t$, which is combined with information from the input $x_t$ and the hidden state $h_{t-1}$ to yield the next hidden state $h_t$ [44]. By utilizing this system of gates, the cell can determine the information to be important or obsolete, in the context of the learning purpose.

## V. Proposed Methodology

This section details the proposed methodology to achieve the objective. The methodology can be categorized into an LSTM scenario generation component and an RL component, denoted by the blue and red shapes respectively in Fig. 3. The objective of the to be implemented RL model is the maximization of the cumulative reward. The LSTM scenario generation element attempts to generate a scenario set that encompasses the trend of the imbalance prices. It is hypothesized that the incorporation of scenario set $Z_t^{IP}$ into the RL model improves the agent's capability in finding the optimal long-term control strategy. The following sections provide a more in-depth overview of both components of the methodology.

### A. *Reinforcement Learning Structure*

The RL model used in this methodology can be defined as an MDP in terms of a 5-element tuple (1). The state space, action space, and reward function of this MDP and the utilized RL algorithm are covered in the following sections.

*1) State Space:* The state space is the environment used for observation by the agent to optimize the reward. This space is updated at each time-step $t$, set at 1 minute, and contains the System Balance Information (SBI). The elements of the SBI are shown in (7). $p_t^{dev,feed}$ and $p_t^{dev,take}$ are the prices of feed and take price-setting bids respectively. $E_t^{reg,down}$ and $E_t^{reg,up}$ represent the quantities of down- and up-regulating capacity respectively. This data is presented in the array $SBI_t \in \mathbb{R}^{4\times T^{window,rl}}$, where $T^{window,rl}$ denotes the number of preceding time-steps of the SBI. Besides the SBI, the imbalance price scenario set $Z_t^{IP}$ and the SoC $soc_t$ are also included into the state space (6). As the SBI is obtained from the TSO with a delay of 3 minutes, the data is presented with a lag of 3 time-steps, to force the agent to take actions based on lagged data.

$$S_t = [SBI_t, Z_t^{IP}, soc_t] \tag{6}$$

$$SBI_t = [p_t^{dev,feed}, p_t^{dev,take}, E_t^{reg,down}, E_t^{reg,up}] \tag{7}$$

*2) Action Space:* The actions space contains the possible actions the agent can take at each time-step. This model utilizes a continuous actions space $A \in [-1, 1]$. Action $A$ represents a fraction of the rate of charge $roc$ and is used to find the transaction amount of electricity $E_t^{tran}$ in MWh on a 1-minute basis. Charging and discharging are denoted by a positive and negative action value respectively. The lower and higher SoC limits $soc^{min}$ and $soc^{max}$ are used to define the operational limits of the BESS. If the proposed action results in an SoC within these limits $E_t^{tran}$ is calculated with (8). $E_t^{tran}$ is set to 0 if the proposed action leads to an SoC outside the operational limits . The SoC is calculated using $soc_t = \frac{Q_t}{Q_n}$, where $Q_n$ and $Q_t$ denote the nominal and available battery capacity respectively. The updated available capacity $Q_{t+1}$ is calculated with (9). In (9) $\eta^{bess}$ denotes the charging and discharging efficiency. It is assumed that an action proposed by the agent is always facilitated by the TSO, and thus cleared instantly.

$$E_t^{tran} = A_t \times \frac{roc}{60} \tag{8}$$

$$Q_{t+1} = \begin{cases} Q_t + \eta^{bess} \times E_t^{tran} & A_t > 0 \\ Q_t + \frac{E_t^{tran}}{\eta^{bess}} & A_t < 0 \end{cases} \tag{9}$$

*3) Reward Function:* The reward function is the incentivization mechanism steering the agent towards the optimal solution by rewarding effective actions while punishing ineffective actions. The reward function used in this study is based on an auxiliary imbalance price $p^{imb,aux}$. The proposed reward derives an auxiliary price from the total cost $C_t$ of the energy stored in the BESS. Subsequently, the reward represents the difference between the auxiliary price and the imbalance price (10). Here, the imbalance prices for taking from or feeding to the grid are denoted by $p_t^{imb,take}$ and $p_t^{imb,feed}$ respectively. The auxiliary price is calculated using (11), with $C_t$ calculated using (12). Of importance is the utilization of the imbalance price in this calculation, which is unknown upfront. To further simulate this uncertainty, the imbalance price is not provided in the state space. The RL agent is thus undertaking actions without knowing the financial gains.

$$r_t^{fin} = \begin{cases} (p_t^{imb,aux} - p_t^{imb,take}) * E_t^{tran} & A_t > 0 \\ (p_t^{imb,feed} - p_t^{imb,aux}) * |E_t^{tran}| & A_t < 0 \end{cases} \tag{10}$$

$$p_t^{imb,aux} = \frac{C_t}{Q_t} \tag{11}$$

$$C_{t+1} = \begin{cases} C_t + p_t^{imb,take} * E_t^{tran} & A_t > 0 \\ C_t + p_t^{imb,aux} * E_t^{tran} & A_t < 0 \end{cases} \tag{12}$$

*4) Reinforcement Learning Algorithm:* This study employs a policy gradient-based algorithm, as they have a strong theoretical convergence compared to Q-learning algorithms [45]. Policy gradient methods have difficulty with large policy updates and have poor data efficiency [42]. As an improvement, PPO was developed to constrain the policy updates. There are two methods this can be performed: a clipped objective or an adaptive Kullback-Leiber penalty. The PPO algorithm utilized in this study used the clipped objective. The objective function of clipped PPO can be defined as [42]:

$$L^{clip}(\theta) = \hat{E}_t[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \tag{13}$$

With probability ratio $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ and $\epsilon$ denoting the clipping parameter. Using this objective, the algorithm is deterred to move $r_t$ outside of boundaries set by the clipping parameter. The PPO algorithm is built on an actor-critic method to improve learning performance. This method employs an actor and a critic model. The actor model represents the policy and learns the ideal actions based on the given state. The subsequent reward is fed into the critic model, which evaluates the action based on the current policy and expected returns. The main parameters for PPO are the clipping parameter and the learning rate. Important parameters for the RL model are the discount factor $\gamma$, which determines the importance of future rewards, while $\lambda$ is the smoothing factor to reduce variance in training. Moreover, MLPs are used to represent both the actor and the critic net, each with a set of layers, and each containing a number of neurons.

### *B. Scenario Generation*

This section covers the proposed methodology for an LSTM scenario generation model. This methodology is largely derived from [33] without the scenario reduction component. The method in utilized this study can be subdivided into 3 components:

- Data Preparation
- Parameter Prediction
- Distribution Sampling

These three components produce a scenario set $Z_t^{IP}$ of the imbalance prices, to be incorporated into the state $S_t$. The following section elaborates on each of the components.

*1) Data Preparation:* To reduce complexity, the SBI is used to define an approximation of the imbalance prices $p^{app,feed}$ and $p^{app,take}$ based on the mechanics presented in [36]. This step is required as the predicted variable needs to be added to the input data to generate the subsequent prediction in the distribution sampling step. Scenario generation requires a transformation to a classification problem, as the final output needs to be a probability distribution to sample from. Therefore, the approximated imbalance prices are transformed into categorical data. This is performed by allocating each of the values to a bin with bin size $N^{bin}$. To decrease the total number of bins, the values are clipped before the allocation. These first steps are performed on both the input and target variables. The target variables are further transformed by employing one-hot encoding resulting in a tensor $y_t \in^{N^{features} \times N^{bin}}$. Here $N^{features}$ denotes the number of features. The input data is to be presented in the form of a tensor $x_t \in \mathbb{R}^{T^{window,lstm} \times N^{features}}$, where $T^{window,lstm}$ denotes the number of previous time-steps presented to the model.

*2) Parameter Prediction:* An LSTM network is used to predict the distribution over $N^{bin}$ classes for one time-step in the future. The input tensor $x_t$ is reshaped to $1 \times T^{window,lstm} \times N^{features}$ as the network requires 3 dimensional tensors. This tensor is processed through the neural network, which assigns weights to temporal dependencies. The depth of the network is increased by stacking layers composed of a number of LSTM neurons, making it better suited to process complex features. The output of the final layer, the most recent hidden state $h_t$, is subsequently fed into a number of dense layers to change the shape to the desired dimension $N^{features} \times N^{bin}$. To accommodate the classification of the labels a soft-max function is utilized as the last layer of the network. This function produces a probability distribution over all classes.

*3) Distribution Sampling:* The probability distribution derived from the soft-max layer is used to sample one prediction $\hat{y}_{t+1}$. This sampled prediction is added to the input sequence $x_{t+1}$ to produce a new probability distribution, from which a new prediction $\hat{y}_{t+2}$ is sampled. Resulting in a scenario $\{\hat{y}_{t+1}, \hat{y}_{t+2}, ...., \hat{y}_{T^{horizon}}\}$, where $T^{horizon}$ denotes the length of the forecast horizon. This whole process is reiterated a total of $N^{cardinality}$ times, to give a scenario set of imbalance prices $Z_t^{IP} \in \mathbb{R}^{N^{cardinality} \times T^{horizon}}$.

## VI. Evaluation

To assess the proficiency and performance of the proposed method, it is evaluated utilizing a study case. The following sections cover the input data, the RL and LSTM model characteristics, and the evaluation methods and metrics.

### *A. Input Data*

Both the RL and LSTM model utilize the SBI as input data (7). For the RL model, data from the complete month of January in 2019 is used to train the model, resulting in a total of 46440 time-steps. The same data is also used for the tuning of the model. To display long-term trends in the SBI, $T^{window,rl}$ is set at 60, resulting in a dimensionality of $4 \times 60$ for $SBI_t$. Scaling between -1 and 1 is applied to accommodate for the activation function in the MLPs used to represent the actor and the critic nets in the RL model.

For scenario generation, SBI data of the complete year of 2018 is used for training, resulting in 525600 time-steps. The tuning data covers the year 2019. Data processing is performed according to the method presented in Section V-B1. First, $p^{app,feed}$ and $p^{app,take}$ are clipped between -50 €/MWh and 200 €/MWh. The data is allocated in bins, with $N^{bin}$ set at 50 bins. Subsequently, $T^{window,lstm}$ is set at 60 time-steps, resulting in a 3D input tensor $x_t$ with a dimension of $1 \times 60 \times 2$. The target array is one-hot encoded to create a 3D target tensor $y_t$ with the dimension of $1 \times 2 \times 50$. The input tensor is scaled between -1 and 1, to accommodate the activation functions used in the LSTM network.

### *B. Reinforcement Learning Model Details*

The agent algorithm implemented is the PPO algorithm. The actor and critic are represented by MLPs. Both MLPs consist of 4 layers, each containing 512 neurons, utilizing the ReLU activation function. The discount factor is set at $\gamma = 0.97$ and smoothing factor $\lambda = 0.04$. The clip parameter is set at 0.2 and the learning rate is $5 \times 10^{-5}$. Optimal hyperparameters are determined using a grid search over the training data. The agent is trained for a total of 80 epochs and evaluated based on $\Pi^{profit}$ (14) over the training data set. An overview of the parameters used in the environment is given in Table I. The presented BESS coefficients have been derived from [46]. The upper and lower SoC limits are conservatively chosen to ensure the durability of the BESS. Starting value for the available capacity $Q_0$ is set at 0.1 MWh. The RL model is implemented using OpenAI and ElegantRL [47], [48].

TABLE I. Overview of used BESS parameters

| Parameter name | Symbol | Value | Unit |
|---|---|---|---|
| Efficiency | $\eta^{bess}$ | 0.95 | - |
| State of Charge limit | $soc^{max}$ | 0.85 | - |
| State of Charge lower limit | $soc^{min}$ | 0.15 | - |
| Nominal capacity | $Q_n$ | 0.2 | MWh |
| Rate of Charge | $roc$ | 0.1 | MW |
| Capital cost | $C^{bess}$ | 60000 | € |
| Degradation coefficient 0 | $\beta_0$ | 4901 | - |
| Degradation coefficient 1 | $\beta_1$ | 1.98 | - |
| Degradation coefficient 2 | $\beta_2$ | 0.016 | - |

### *C. Scenario Generation Network Details*

The LSTM network consists of; 2 LSTM layers, 1 dense layer, concluded by a SoftMax layer to gain a probability distribution over the 50 bins. The first 2 LSTM layers each contain 256 LSTM cells with a tan activation function. The 3 dense layers represent the final probability distribution and thus contain $2 \times 50$ cells, and both employ ReLU activation functions. Categorical Cross-Entropy is used as the loss function. Compilation of the model is performed with the Adam optimizer [49], with accuracy as the evaluation metric. The model is trained for 60 epochs with a starting learning rate of $1 \times 10^{-4}$. In addition, the learning rate is halved when a performance plateau is detected based on the accuracy of the model on the tuning data. Optimal hyperparameters were determined using a grid search. The model is built and trained with the Tensorflow library [50].

### *D. Evaluation Components*

Evaluation of the complete model is conducted by evaluating the performance of the RL agent over the month of January 2020. The agent is evaluated on the metrics presented in the following section. An agent trained with a state in which the forecasting element $Z_t^{IP}$ is omitted is used as a baseline model (RL-BASE). The evaluation consists of three components:

- A benchmarking element to indicate the performance of the model compared to an industry-standard benchmark.
- A comparative element to evaluate the impact of the incorporation of the forecasting component compared to the baseline model. The impact of cardinality and the forecast horizon is also evaluated.
- A comparative element to evaluate the impact of an altered reward function on the performance of the baseline model.

### E. Evaluation Metrics

Two metrics are used for evaluation; a market performance and a BESS cost metric. For market performance, the daily market profit $\Pi_t^{profit}$ is used for evaluation:

$$\Pi_t^{profit} = \sum_t^{T_{day}} \begin{cases} p_t^{imb,take} * E_t^{tran} & A_t > 0 \\ p_t^{imb,feed} * E_t^{tran} & A_t < 0 \end{cases} \tag{14}$$

The BESS cost metric is based on the relation between the SoC and the number of life cycles $N^{cycle}$ of a BESS. This relation can be defined by a curve-fitting function (15), where $\beta_0$, $\beta_1$ and $\beta_2$ denote coefficients utilized for curve-fitting [8]. Utilizing this relation, the daily BESS cost can be derived with (16), in which $C^{bess}$ denotes the capital cost of the BESS. The BESS cost is only assigned when discharging and is related to the difference between $N^{cycle}_{soc_{t+1}}$ and $N^{cycle}_{soc_t}$. This method is derived from [8]. Battery calendar life is not considered, as it is mainly influenced by locational circumstances, and can thus be regarded as a non-operational factor to this study.

$$N^{cycle}_{soc} = \beta_0 * (1 - soc_t)^{-\beta_1} * \exp(\beta_2 * soc_t) \tag{15}$$

$$\Pi^{bess} = \sum_t^{T_{day}} \begin{cases} 0 & A_t > 0 \\ \frac{C^{bess}}{N^{cycle}_{soc_{t+1}} - N^{cycle}_{soc_t}} & A_t < 0 \end{cases} \tag{16}$$

### F. Benchmarking Evaluation

A rolling intrinsic benchmark (BM-LP) is used to give an indication of the optimal performance of a model based on Linear Programming (LP). Only $p_t^{imb,feed}$ is considered, and $\eta^{bess}$ is omitted to reduce the complexity of the LP optimization algorithm. The algorithm maximizes the revenue $P_t$ over period $T_{bench}$ based on the starting battery capacity $Q_0^{lp}$ and the $p_t^{imb,feed}$. The obtained action $A_t^{lp}$ is subsequently used to determine $Q_0^{lp}$ with (9) for the next iteration of the algorithm. The model is constrained to the same BESS constraints as the RL agent given in Table I. An overview of the algorithm is given in Algorithm 1, where $P$ is maximized (17), while constrained by (18) with $Q_{t+1}$ defined in (19). $T^{bench}$ is set at 60 minutes, corresponding to 4 settlement periods. Moreover, a rule-based model is also implemented as a benchmark (BM-RULE). Equation 20 displays the method, where action $A_t^{rule}$ is determined based on the feed imbalance bid price $p^{dev,feed}$ and the rolling average over 60 time-steps of the same price $\sigma^{dev,feed}$.

$$A_t^{rule} = \begin{cases} -1 & p^{dev,feed} > \sigma^{dev,feed} \\ 1 & p^{dev,feed} < \sigma^{dev,feed} \end{cases} \tag{20}$$

### G. Forecasting Evaluation

This part concerns the evaluation of the impact on the performance from the inclusion of the generated scenarios on the RL agent. Therefore, the RL agent supported by a scenario set with $T^{horizon} = 10$ and $N^{cardinality} = 10$ (RL-SG) is compared to the baseline model. Moreover, RL-SG is also compared to an RL agent supported by a more conventional regression-based sequence forecasting method (RL-ENC). This method employs an LSTM encoder-decoder model to produce point forecasts for the subsequent 10 time-steps. Apart from the encoding-decoding component, the model employs the same number of LSTM neurons and layers. Due to the model being regression-based, mean squared error is used as the training metric, and the softmax layer is omitted. The BESS cost is not considered in this comparison, as only the forecasting component is evaluated.

**Algorithm 1** Rolling intrinsic optimization over a finite horizon at a single time-step

```
Input:
  Feed imbalance price $p_t^{imb,feed}$
  State of Charge upper limit $soc^{max}$
  State of Charge lower limit $soc^{min}$
  Rate of Charge $roc$
  Initial battery capacity $Q_0^{lp}$
  Nominal battery capacity $Q_n$
  Benchmark Horizon $T^{bench}$
Output:
  Profit $P$
  Action $A_t^{lp}$
  Procedure:
  Solve:
      $P = \max \sum_{t=0}^{T^{bench}} -A_t^{lp} \times \frac{roc}{60} \times p_t^{imb,feed}$    (17)
  Constrained by:
      $soc^{min} <= \frac{Q_{t+1}^{lp}}{Q_n} <= soc^{max}$    (18)
  With:
      $Q_{t+1}^{lp} = Q_t^{lp} + (\frac{roc}{60} \times A_t^{lp})$    (19)
  End Procedure
```

$$r_t^{alt} = r_t^{fin} - r_t^{bess} \tag{21}$$

$$r_t^{bess} = \begin{cases} 0 & A_t > 0 \\ \frac{C^{bess}}{(N^{cycle}_{soc_{t+1}} - N^{cycle}_{soc_t})} & A_t < 0 \end{cases} \tag{22}$$

### H. Reward Function Evaluation

An RL agent is trained with an altered reward function (21), with the degradation cost $r_t^{bess}$ calculated using (22). The goal of the altered reward function is the maximization of profit while minimizing BESS cost (RL-BAT). In this evaluation the adjusted profit metric $\Pi^{adj} = \Pi^{profit} + \Pi^{bess}$ is utilized.

TABLE II. Overview of models used in the evaluation

| Model name | Model type | Forecasting element | Reward function |
|---|---|---|---|
| RL-BASE | RL | None | Financial reward (10) |
| RL-SG | RL | Scenario set | Financial reward |
| RL-ENC | RL | Point forecast | Financial reward |
| RL-BAT | RL | None | Altered reward (21) |
| BM-LP | LP | N/A | N/A |
| BM-RULE | Rule-based | N/A | N/A |

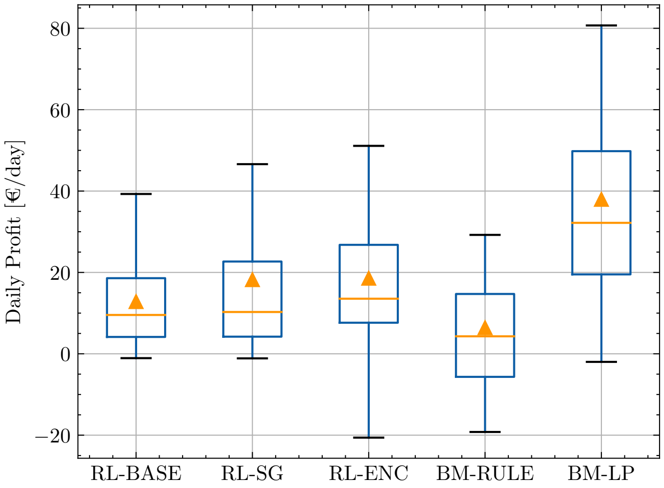


Fig. 4. Comparison of models on daily profits in €/day

## VII. Test Results

The results of both the benchmarking and the comparative evaluations are presented in this section. An overview of the used models is given in Table II.

### A. Benchmarking Evaluation Results

A comparison of the performance of BM-LP, RL-SG, and BM-RULE is displayed in Fig. 4. Here, a boxplot is shown of the daily profits, where the orange triangle denotes the mean daily profits. BM-LP outperforms RL-SG, generating 108% more in terms of mean daily profits. BM-RULE performs worse than RL-SG, generating 65% less in mean daily profits. The charging profile is given in Fig. 5, where the SoC over time of BM-LP and RL-SG are shown by the green and blue plots respectively. A clear difference in charging profile between RL-SG and BM-LP is visible. It can be observed that the SoC of RL-SG is consistently high. The SoC profile of BM-LP is more balanced as the whole capacity of the BESS is utilized.

### B. Forecasting Evaluation Results

The evaluation of the various models supported by a forecasting or trend indicating element compared to RL-BASE is presented in this section. The performance of both RL-SG and RL-ENC is relatively similar, and offers an improvement of 42% and 45% respectively over RL-BASE. Fig. 4 shows that both RL-ENC and RL-SG have more variance in daily profits compared to RL-BASE, with RL-SG showing a longer lower whisker. Regarding the charging profile, shown in Fig. 5, a clear difference is observed between RL-SG and RL-BASE. A predominantly low SoC is visible for RL-BASE, with little to no overlap with RL-SG and BM-LP. Fig. 6 shows a comparison of performance for various scenario forecast horizons and cardinalities. Fig. 6a shows the impact of $N^{cardinality}$ with $T^{horizon} = 10$. The best performing cardinality is 20, although the difference in performance is small. Fig. 6b shows the impact of $T^{horizon}$, while $N^{cardinality} = 10$. It appears that an increase in the forecast horizon does not yield a performance improvement.

TABLE III. Comparison of the daily market profit, BESS cost, and adjusted profit in mean €/day

| Model name | Market profit | BESS cost | Adjusted profit |
|---|---|---|---|
| RL-BASE | 12.79 | 34.99 | -22.20 |
| RL-BAT | 11.37 | 1.21 | 10.16 |

### C. Reward Function Evaluation Results

A comparison between RL-BASE and RL-BAT employing adjusted daily profit $\Pi^{adj}$ is presented in Table III. RL-BAT is outperforming RL-BASE in terms of mean adjusted daily profit by a large margin. For RL-BAT, a small decrease in daily profit is compensated by a large decrease in daily BESS cost, resulting in a significantly higher daily adjusted profit.

## VIII. Discussion

### A. Results Interpretation

Regarding the benchmarking evaluation, it is clear that the BM-LP model is more capable of achieving constant high daily market profits compared to RL-SG. This is to be expected considering the model knows the imbalance price for the subsequent hour. For the charging profile, a clear contradiction between BM-LP and RL-SG is visible. It appears that RL-SG focuses on discharging at a high imbalance price while making sure that SoC is kept high. This strategy ensures that a charge is available to make a profit when the imbalance price is high. BM-LP makes use of both high prices to discharge and low prices to charge.

In the context of the performance of the forecasting component, it was observed that RL-SG outperformed RL-BASE. RL-SG and RL-BASE show contradicting charging profiles with a high and low mean SoC respectively. This could indicate that RL-BASE focuses on low imbalance prices to charge profitably, while RL-SG focuses on high imbalance prices to discharge profitably. Furthermore, lengthening the forecast horizon above 10 did not yield an increase in the performance of RL-SG in this study. This could be the result of the uncertainty in prediction over many time-steps. Increasing the cardinality above 10 did appear to yield better performance, albeit a small improvement. The increase in performance must be considered in the light of longer training times, a result of the higher state dimensions.

Interestingly, RL-ENC performs better than RL-SG. Still, it must be noted that RL-SG appears to show more consistency

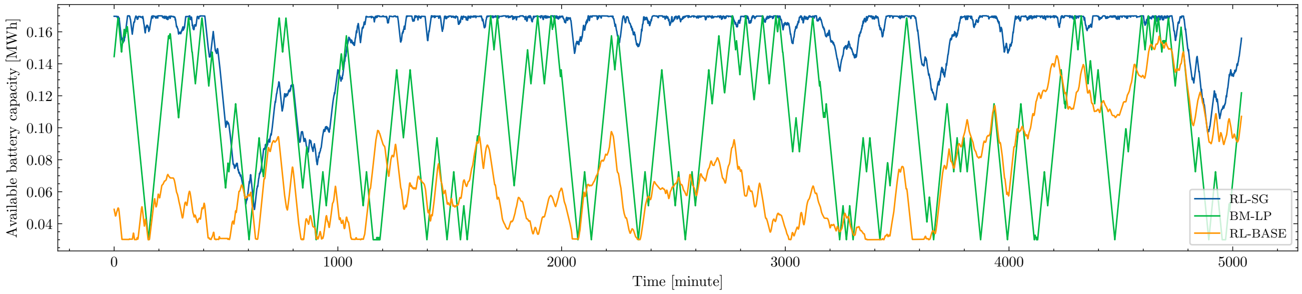
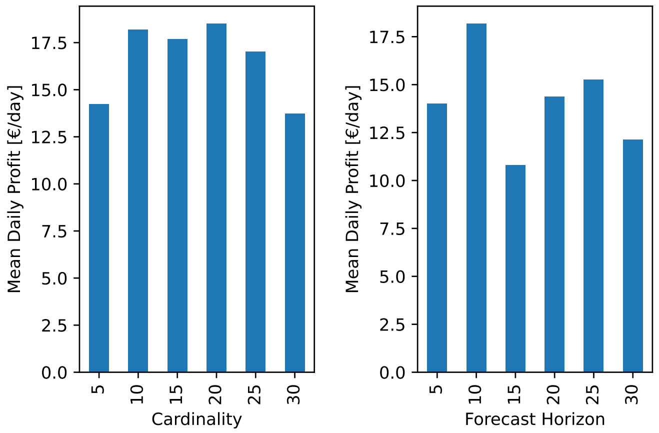


Fig. 5. Comparison of charging profiles



(a) Impact of cardinality with $T^{horizon} = 10$

(b) Impact of forecast horizon with $N^{cardinality} = 10$

Fig. 6. Evaluation of the impact of cardinality and forecast horizons on mean daily profits in €/day

in achieving positive daily profits compared to the larger variation in daily profits of RL-ENC. The dimensions of the state space are much lower for RL-ENC, resulting in faster and more stable training compared to RL-SG. An increase in state space increases the complexity of the optimal policy determination problem [51].

As for the reward function evaluation, RL-BAT showed an improvement in adjusted daily profits. Therefore, the inclusion of $r_t^{bess}$ into the reward function seems to yield the expected results and thus could improve the long-term performance of the BESS.

### B. Constraints and Limitations

The constraints and limitations cover five points of interest. The most pertinent point is the utilized model structure, where the output of one model is utilized as the input for another model. The LSTM model central to the SG component of this study is optimized to maximize the accuracy of the forecasted imbalance price, while the RL model is optimized to maximize the profit. This mismatch in optimization goals leads inherently to complications in training. This problem also impacts the robustness and reproducibility of the result.

The incorporation of the scenario set in the state increases the state dimensions, resulting in increased training times. This could be solved by applying a scenario reduction method, minimizing the dimensions of the scenario set. A simple reduction method was tested but showed inadequate results. This method defined a scenario set as by the mean, variance, skewness, and kurtosis of that set. A more advanced scenario reduction method could improve the performance of scenario sets with high cardinality.

The generation of scenarios is a computationally heavy task. This study was unable to reduce this computational strain, resulting in the evaluation of scenario sets with relatively small dimensions. An increase in cardinality or forecasting horizon could yield better results, but have not been tested in this study. To further confirm the findings in this study, the models must be trained and evaluated over longer periods. Parallelization could be leveraged to increase the efficiency in training and scenario generation.

The minimization of the BESS cost proves difficult with the intrinsic rolling benchmark method, due to the non-linear nature of (15). This could be solved by linearization of this function but has not been attempted in this study. Therefore, BM-LP was not considered in the reward function evaluation. Furthermore, Algorithm 1 considered only one of the imbalance prices. For the evaluation period, $p_t^{imb,feed}$ and $p_t^{imb,take}$ are the same in 91% of the cases. Therefore, the benchmark could perform better if both prices are considered.

Hyperparameter tuning was performed on the RL-SG model but was not repeated when the cardinality or the time horizon was altered. This could explain the unstable training process over long periods when the state space was significantly increased. The performance of RL-BASE could also be further optimized as it also has been evaluated using these hyperparameters. Hyperparameter tuning was not repeated between models to keep the differences between the compared models as small as possible.

### C. Recommendations

Future studies could consider creating an RL model in which the actor and critic nets are represented by LSTM networks. The scenario set is sequential in nature and therefore, the LSTM network could improve the agent's understanding of the presented scenario set.

A clear advantage of the RL-ENC model is the possibility to incorporate external features as the input for the LSTM encoder model, possibly improving its performance. Scenario generation requires the same features for both input and output data, as the predicted parameters are used to generate a prediction for the subsequent time-step. Therefore, more experiments with the RL-ENC model should be performed to explore its potential.

The RL model could be expanded to accommodate RESs. The intermittent power generation of RES is similar in nature to the FRRM regarding uncertainty. The inclusion of scenario sets of RES power generation could improve the performance of the expanded RL model. Nevertheless, more tests must be conducted with an expanded RL model to prove the possible advantages of this concept.

Further improvements in the BESS degradation model could be made by incorporating the minimization of charging cycles into the reward function. Furthermore, a dynamic rate of charge based on the SoC could be implemented to describe charging mechanics more accurately.

## IX. Conclusion

This study proposed a methodology to incorporate scenario sets in a Reinforcement Learning (RL) model to optimize a Battery Energy Storage System (BESS) control strategy on the frequency reserve restoration market. A class-driven scenario generation method based on Long Short-Term Memory (LSTM) networks is implemented to aid in identifying temporal dependencies. The results indicate that the incorporation of scenario sets improves the performance of the RL in terms of daily profits. Increases in the forecast horizon and cardinality of the scenario sets do not appear to yield a significant performance improvement. More tests with longer evaluation periods need to be performed to confirm these findings. The results from the implementation of a BESS degradation cost in the reward function appear to show that profit can be kept at a similar level while reducing battery degradation.

In the context of future research, one interesting direction of inquiry would be the application of scenario reduction to decrease the dimensions of the scenario set. This would reduce the state dimensions of the RL model, reducing training times while increasing stability in training.

## Acknowledgement

This work was supported by Scholt Energy B.V., with B. Stappers as an external advisor. B. Stappers is also with Scholt Energy B.V..

## References

[1] C. Kockel, L. Nolting, J. Priesmann, and A. Praktiknjo, "Does renewable electricity supply match with energy demand? – a spatio-temporal analysis for the german case," *Applied Energy*, vol. 308, 2022, Art. no. 118226.

[2] K. Warren, R. Ambrosio, B. Chen, Y. H. Fu, S. Ghosh, D. Phan, M. Sinn, C. H. Tian, and C. Visweswariah, "Managing uncertainty in electricity generation and demand forecasting," *IBM Journal of Research and Development*, vol. 60, no. 1, pp. 8:1–8:13, 2016.

[3] Y. Zhang, J. Wang, and Z. Li, "Uncertainty modeling of distributed energy resources: techniques and challenges," *Current Sustainable/Renewable Energy Reports*, vol. 6, no. 2, pp. 42–51, 2019.

[4] H. Yi, M. H. Hajiesmaili, Y. Zhang, M. Chen, and X. Lin, "Impact of the uncertainty of distributed renewable generation on deregulated electricity supply chain," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6183–6193, 2018.

[5] J. W. Taylor and M. B. Roberts, "Forecasting Frequency-Corrected Electricity Demand to Support Frequency Control," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 1925–1932, 2016.

[6] N. Günter and A. Marinopoulos, "Energy storage for grid services and applications: Classification, market review, metrics, and methodology for evaluation of deployment cases," *Journal of Energy Storage*, vol. 8, pp. 226–234, 2016.

[7] J. Tan and Y. Zhang, "Coordinated Control Strategy of a Battery Energy Storage System to Support a Wind Power Plant Providing Multi-Timescale Frequency Ancillary Services," *IEEE Transactions on Sustainable Energy*, vol. 8, no. 3, pp. 1140–1153, 2017.

[8] J. O. Lee and Y. S. Kim, "Novel battery degradation cost formulation for optimal scheduling of battery energy storage systems," *International Journal of Electrical Power and Energy Systems*, vol. 137, 2022, Art. no. 107795.

[9] Y. J. A. Zhang, C. Zhao, W. Tang, and S. H. Low, "Profit-maximizing planning and control of battery energy storage systems for primary frequency control," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 712–723, 2018.

[10] B. Xu, A. Oudalov, J. Poland, A. Ulbig, and G. Andersson, "Bess control strategies for participating in grid frequency regulation," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 4024–4029, 2014.

[11] M. Faisal, M. Hannan, P. J. Ker, M. A. Rahman, R. Begum, and T. Mahlia, "Particle swarm optimised fuzzy controller for charging–discharging and scheduling of battery energy storage system in mg applications," *Energy Reports*, vol. 6, pp. 215–228, 2020.

[12] T.-Y. Lee, "Operating schedule of battery energy storage system in a time-of-use rate industrial user with wind turbine generators: a multipass iteration particle swarm optimization approach," *IEEE Transactions on Energy Conversion*, vol. 22, no. 3, pp. 774–782, 2007.

[13] H.-S. Lee, B.-G. Koo, S.-W. Lee, W. Kim, and J.-H. Park, "Optimal control of bess in microgrid for islanded operation using fuzzy logic," in *2014 5th International Conference on Intelligent Systems, Modelling and Simulation*, 2014, pp. 468–473.

[14] K. H. Chua, Y. S. Lim, and S. Morris, "A novel fuzzy control algorithm for reducing the peak demands using energy storage system," *Energy*, vol. 122, pp. 265–273, 2017.

[15] M. Al-Saffar and P. Musilek, "Reinforcement learning-based distributed bess management for mitigating overvoltage issues in systems with high pv penetration," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 2980–2994, 2020.

[16] Y. Tao, J. Qiu, and S. Lai, "A hybrid cloud and edge control strategy for demand responses using deep reinforcement learning and transfer learning," *IEEE Transactions on Cloud Computing*, vol. 10, no. 1, pp. 56–71, 2021.

[17] H. Yuan, G. Tang, D. Guo, K. Wu, X. Shao, K. Yu, and W. Wei, "Bess aided renewable energy supply using deep reinforcement learning for 5G and beyond," *IEEE Transactions on Green Communications and Networking*, 2021.

[18] B. Huang and J. Wang, "Deep-Reinforcement-Learning-Based Capacity Scheduling for PV-Battery Storage System," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2272–2283, 2021.

[19] Y. Dong, Z. Dong, T. Zhao, and Z. Ding, "A strategic day-ahead bidding strategy and operation for battery energy storage system by reinforcement learning," *Electric Power Systems Research*, vol. 196, 2021, Art. no. 107229.

[20] Z. Yan, Y. Xu, Y. Wang, and X. Feng, "Deep reinforcement learning-based optimal data-driven control of battery energy storage for power system frequency support," *IET Generation, Transmission & Distribution*, vol. 14, no. 25, pp. 6071–6078, 2020.

[21] B. Cheng and W. B. Powell, "Co-optimizing battery storage for the frequency regulation and energy arbitrage using multi-scale dynamic programming," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1997–2005, 2016.

[22] 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, 2021, Art. no. 8365.

[23] 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.

[24] F. Chang, T. Chen, W. Su, and Q. Alsafasfeh, "Control of battery charging based on reinforcement learning and long short-term memory networks," *Computers and Electrical Engineering*, vol. 85, 2020, Art. no. 106670.

[25] 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, 2021, Art. no. 102868.

[26] S. Muzaffar and A. Afshari, "Short-term load forecasts using lstm networks," *Energy Procedia*, vol. 158, pp. 2922–2927, 2019.

[27] J. Dumas, I. Boukas, M. M. de Villena, S. Mathieu, and B. Cornélusse, "Probabilistic forecasting of imbalance prices in the belgian context," in *2019 16th International Conference on the European Energy Market (EEM)*, Ljubljana, Slovenia, Sept. 18-20, 2019.

[28] T. Jónsson, P. Pinson, H. A. Nielsen, and H. Madsen, "Exponential smoothing approaches for prediction in real-time electricity markets," *Energies*, vol. 7, no. 6, pp. 3710–3732, 2014.

[29] Z. Chang, Y. Zhang, and W. Chen, "Effective adam-optimized lstm neural network for electricity price forecasting," in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, Nov. 23-25, 2018.

[30] G. Memarzadeh and F. Keynia, "Short-term electricity load and price forecasting by a new optimal lstm-nn based prediction algorithm," *Electric Power Systems Research*, vol. 192, 2021, Art. no. 106995.

[31] S. I. Vagropoulos, E. G. Kardakos, C. K. Simoglou, A. G. Bakirtzis, and J. P. Catalao, "Ann-based scenario generation methodology for stochastic variables of electric power systems," *Electric Power Systems Research*, vol. 134, pp. 9–18, 2016.

[32] J. M. Morales, S. Pineda, A. J. Conejo, and M. Carrion, "Scenario reduction for futures market trading in electricity markets," *IEEE Transactions on Power Systems*, vol. 24, no. 2, pp. 878–888, 2009.

[33] B. Stappers, N. G. Paterakis, K. Kok, and M. Gibescu, "A class-driven approach based on long short-term memory networks for electricity price scenario generation and reduction," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 3040–3050, 2020.

[34] L. Vandezande, L. Meeus, R. Belmans, M. Saguan, and J.-M. Glachant, "Well-functioning balancing markets: A prerequisite for wind power integration," *Energy policy*, vol. 38, no. 7, pp. 3146–3154, 2010.

[35] R. A. C. van der Veen and L. J. De Vries, "Balancing market design for a decentralized electricity system: Case of the netherlands," in *2008 First International Conference on Infrastructure Systems and Services: Building Networks for a Brighter Future (INFRA)*, Rotterdam, Netherlands, Nov. 10-12, 2008.

[36] TenneT B.V., "Onbalansprijssystematiek," 2020, accessed on 2022-03-08. [Online]. Available: https://www.tennet.eu/fileadmin/user_upload/SO_NL/Onbalansprijssystematiek.pdf

[37] G. A. Laugs, R. M. Benders, and H. C. Moll, "Balancing responsibilities: Effects of growth of variable renewable energy, storage, and undue grid interaction," *Energy Policy*, vol. 139, 2020, Art. no. 111203.

[38] D. Kottick, M. Blau, and D. Edelstein, "Battery energy storage for frequency regulation in an island power system," *IEEE Transactions on Energy Conversion*, vol. 8, no. 3, pp. 455–459, 1993.

[39] F. Conte, S. Massucco, G. P. Schiapparelli, and F. Silvestro, "Day-ahead and intra-day planning of integrated BESS-PV systems providing frequency regulation," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 3, pp. 1797–1806, 2020.

[40] S. Karagiannopoulos, E. Vrettos, G. Andersson, and M. Zima, "Scheduling and real-time control of flexible loads and storage in electricity markets under uncertainty," in *11th International Conference on the European Energy Market (EEM14)*, Krakow, Poland, May 28-30, 2014.

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

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

[43] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[44] X.-H. Le, H. V. Ho, G. Lee, and S. Jung, "Application of long short-term memory (LSTM) neural network for flood forecasting," *Water*, vol. 11, no. 7, 2019, Art. no. 1387.

[45] N. Vanvuchelen, J. Gijsbrechts, and R. Boute, "Use of proximal policy optimization for the joint replenishment problem," *Computers in Industry*, vol. 119, 2020, Art. no. 103239.

[46] C. Ju, P. Wang, L. Goel, and Y. Xu, "A two-layer energy management system for microgrids with hybrid energy storage considering degradation costs," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6047–6057, 2017.

[47] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv:1606.01540*, 2016.

[48] X.-Y. Liu, Z. Li, Z. Yang, J. Zheng, Z. Wang, A. Walid, J. Guo, and M. I. Jordan, "ElegantRL-Podracer: Scalable and elastic library for cloud-native deep reinforcement learning," *arXiv:2112.05923*, 2021.

[49] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2015.

[50] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous, "Tensorflow distributions," *arXiv:1711.10604*, 2017.

[51] M. Yuan, M.-o. Pun, D. Wang, Y. Chen, and H. Li, "Multimodal reward shaping for efficient exploration in reinforcement learning," *arXiv:2107.08888*, 2021.