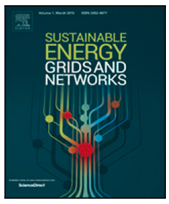
Sustainable Energy, Grids and Networks 42 (2025) 101677

Contents lists available at ScienceDirect

Sustainable Energy, Grids and Networks

journal homepage: www.elsevier.com/locate/segan



On the participation of energy storage systems in reserve markets using Decision Focused Learning☆

Ángel Paredes [a,*], Jean-François Toubeau [b], José A. Aguado [a], François Vallée [b]

[a] University of Málaga, Bulevar Louis Pasteur, 35, Málaga, 29071, Spain
[b] University of Mons, Place du Parc, 20, Mons, 7000, Belgium

ARTICLE INFO

Keywords:
Energy Storage
Decision Focused Learning
Market participation
Reserve markets

ABSTRACT

Battery Energy Storage Systems (BESSs) are particularly well-suited to deepen the decarbonisation of reserve markets, traditionally dominated by non-renewable generators. BESSs operators often rely on Predict-Then-Optimise (PTO) methods to participate in these markets, which focus on forecasting market conditions without directly considering the impact of subsequent decisions during training. Recently, learning models have evolved to incorporate decision outcomes during training, known as Decision Focused Learning (DFL) methodologies, which have the potential to increase market benefits. This paper introduces a DFL approach that integrates the decision-making process of BESSs when participating in reserve markets into the training of their predictive models. By expressing the optimisation problem as a primal–dual mapping using the Karush–Kuhn–Tucker (KKT) conditions, the proposed DFL method enables the regressor to learn from the BESS's decisions, refining its predictions based on observed outcomes, improving decision accuracy and market performance. Results show that the proposed DFL approach outperforms traditional PTO methods, with up to a 9.5% increase in profits for a case study based on the Belgian secondary reserve market, highlighting its effectiveness in managing the complexities of dynamic market conditions.

As the integration of renewable energy sources accelerates, Battery Energy Storage Systems (BESSs) have become vital for reducing reliance on fossil-based generation in reserve markets. Their ability to provide flexibility while nourishing from renewable energy sources makes them particularly well-suited for balancing the grid, addressing the intermittency of wind and solar power, and supporting the shift towards cleaner energy [1]. Traditional methods for participating in reserve markets often rely on Predict-Then-Optimise (PTO) approaches that do not fully capture the complexities of dynamic market conditions. This is especially relevant for the activation of reserve events, which are challenging to predict due to the inherent variability in system balance and which have a significant impact on the market participants' returns [2].

Several works have focused on enhancing the participation of BESSs by optimising their operation in these markets. However, many of these approaches still face computational complexity issues, particularly in the stochastic optimisation of energy storage systems [3,4]. Current trends increasingly rely on machine learning techniques to predict future market conditions and optimise market participation based on these forecasts [5]. For example, Shapley values are employed to interpret a complex model that predicts energy activation in reserve markets by [2], while [6] evaluates the performance of deep learning methods such as Long-Short Term Memory (LSTM) in forecasting reserve market prices. Similarly, [7,8] propose models that focus on market price uncertainty, optimising participation while ensuring delivery guarantees. Other methods, such as stochastic dynamic programming, are also used to optimise BESSs participation, albeit with significant computational burdens [9]. In a similar vein, works like [10] have proposed multi-level optimisation models, but these methods still depend heavily on accurate forecasts, which are not guaranteed in practice. Notably, much of the literature lacks a focus on reducing the computational complexity during optimisation phase, and face issues when incorporating decision-making-process into the training of predicting models.

Recent advances in machine learning have proposed models to deal with real-time conditions in the context of market participation,

☆ The work of Ángel Paredes was supported by Spanish Ministry of Education [FPU19/03791] and by the University of Málaga. José A. Aguado and Ángel Paredes were supported by Ministerio de Ciencia e Innovación, Spain [TED2021-132339B-C42, PID2022-142372OB-C22] and by Horizon Europe Programme [101096787, 101123556]. The authors thankfully acknowledge the computer resources provided by the SCBI center of the University of Málaga.

* Corresponding author.
E-mail addresses: angelparedes@uma.es (Á. Paredes), jean-francois.toubeau@umons.ac.be (J.-F. Toubeau), jaguado@uma.es (J.A. Aguado), francois.vallee@umons.ac.be (F. Vallée).

https://doi.org/10.1016/j.segan.2025.101677
Received 27 December 2024; Received in revised form 25 February 2025; Accepted 3 March 2025
Available online 13 March 2025

such as Model Predictive Control (MPC) and reinforcement learning. For instance, [11] presents a forecast-informed MPC methodology for BESSs in imbalance settlement mechanisms. Similarly, [12] proposes a stochastic MPC framework for real-time commitments in energy and frequency regulation markets, which mitigates long-term demand charges and improves payback periods for stationary batteries, outperforming deterministic MPC approaches. In addition, [13] employs a model-free deep reinforcement learning method to address battery degradation cost estimation for energy arbitrage. Furthermore, [14] introduces an inverse reinforcement learning framework that identifies the bidding decision objectives for BESSs across coupled multi-markets. Reinforcement learning approaches are also explored by [15], who proposes a proximal policy optimisation agent for capacity scheduling of photovoltaic-battery systems, which can be enhanced by including a control policy correction framework as in [16]. Furthermore, authors in [17] develops a temporal-aware deep reinforcement learning for BESS bidding in energy and reserve markets, utilising a transformer-based temporal feature extractor to respond only to price fluctuations.

Similarly, direct Decision Focused Learning (DFL) approaches have also emerged which map features straightly to decisions without forecasting intermediate market conditions. Ref. [18] proposes an extreme learning approach for short-term renewable energy forecasting, where the subsequent decisions are included in the training process of the regressor model. Likewise, extreme learning for wind power reserve quantification is used in [19] to optimise both prediction intervals and reserve amounts. Authors in [20] introduces a model-free end-to-end learning framework for economic dispatch, illustrating the inefficiencies of traditional PTO approaches. Nonetheless, both direct DFL and model-free methods like these are unsuitable for BESSs participation in reserve markets due to the lack of guaranteed feasibility in the solutions, as they approximate the optimal decisions without explicitly considering the constraints of the BESS operation.

Indirect DFL approaches offer greater flexibility and guarantee solution feasibility [21,22] defining the optimisation problem as a layer of the regressor. This method is used by [23] in the development of a neural network structure that takes into account energy system conditions to generate wind power forecasts. Authors in [24] present a method for prescriptive trees that learns decision strategies directly from data by minimising the expected cost through a weighted sample average approximation, though gradients of the decision strategies concerning training parameters are not computed. The computation of gradients of the solution of optimisation problems with respect to the parameters of the regressor remains a challenge. Even simple linear programming resolution methods, such as Dantzig's Simplex, are not differentiable, since gradients are only defined at the vertices of the feasible region, being undefined elsewhere.

Recent advances have proposed surrogate models to address the difficulty of computing these gradients. For instance, [25] introduces Gaussian process based surrogate models for optimisation problems, allowing gradient back-propagation in non-differentiable settings. Other recent works, such as [26], propose hybrid loss functions that account for both prediction errors and deviations from optimal decisions, further refining the decision-making process. While these methods are potentially well-adapted, they have not yet been applied to BESSs in real-world reserve market participation, where deviations from schedules during actual service provision must be considered. Additionally, most existing approaches fail to account for uncertainty in the optimisation constraints across different problem domains, as evidenced by the review of data-driven applications in [27], which underscores the challenges of end-to-end learning frameworks in effectively integrating user-imposed constraints. Addressing this limitation is especially crucial for BESSs participating in reserve markets, since the duration of reserve events directly affects these constraints, as highlighted by [28].

The conducted literature review reveals the following research gaps in knowledge:

- G1 The uncertainty of reserve market conditions is not adequately addressed during the learning process of predictive models. Specifically, the impact of reserve activation forecasts on BESS decision-making is often overlooked in the literature, largely because incorporating these factors into the constraints of the optimisation problem is challenging.
- G2 Existing end-to-end learning frameworks lacks from methodologies that embed optimal decision-making with constraint uncertainty for BESSs in reserve markets within the training process. Current approaches predominantly focus on predicting market conditions without addressing the critical impact of deviations from optimal decisions, which can result in significant financial losses.

This paper firstly presents a methodology that includes the decision-making process of BESSs participating in reserve markets in the training of the regressor model. The key characteristics of the proposed methodology are:

- C1 To the authors' knowledge, this is the first paper to propose a DFL approach for the participation of BESSs in reserve markets. The methodology employs a hybrid loss function that considers both the prediction error and the profit loss compared to the optimal decision provided by an oracle.
- C2 The methodology extends surrogate models by allowing the BESS to learn from its decisions while correcting its initial day-ahead strategy in real time. This is of paramount importance because of the impact on profits, and cannot be achieved with existing methods.
- C3 Introduction of the uncertainty in the set of constraints of the optimisation problem, which cannot be done with existing DFL methodologies based on surrogate models. Karush–Kuhn–Tucker (KKT) conditions are used to express the problem as a primal–dual mapping which can be differentiated and included as a layer of the regressor.

The reminder of the paper is structured as follows. Section 1 presents the traditional PTO approach for BESS participation in reserve markets. Section 2 describes the proposed DFL method. Section 3 presents a case study to demonstrate the effectiveness of the proposed method. Finally, Section 4 concludes the paper.

1. Traditional predict-then-optimise

The framework for the participation of BESSs in reserve markets using PTO methodology is shown in Fig. 1. The methodology firstly trains a regressor model f_{NN} to predict the market prices and the duration of the activation of the reserve events. This is done by training a regressor model $\hat{y} = f_{NN}(\theta, x)$ using gradient descent to fit its output $\hat{y}$ to actual information y by modifying tuneable parameters θ based on a set of features x. Then, the trained neural network is used to predict the uncertain parameters $\hat{y}$ during the day-ahead scheduling phase based on the last available information x. An optimisation problem $\phi(\cdot)$ is solved to compute the bids u^* for the next 24 h. Lastly, this schedule is corrected in real-time $u^{*'}$ after the uncertainty is realised $\hat{y}^*$, and the BESS position in the market is updated, obtaining true profits p.

1.1. Regressor model

In the context of BESS participation in reserve markets, regressor models predict automatic Frequency Restoration Reserve (aFRR) capacity prices $\hat{\lambda}_t^{r,u}$, $\hat{\lambda}_t^{r,d}$, energy prices $\hat{\lambda}_t^{e,u}$, $\hat{\lambda}_t^{e,d}$, and event duration $\hat{d}_t^u$, $\hat{d}_t^d$ for the next 24 h. Let x_t represents the set of features available at time t (e.g., historical prices, market conditions, weather data). Without loss of generality, a simple feed-forward Neural Network (NN) regressor model is defined as $\hat{y} = f_{NN}(x_t, \theta)$, where θ represents the tuneable parameters of the model. Note that both the PTO and DFL

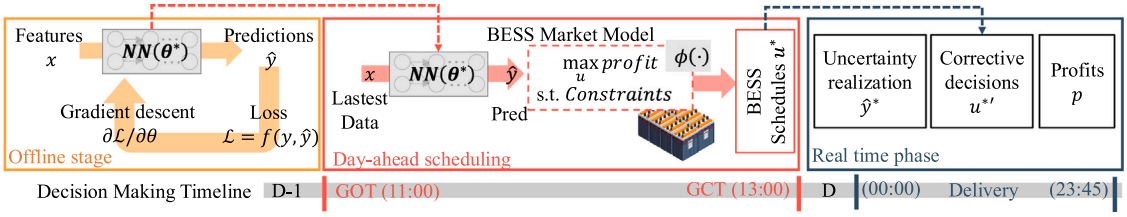


Fig. 1. Framework for the participation of BESSs in reserve markets using PTO methodology, where the regressor model is solely trained based on predictions $\hat{y}$ without considering the impact of decisions u^*.

methodologies are model-agnostic, allowing the both frameworks to be applied to any regressor structure.

The neural network $f_{NN}(x_t, \theta)$ consists of L layers, each with weights θ_l^w and biases θ_l^b. The output of a network layer y_{l+1} is computed as a function of the previous y_l:

$$y_{l+1} = \sigma(\theta_l^w y_l + \theta_l^b), \quad l = 1, 2, \ldots, L \tag{1a}$$

where σ is the activation function and $y_0 = x$, $y_L = \hat{y}$. The model is trained by minimising a loss function $\mathcal{L}(\hat{y}, y)$ that measures the difference between predicted values $\hat{y}$ and actual historical data y. The learning problem is formulated as:

$$\theta^* = \arg\min_{\theta}\{\mathcal{L}(\hat{y}, y) \text{ s.t. } \hat{y} = f_{NN}(x_t, \theta)\} \tag{1b}$$

The training process uses gradient descent, where the parameters θ are updated iteratively based on the gradients of the loss function:

$$\theta_i^{(k+1)} = \theta_i^{(k)} + \alpha \frac{\partial \mathcal{L}(\hat{y}, y)}{\partial \theta_i}, \quad \forall i, k \tag{1c}$$

where α is the learning rate. The process continues until the parameters converge to a (local) minimum of the loss function. Once trained, the regressor generates predictions $\hat{\lambda}_t^{r,u}$, $\hat{\lambda}_t^{r,d}$, $\hat{\lambda}_t^{e,u}$, $\hat{\lambda}_t^{e,d}$, $\hat{d}_t^u$, and $\hat{d}_t^d$, which are then used as inputs for the subsequent BESS optimisation problem.

1.2. aFRR market participation

Reserve markets in Europe maintain grid stability by procuring flexibility services to balance supply and demand in real time. These markets are organised into primary, secondary, tertiary, and restoration services, with secondary markets being particularly attractive for BESSs as their response capabilities are particularly aligned with aFRR products traded. aFRR products operate in two stages: capacity contracting and energy activation. In the day-ahead auction, the Transmission System Operator (TSO) contracts power capacity for the next day on an hourly basis, with Gate Opening Time (GOT) and Gate Closing Time (GCT) typically set at 11:00 and 13:00, respectively. Then, activation occurs in real-time based on system frequency deviations, being typically partial and characterised by event duration in both upward d_t^u and downward d_t^d directions. This activation is linked to previously contracted capacities after the clearing of the day-ahead capacity market [29]. The optimisation problem (2) is formulated to generate bids for the next 24 h on a day-ahead basis and is updated in real-time to reflect the BESS market position.

$$\max \sum_t \left[\hat{\lambda}_t^{r,u} p_t^u + \hat{\lambda}_t^{r,d} p_t^d\right] + \sum_t \left[\hat{\lambda}_t^{e,u} p_t^u \hat{d}_t^u - \hat{\lambda}_t^{e,d} p_t^d \hat{d}_t^d\right] - \\ - C^{DEG} \sum_t \left[\Delta b_{loss,t}^{cal} + \Delta b_{loss,t}^{cyc}\right] \tag{2a}$$

Subject to,

$$soc_t = soc_{t-1} + \eta^{CH} p_t^d \hat{d}_t^d - p_t^u \hat{d}_t^u / \eta^{DIS} \quad \forall t \tag{2b}$$

$$\underline{SOC} \le soc_t \le \overline{SOC} \quad \forall t \tag{2c}$$

$$0 \le p_t^u \le P^{CONV} \quad \forall t \tag{2d}$$

$$0 \le p_t^d \le P^{CONV} \quad \forall t \tag{2e}$$

$$\Delta b_{loss,t}^{cal} \ge a_m soc_t + b_m \quad \forall m, \forall t \tag{2f}$$

$$\Delta b_{loss,t}^{cyc} \ge c_m (p_t^d + p_t^u)/P^{CONV} + d_m \quad \forall m, \forall t \tag{2g}$$

The objective function (2a) maximises expected revenue from predicted aFRR capacity prices $\hat{\lambda}_t^{r,u}$, $\hat{\lambda}_t^{r,d}$, energy prices $\hat{\lambda}_t^{e,u}$, $\hat{\lambda}_t^{e,d}$, power bids p_t^u, p_t^d, and event duration $\hat{d}_t^u$, $\hat{d}_t^d$ in upward u and downward d directions [30]. The revenue is offset by the degradation cost C^{DEG}, which accounts for calendar $\Delta b_{loss,t}^{cal}$ and cyclic $\Delta b_{loss,t}^{cyc}$ degradation. Traditional methods rely on binary variables or complementary constraints to prevent simultaneous charging and discharging. However, as demonstrated by [31], this can be effectively enforced by incorporating a penalty term of the form $P \cdot \sum_t (p_t^u + p_t^d)$ in the objective function, provided that $P \ge 0$ [32]. In Problem (2), the cyclic degradation cost term $C^{DEG} \sum_t \Delta b_{loss,t}^{cyc}$ in (2a) serves the same purpose while preserving convexity. The constraints define State of Charge (SOC) dynamics in (2b), ensure SOC limits (2c), and impose converter power bounds (2d) and (2e). The SOC at time t, soc_t, is bounded by $\underline{SOC}$ and $\overline{SOC}$, with charging and discharging efficiencies η^{CH} and η^{DIS}. The converter power limit is P^{CONV}. Degradation is modelled by (2f) and (2g), using coefficients a_m, b_m, c_m, and d_m for calendar and cyclic degradation mechanisms, given a set of intervals m [33].

1.3. Real-time correction

In practice, actual market conditions may differ from scheduled values after solving (2), leading to deviations in the BESS market position. These deviations must be corrected by trading energy in the imbalance market. The process is outlined in Algorithm 1.

Algorithm 1 BESS Reserve Market Participation

Input: Trained regressor $f_{NN}(\theta^*, \cdot)$, features x
Predict: $\hat{\lambda}_t^{r,u}, \hat{\lambda}_t^{r,d}, \hat{\lambda}_t^{e,u}, \hat{\lambda}_t^{e,d}, \hat{d}_t^u, \hat{d}_t^d \leftarrow f_{NN}(\theta^*, x)$
Compute day-ahead schedules: $\hat{p}_t^u, \hat{p}_t^d \leftarrow \arg(2)$
for $t = 1$ to T **do**
  Uncertainty realisation: $\lambda_t^{r,u}, \lambda_t^{r,d}, \lambda_t^{e,u}, \lambda_t^{e,d}, d_t^u, d_t^d$
  Update realised SOC soc_t', compute imbalance power:
    Downward: $p_t^{\text{im},d} \leftarrow \max(0, (soc_t' - \overline{SOC})/\Delta t)$
    Upward: $p_t^{\text{im},u} \leftarrow \max(0, (\underline{SOC} - soc_t')/\Delta t)$
    $soc_t' \leftarrow soc_t + \Delta t[-\eta^{CH} p_t^{\text{im},d} + p_t^{\text{im},u}/\eta^{DIS}]$
end for
Compute: True profits using (3)

The algorithm starts by predicting aFRR capacity prices, energy prices, and event duration using the neural network regressor $f_{NN}(\theta^*)$ based on input features x. The optimisation problem (2) is then solved to compute day-ahead schedules $\hat{p}_t^u$ and $\hat{p}_t^d$. In real time for every period t, once uncertainty is realised, if the SOC exceeds the predefined limits, the BESS operator trades $p_t^{\text{im},u}$ and $p_t^{\text{im},d}$ in the imbalance market and the SOC is updated accordingly. Finally, the true profits are calculated using (3).

$$\text{True Profits} = \sum_t \left[\lambda_t^{r,u}(p_t^u - p_t^{im,u}) + \lambda_t^{r,d}(p_t^d - p_t^{im,d})\right] + \\ + \sum_t \left[\lambda_t^{e,u}(p_t^u d_t^u - p_t^{im,u}\Delta t) - \lambda_t^{e,d}(p_t^d d_t^d - p_t^{im,d}\Delta t)\right] -$$

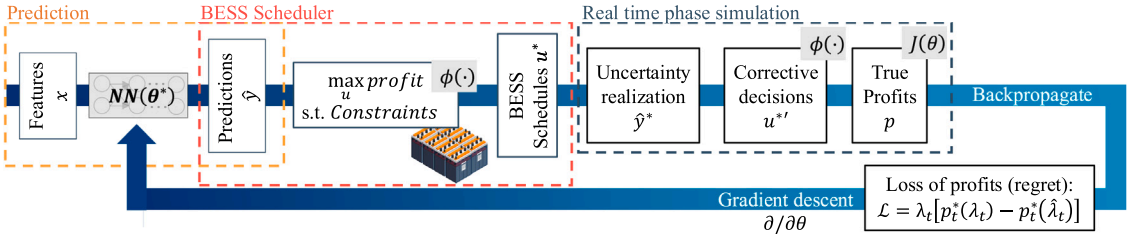


Fig. 2. DFL-based regressor training process for BESSs participation in reserve markets.

$$- \sum_t \lambda_t^{im}(p_t^{im,u} + p_t^{im,d})\Delta t \tag{3}$$

The PTO methodology does not capture real-time corrections required to adjust the BESS position after uncertainty is realised. This limitation prevents the regressor from learning and improving its strategy based on actual outcomes.

2. Methodology: Decision focused learning

The proposed methodology is based on DFL, enabling the BESS to learn from its decisions and adjust the training process of the regressor model. This improves performance in maximising true profits, rather than just predicting market uncertainties. The DFL problem is defined as finding the optimal parameters θ^* of the regressor $f_{NN}(\theta, x)$ that minimise the regret $\mathcal{L}(z^*(\hat{y}), z^*(y))$, where $z^*(\hat{y})$ represents optimal decisions obtained with predicted values $\hat{y}$, and $z^*(y)$ represents optimal decisions obtained with actual values y. Formally, the DFL learning problem is:

$$\theta^* = \arg\min_\theta \Big\{ \mathcal{L}\left(z^*(\hat{y}), z^*(y)\right), \tag{4a}$$

$$\text{s.t.} \quad \hat{y} = f_{NN}(\theta, x) \tag{4b}$$

$$z^*(\hat{y}) = \arg\min_z \Big\{ c(z, \hat{y}), \tag{4c}$$

$$\text{s.t.} \quad g(z, \hat{y}) \le 0 \Big\} \Big\} \tag{4d}$$

Fig. 2 illustrates the training loop. Unlike PTO, the DFL approach integrates both the scheduling and real-time phases into training. During the forward pass, the model predicts market prices and event duration, optimises bids for the next 24 h by (2), and adjusts these bids in real time using Algorithm 1. After that, true profits are computed, and gradients are back-propagated through the optimisation problem, allowing the BESS to learn and refine its strategy. Nevertheless this back-propagation remains as a challenge, as the uncertainties in the constraints make difficult to effectively define a surrogate model that always provides feasible solutions during training.

2.1. Forward pass

The forward pass involves predicting market prices and event duration using the regressor $f_{NN}(\theta, x)$ as defined in (1a), but other structures such as LSTM, or Recurrent Neural Network (RNN) can also be used. The regressor's output $\hat{\lambda}_t$ is used to compute day-ahead bids by solving (2). These bids are adjusted in real time $u^{*\prime}$ following Algorithm 1. This process yields true profits p, which are calculated using (3) at the end of the forward pass. The true profits are then used to compute the loss function $\mathcal{L}$, or regret, which measures the difference between DFL-based profits $\mathcal{R}(\hat{\lambda}_t)$ and the maximum possible profits from an oracle with perfect information $\mathcal{R}(\lambda_t)$.

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}\left(\mathcal{R}(\lambda_t) - \mathcal{R}(\hat{\lambda}_t)\right)^\beta + \frac{\gamma}{T}\sum_{t=1}^{T}\left(\lambda_t - \hat{\lambda}_t\right)^2 \tag{5}$$

The regret $\mathcal{L}$ is a non-negative function, as no market decision can outperform one made with perfect information. The hyper-parameter γ controls the influence of Mean-Squared Error (MSE) in the prediction, while β determines whether the profits lost are penalised linearly ($\beta = 1$) or quadratically ($\beta = 2$).

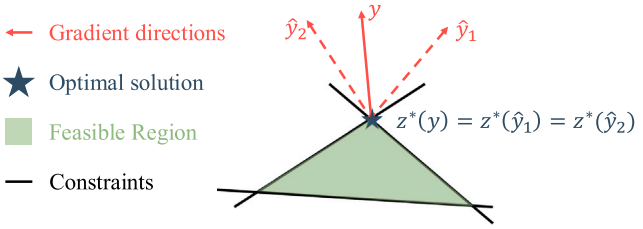


Fig. 3. Illustration of gradients of the optimal solution with respect to predicted values. The solution $z^*(y)$ is the same for both $\hat{y}_1$ and $\hat{y}_2$, but the gradients differ.

2.2. Back-propagation

The gradient $\nabla_\theta L\left(\hat{y}^i, y^i\right)$ is computed using the chain rule:

$$\frac{\partial \mathcal{L}(z^*(\hat{y}), z^*(y))}{\partial \theta} = \underbrace{\frac{\partial \mathcal{L}(z^*(\hat{y}), z^*(y))}{\partial z^*(\hat{y})}}_{①} \cdot \underbrace{\frac{\partial z^*(\hat{y})}{\partial \hat{y}}}_{②} \cdot \underbrace{\frac{\partial \hat{y}}{\partial \theta}}_{③} \tag{6}$$

Terms ① and ③ are straightforward to compute, as they involve differentiable functions. The regret gradient, $\partial\mathcal{L}(z^*(\hat{y}), z^*(y))/\partial z^*(\hat{y})$, is a simple difference, and the gradient of the regressor model $\partial\hat{y}/\partial\theta$ is differentiable for the wide-spread activation functions σ used in machine learning. However, term ② is not convex and non-differentiable. In a simple linear programming problem, the solver finds optimal solutions at the vertices of the polytope defined by the constraints, typically following the simplex method. Thus, $\partial z^*(\hat{y})/\partial\hat{y}$ is undefined except at the vertices of the feasible region, as illustrated by Fig. 3, where the optimal solution $z^*(y)$ is the same for different predicted values $\hat{y}_1$ and $\hat{y}_2$, but the gradients differ.

To overcome this issue, we treat the optimisation (2) as an implicit function using the primal–dual relationship via the KKT conditions [34], necessary and sufficient for optimality in convex optimisation problems. Let $z^*(\lambda)$ be the optimal solution of:

$$z^*(\lambda) = \arg\min_z \{ f(z, \lambda) \text{ s.t. } g(z, \lambda) \le 0, \quad h(z, \lambda) = 0 \} \tag{7}$$

Here, $f(z, \lambda)$ is the objective, $g(z, \lambda)$ are inequality constraints, and $h(z, \lambda)$ are equality constraints. Let ν and μ be the dual variables for these constraints. The optimal primal–dual solution $(z^*, \nu^*, \mu^*)(\lambda)$ can be written as a function of the predicted values λ. The key aspect to enable the differentiation through these problems is to view the solver as a procedure of fixed-point iterations that find the root of the KKT system:

$$G(z, \nu, \mu) = \begin{bmatrix} \nabla_z f(z, \lambda) + \partial_z g(z, \lambda)^T \nu + \partial_z h(z, \lambda)^T \mu \\ \nu \circ g(z, \lambda) \\ h(z, \lambda) \end{bmatrix} \tag{8a}$$

where $\partial_z g(z,\lambda)$ is the Jacobian of g with respect to z. Thus, we can treat the convex solver as a root-finding method that attempts to find (z^*, ν^*, μ^*) such that:

$$G(z^*, \nu^*, \mu^*) = 0, \quad g(z^*, \lambda) \leq 0, \quad \lambda^* \geq 0 \tag{8b}$$

To differentiate through convex problems, the focus can be on the equality conditions of the KKT system, $G(z^*, \nu^*, \mu^*) = 0$. This simplification is possible because the complementarity conditions, $\nu^* \circ g(z^*, \lambda) = 0$, imply that either $\nu_i^* = 0$ or $g_i(z^*, \lambda) = 0$ for each i. Only one of these conditions is active, allowing us to focus on the equality constraints. Thus, the optimisation is viewed as a root-finding problem for the system $G(z^*, \nu^*, \mu^*) = 0$. To compute term ②, we implicitly differentiate this system with respect to $(z^*, \nu^*, \mu^*)(\lambda)$:

$$\partial_{z,\nu,\mu} G(z^*(\lambda), \nu^*(\lambda), \mu^*(\lambda), \lambda) = 0 \tag{9a}$$

Applying the chain rule, yields:

$$\partial_{z,\nu,\mu} G(z^*, \nu^*, \mu^*, \lambda) \cdot \partial_\lambda (z^*, \nu^*, \mu^*)(\lambda) + \partial_\lambda G(z^*, \nu^*, \mu^*) = 0 \tag{9b}$$

Thus, the gradient of the optimal primal–dual solution $(z^*, \nu^*, \mu^*)(\lambda)$ with respect to λ becomes:

$$\partial_\lambda (z^*, \nu^*, \mu^*)(\lambda) = -\left(\partial_{z,\nu,\mu} G(z^*, \nu^*, \mu^*, \lambda)\right)^{-1} \cdot \partial_\lambda G(z^*, \nu^*, \mu^*) \tag{9c}$$

Substituting this into the back-propagation Eq. (6)

$$\frac{\partial \mathcal{L}(z^*(\hat{y}), z^*(y))}{\partial \theta} = -\frac{\partial \mathcal{L}}{\partial z^*}\left[\frac{\partial G(z^*, \nu^*, \mu^*, \lambda)}{\partial \lambda}\right]^{-1} \frac{\partial G(z^*, \nu^*, \mu^*, \lambda)}{\partial \lambda} \frac{\partial \lambda}{\partial \theta} \tag{10a}$$

Most automatic differentiators compute the gradient of the loss function as the transpose of the Jacobian $\partial \mathcal{L}/\partial \theta$:

$$\nabla_\theta \mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial \theta}\right)^T = \nabla_\theta \lambda \cdot \left(\frac{\partial G(z^*, \mu^*, \nu^*, \lambda)}{\partial \lambda}\right)^T \cdot \cdot \left(\frac{\partial G(z^*, \mu^*, \nu^*, \lambda)}{\partial (z, \nu, \mu)}\right)^{-T} \cdot \nabla_{z^*} \mathcal{L} \tag{10b}$$

To avoid computing the inverse transpose of the Jacobian $G(z^*, \mu^*, \nu^*, \lambda)$, define vector $\mathbf{v}$ by solving the linear system:

$$\left(\frac{\partial G(z^*, \mu^*, \nu^*, \lambda)}{\partial (z, \nu, \mu)}\right)^T \cdot \mathbf{v} = \nabla_{z^*} \mathcal{L} \tag{10c}$$

This system can be efficiently solved using wide-spread direct or iterative methods, such as LU decomposition or Newton–Raphson [35]. Under standard regularity conditions, $G(z, \nu, \mu)$ is differentiable at the optimal solution $(z^*, \mu^*, \nu^*, \lambda)$ since $f(z, \lambda), g(z, \lambda)$, and $h(z, \lambda)$ are continuously differentiable for problem at hand. The Jacobians $\partial G/\partial \lambda$ and $\partial G/\partial (z, \nu, \mu)$ exist and enable implicit differentiation as in (10b) and (10c). The final expression for the gradient of the loss function with respect to θ is:

$$\nabla_\theta \mathcal{L} = \nabla_\theta \cdot \lambda \left(\frac{\partial G(z^*, \mu^*, \nu^*, \lambda)}{\partial \lambda}\right)^T \cdot \mathbf{v} \tag{10d}$$

Thus, back-propagating through the optimisation problem is feasible and computationally efficient, requiring solving the system (10c), evaluating the KKT gradients at the optimal solution (z^*, ν^*, μ^*), and computing the gradients of the regressor model $\nabla_\theta \lambda$.

2.3. Training algorithm

The DFL methodology trains the regressor parameters θ using the proximal stochastic gradient method [36]. The goal is to iteratively update θ to minimise the regret function $\mathcal{L}$, defined by (5). The algorithm is outlined below:

In this algorithm, the regressor's parameters θ are initialised, and the training loop runs for $k = 1, \ldots, K$ iterations. In each iteration, a mini-batch $\mathcal{B}^k$ is sampled from the training set $\mathcal{T}$ of size N, with input features x^N and output predictions y^N. The forward pass involves: (i) predicting market conditions $\hat{y}^k = [\hat{\lambda}_t^{r,u}, \hat{\lambda}_t^{r,d}, \hat{\lambda}_t^{e,u}, \hat{\lambda}_t^{e,d}, \hat{d}_t^u, \hat{d}_t^d]$, (ii) computing the optimal scheduling $z^*(\hat{y}^k) = [p_t^u, p_t^d]$ based on these predictions, (iii) realising uncertainty $\hat{y}^*$, (iv) applying corrective actions $u^{*\prime k} = [p_t^{im,u}, p_t^{im,d}]$, and (v) evaluating true profits p^k. Finally, the backward pass is performed, evaluating the loss function $\mathcal{L}$ as in (10d). The projection Π_Θ ensures that the updated parameters remain within a feasible set Θ. Importantly, while day-ahead aFRR market commitments (p_t^u, p_t^d) remain binding, the BESS still may need to adjust real-time dispatch due to forecast errors in event duration, requiring corrective actions via the imbalance market. This structure is the same than the PTO operations; however, DFL enhances learning by incorporating the gradients of the real-time corrections. The package CVXPYLayers [37] is used to implement this training algorithm, integrating convex optimisation problems as differentiable layers within the regressor, and enabling back-propagation through these optimisation problems during training. This allows the regressor to directly incorporate the optimisation logic in the training loop, aligning the learning with market objectives.

Algorithm 2 Proximal Stochastic Gradient Method for DFL

Input: Training data $\mathcal{T} = \{(x^1, y^1), \ldots, (x^N, y^N)\}$, learning rate α
Initialise: Parameters of the regressor $\theta^1 \leftarrow \theta$
for $k = 1$ to K **do**
  Sample a mini-batch $\mathcal{B}^k \subset \mathcal{T}$
  Forward pass:
    Prediction: $\hat{y}^k = f_{NN}(\theta^k, x)$
    Solve optimisation problem: $z^*(\hat{y}^k) \leftarrow$(2)
    Realise uncertainty $\hat{y}^*$
    Compute corrective actions: $u^{*\prime k} \leftarrow$ **Algorithm 1**
    Compute true profits: $p^k(z^*(\hat{y}^k)) \leftarrow$(3)
  Backward pass:
    Solve system of equations (10c) to compute $\mathbf{v}^k$.
    Compute mini-batch gradient: $\nabla_\theta \mathcal{L} \leftarrow$ (10d)
  Compute epoch gradient: $g^k = \frac{1}{|\mathcal{B}^k|} \sum_{i \in \mathcal{B}^k} \nabla_\theta \mathcal{L}$
  Update parameters: $\theta^{k+1} \leftarrow \Pi_\Theta(\theta^k + \alpha^k g^k)$
end for
Output: Optimal parameters θ^*

3. Case study

The proposed methodology is evaluated using real data from the Belgian aFRR market. Market prices for training can be found in [38], while minute-resolution activation data is available in [39]. The BESS is modelled as a 4 MW/4 MWh system, with the degradation model based on [33]. Charging and discharging efficiencies are set to 0.95 and 0.92, respectively, over a 24-hour horizon with 15-minute resolution.

The methodology is implemented in Python 3.9.18, utilising PyTorch [40] for training the regressor, and CVXPYLayers [37] to integrate optimisation within the training process. Hyper-parameter tuning is performed on a cluster with 4 TB RAM, 32 nodes (2 x AMD EPYC 7742 CPUs at 2.25 GHz) and Nvidia A100 GPUs, running Suse Leap 42 Linux [41]. Regular model training takes approximately 1 h for 200 epochs on an Apple M1 with 16 GB RAM.

3.1. Datasets and models

The analysis involved preprocessing several datasets, including weather data (temperature, humidity, pressure, cloud cover, wind speed, sun duration, and direct radiation), electricity generation mix (coal, gas, nuclear, hydro, wind, residual shares, and total generation), and aFRR market data (upward/downward capacity prices, energy prices, event duration, current and cumulative imbalances). Data from the year 2023 was used for the analysis, with 20% of the dataset randomly selected for testing. This corresponds to 292 days for training and 73 days for testing. Missing data were linearly interpolated, and

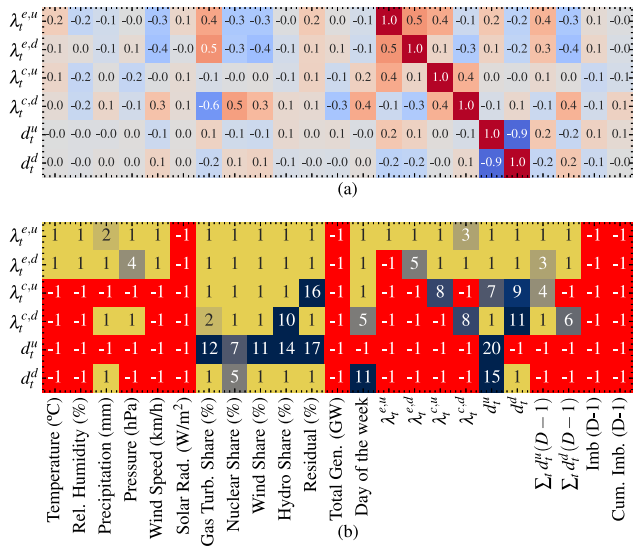


Fig. 4. (a) Spearman correlation matrix of features used in training. Prediction variables are in rows, features in columns. Only features with a correlation above |40%| are used. A correlation of 1 indicates perfect positive, −1 perfect negative, and values near 0 indicate no correlation. (b) Recursive Feature Elimination with Cross-Validation results for selecting key features (horizontal) to predict target variables (vertical). Features are ranked by importance, with −1 meaning discarded; higher values indicate less relevance.

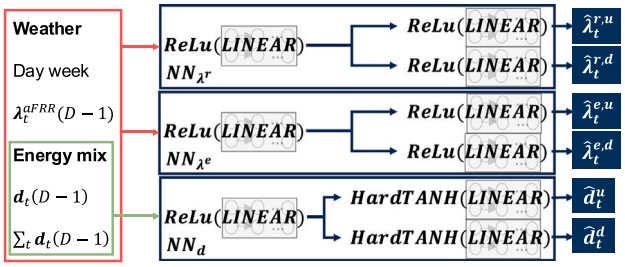


Fig. 5. Regressor architecture. It includes two linear layers with ReLU and HardTANH activation functions to constrain outputs and prevent infeasibility during training. Linear layers are replaced by LSTM and RNN layers in these models.

all datasets were resampled to 15-minute intervals. We conducted a correlation analysis using the Spearman coefficient, which is effective for non-linear relationships, ordinal data, or outliers. Features with a correlation above 40% were retained, as shown in Fig. 4(a). Features with lower correlation were excluded.

The event duration is the hardest parameter to predict due to low correlations with other features, a known challenge in the field [2]. A principal component analysis was conducted to further validate feature selection, showing that 18 features explained 98% of the variance, while 13 explained 90%. Recursive feature elimination with cross-validation was used to select key features for predicting the target variables. This method iteratively removes the least relevant features and optimises model performance through cross-validation. Fig. 4(b) ranks the features according to their importance. We also analysed auto-correlation, incorporating past-day (D-1) values to enhance the model's predictive power.

Two key insights emerge: first, the energy mix helps predict event duration, especially when combined with past-day data. Second, aFRR price forecasts benefit from most features, except solar radiation, total generation, and imbalance. Thus, based on the previous, the regressor model depicted in Fig. 5 is trained using the selected features.

Table 1
Hyper-parameter Search Space.

Hyper-parameter	Search space
Number of Layers	[1,2,3]
Optimisers	Adadelta, Adagrad, Adam, AdamW, Adamax, ASGD, NAdam, RAdam, RMSprop, Rprop, SGD
Layer Size	[20, 2208)
Learning Rate	$[10^{-5},10^{-3})$

Table 2
Best hyper-parameters.

Hyper-parameter	NN	LSTM	RNN
Number of Layers	1	2	3
Optimiser	SGD	RMSprop	Adadelta
Layer Size	2164	1333	983
Learning Rate (x10^{-4})	7.02596	0.2358	97.55

3.2. Workflow

The performance of the proposed DFL methodology is evaluated with respect to PTO approach, and two end-to-end methodologies, i.e. direct DFL and Extreme Learning Machine Regressor (ELMR). We first trained three regressors (NN, RNN, and LSTM) using the PTO method, focusing solely on minimising the MSE and optimising their hyper-parameters. These best configurations were then used to train the same models with the proposed DFL and the direct DFL approaches. The main difference between the proposed and the direct DFL approach is that the latter directly predict decision variables p_t^u and p_t^d without considering the physics of the problem at hand. Lastly, an ELMR is also trained and hyperoptimized to also predict these two decisions variables directly. We evaluated these methodologies by comparing the profits and the accuracy of the predictions obtained on the unseen testing set for the first two methodologies. Besides, for the proposed DFL, we define a grid of hyper-parameters, i.e. the regularisation term γ, the loss function β, and the number of pretraining PTO epochs before starting DFL training, to investigate their impact on the market performance.

3.3. Hyper-parameter optimisation

The hyper-parameter search space for each regressor is defined in Table 1. For optimisation, we used the Tree of Parzen Estimators method [42], which selects the next evaluation point by maximising the ratio $l(x)/g(x)$, where $l(x)$ represents observed points and $g(x)$ unobserved ones. Hyper-parameter optimisation was performed on an HPC cluster [41] using SPARK [43] for distributed computation. Each model's performance was evaluated on the training set, and the best hyper-parameters were selected. The search spaces for NN, RNN, and LSTM are shown in Table 1, with a batch size of 134 and 5000 training epochs. The best hyper-parameters found for these models are shown in Table 2. These configurations were then used for training with the direct and proposed DFL methodologies. Lastly, the best performing number of hidden neurons for the ELMR is found to be 1891 using the same search space. The training results are shown in Fig. 6, where the NN achieved lower training MSE values than the RNN and LSTM networks.

3.4. Proposed DFL training

To evaluate and compare the performance of the DFL training loop, we trained LSTM, NN, and RNN models using the same hyper-parameters from the PTO process showed in Table 2. Specifically, we analysed the impact of (i) regularisation term γ, (ii) loss function type β, and (iii) the number of pretraining epochs on the performance of the DFL-trained regressors. In total, 24 models were trained for each

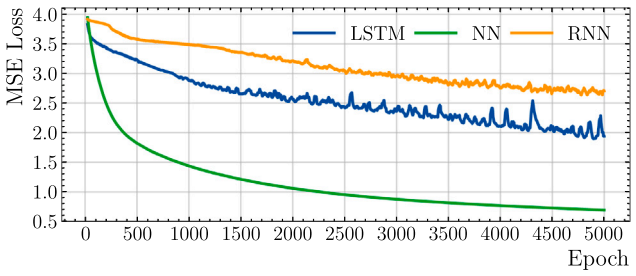


Fig. 6. Training results for LSTM, NN, and RNN regressors with best-performing hyper-parameters using PTO methodology.

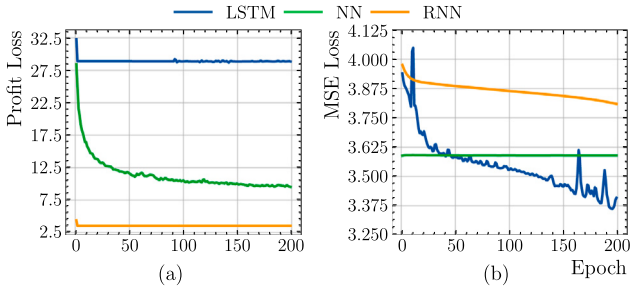


Fig. 7. Profit loss (a) and MSE loss (b) evolution for the best performing LSTM, NN, and RNN networks based on test set's true profits using the proposed DFL methodology. Best-performing LSTM: Quadratic loss, $N_{pre} = 0$, $\gamma = 10$. Best-performing NN: Quadratic loss, $N_{pre} = 50$, $\gamma = 0$. Best-performing RNN: Linear loss, $N_{pre} = 0$, $\gamma = 1$.

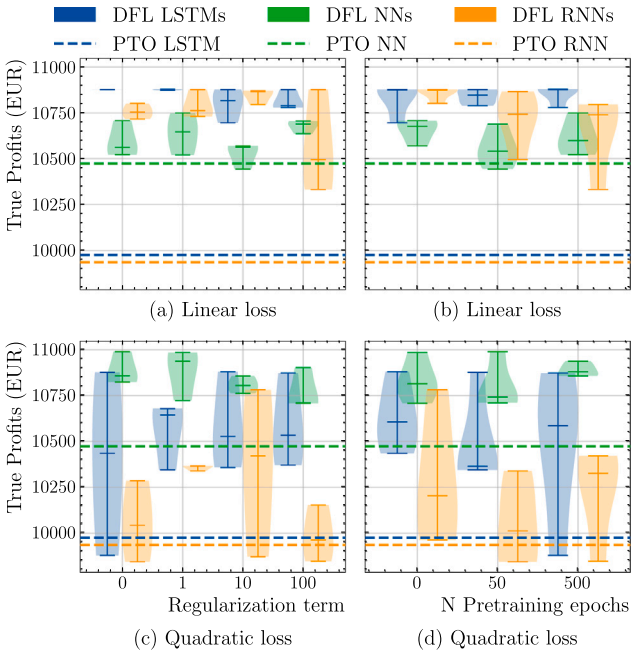


Fig. 8. Profits in the reserve market using PTO (dashed lines) and DFL (violin plots). Profits are shown by regularisation term for linear (a) and quadratic (c) loss functions, and by pretraining epochs for linear (b) and quadratic (d) loss functions. Oracle mean true profits: 14,972.11 €.

regressor type, with γ values of 0, 1, 10, and 100, β values of 1 (linear) and 2 (quadratic), and pretraining epochs N_{pre} set to 0, 50, or 500. Each model was trained for 200 epochs, with a batch size of 134, using the PTO-optimised hyper-parameters to ensure a fair comparison.

Fig. 7 presents the training results for the best-performing models of each regressor type, selected based on mean true profits from the training set. The results show that models learn in terms of mean true profits, while changes in the MSE loss are less pronounced. This is particularly evident for the NN model, where MSE loss does not significantly decrease during training.

3.5. Prediction results benchmarking

Table 3 shows that, in terms of accuracy, PTO models generally achieve lower MSE and Mean Absolute Error (MAE) values, particularly for NN_{λ^r} and NN_{λ^e}. DFL configurations with $\gamma = 0$, 1, and 10, the MSE increases with respect to PTO instances. For instance, the DFL LSTM ($\gamma = 0$) model shows a 37.19% higher MSE for NN_{λ^r} compared to its PTO counterpart. Nevertheless, the PTO NN model has an MSE of 80.503 for NN_{λ^r}, while the best DFL NN model ($\gamma = 100$) achieves a comparable MSE of 78.884. Despite the rise in MSE for some DFL models, the MAE often remains comparable, particularly in DFL NN models. For example, the DFL NN ($\gamma = 100$) achieves a MAE of 6.143 for NN_{λ^r}, close to the PTO NN model's 6.242. This suggests that increasing γ enhances prediction accuracy at the expense of reducing mean true profits, as seen in the last column. Similarly, models with $N_{pre} = 500$ tend to have lower MSE and MAE values, but with slightly reduced true profits.

The R^2 values reveal a trade-off between the two methodologies. PTO models typically achieve higher R^2 values for NN_{λ^e}, with the PTO NN reaching 0.701. Conversely, DFL models, especially for NN_{λ^e}, often show lower R^2 values, frequently below 0.6. However, DFL models still perform competitively for NN_{λ^r}, as shown by the DFL NN ($\gamma = 10$), which achieves an R^2 of 0.604, close to the best PTO model. This suggests that DFL models preserve prediction accuracy while improving mean true profits by learning from the BESS decision-making process and gaining insights into the system's underlying dynamics.

3.6. Market results benchmarking

3.6.1. Profits and bidding analysis

Table 3 shows that the DFL methodology consistently yields higher profits than the PTO method. For instance, the DFL NN with $N_{pre} = 500$ achieves mean true profits of 10,937.41 €, outperforming the best PTO NN, which reaches 10,473.60 € both on the unseen training set. This trend holds for most DFL configurations when compared with their respective PTO counterpart, as shown in Fig. 8. DFL outperforms PTO in profits across various loss types, regularisation terms, and pretraining epochs. Some exceptions occur in quadratic loss cases due to gradient instability during training. Mean True Profits increase by up to 4.06% for NN, 9.47% for RNN, and 9.08% for LSTM regressors with DFL compared to PTO. Not only are the profits higher for linear loss of profits, but also linear loss functions in DFL are more stable and profits are more consistent with lower deviations across configurations, particularly for LSTM, where the spread is reduced from 800 € to 500 €. Moreover, results in Table 4 show that other end-to-end learning approaches, lead to substantial financial losses, with direct DFL using NN reaching losses of 618,150 € in mean true profits. This can be attributed to the lack of physical awareness in the decision-making process. Although direct DFL models using LSTM and RNN exhibit higher correlation with oracle decisions (R^2 values of 0.798 and 0.800 for p_t^{μ}, respectively), their high imbalance costs (over 17,000 € in both cases) results in losses. This suggests that, while the regressor's outputs align with oracle decisions in relative terms, their incorrect scaling leads to high imbalance costs, ultimately resulting in suboptimal bidding or even leading to market exclusion.

3.6.2. Imbalance analysis

Fig. 9 shows mixed results for mean power purchased by BESS in the imbalance market. For linear losses ($\beta = 1$), DFL-trained LSTM and RNN regressors purchase slightly more power than PTO models, due to increased reserve market activity. In contrast, NN models consistently purchase less power than PTO counterparts, which contributes to higher mean true profits. Linear loss functions in DFL also show

Table 3
Prediction performance evaluation of PTO and proposed DFL regressors. For DFL models, each row corresponds to the best-performing model in terms of profit loss, with the regularisation term γ and pretraining epochs N_{pre} in brackets.

	MSE			MAE			R2			Mean true
	NN_{λ^r}	NN_{λ^c}	NN_d	NN_{λ^r}	NN_{λ^c}	NN_d	NN_{λ^r}	NN_{λ^c}	NN_d	Profits (€)
PTO NN	80.503	7416.628	0.005	6.242	59.745	0.042	0.588	0.701	−0.571	10473.60
PTO RNN	138.327	10601.233	0.003	8.773	74.739	0.044	0.292	0.573	0.020	9935.75
PTO LSTM	131.151	11222.596	0.004	8.248	77.919	0.044	0.328	0.548	−0.059	9975.48
DFL LSTM ($\gamma = 0$)	141.466	15395.836	0.005	8.887	89.031	0.037	0.276	0.380	−0.391	10877.16
DFL LSTM ($\gamma = 1$)	158.962	10405.753	0.013	9.378	74.933	0.088	0.186	0.581	−2.686	10879.69
DFL LSTM ($\gamma = 10$)	160.800	12227.220	0.005	9.810	80.955	0.037	0.177	0.508	−0.384	10880.88
DFL LSTM ($\gamma = 100$)	161.802	11677.218	0.005	9.929	79.298	0.037	0.171	0.530	−0.382	10876.98
DFL NN ($\gamma = 0$)	151.527	12682.537	0.014	9.464	80.604	0.081	0.224	0.489	−3.083	10989.60
DFL NN ($\gamma = 1$)	131.510	12032.767	0.015	8.682	78.188	0.083	0.327	0.515	−3.258	10984.76
DFL NN ($\gamma = 10$)	77.416	7821.279	0.011	6.264	61.657	0.068	0.604	0.685	−2.238	10857.44
DFL NN ($\gamma = 100$)	78.884	7478.019	0.007	6.143	59.677	0.050	0.596	0.699	−1.078	10902.32
DFL RNN ($\gamma = 0$)	179.892	13162.582	0.013	10.410	81.422	0.071	0.079	0.470	−2.667	10801.14
DFL RNN ($\gamma = 1$)	176.686	12865.908	0.005	10.175	81.167	0.037	0.095	0.482	−0.391	10877.16
DFL RNN ($\gamma = 10$)	173.850	12389.629	0.005	10.015	80.315	0.037	0.110	0.501	−0.383	10871.18
DFL RNN ($\gamma = 100$)	177.358	12277.300	0.005	10.077	80.345	0.037	0.092	0.506	−0.382	10877.16
DFL LSTM ($N_{pre} = 0$)	160.800	12227.220	0.005	9.810	80.955	0.037	0.177	0.508	−0.384	10880.88
DFL LSTM ($N_{pre} = 50$)	157.294	11405.510	0.005	9.782	78.435	0.037	0.195	0.541	−0.392	10877.16
DFL LSTM ($N_{pre} = 500$)	158.962	10405.753	0.013	9.378	74.933	0.088	0.186	0.581	−2.686	10879.69
DFL NN ($N_{pre} = 0$)	131.510	12032.767	0.015	8.682	78.188	0.083	0.327	0.515	−3.258	10984.76
DFL NN ($N_{pre} = 50$)	151.527	12682.537	0.014	9.464	80.604	0.081	0.224	0.489	−3.083	10989.60
DFL NN ($N_{pre} = 500$)	81.735	8694.752	0.011	6.557	65.647	0.068	0.581	0.650	−2.197	10937.41
DFL RNN ($N_{pre} = 0$)	176.686	12865.908	0.005	10.175	81.167	0.037	0.095	0.482	−0.391	10877.16
DFL RNN ($N_{pre} = 50$)	173.278	12205.168	0.005	9.883	80.001	0.037	0.113	0.508	−0.391	10866.14
DFL RNN ($N_{pre} = 500$)	165.656	11577.170	0.007	9.828	77.640	0.050	0.152	0.534	−1.020	10795.38

Table 4
Ex-post market performance comparison of PTO, direct DFL (dDFL), ELMR and best performing regressor trained with proposed DFL methodology.

	Mean true profits (€)	Total Vol Im (MW)	Imbalance Costs (€)	Degrad. (kWh)	R2		MAE		MSE	
					p_t^u	p_t^d	p_t^u	p_t^d	p_t^u	p_t^d
PTO NN	10,481.56	50.39	9,133.05	2.300	−0.339	−1.100	0.795	0.499	1.932	3.061
PTO LSTM	10,797.88	40.55	6,580.89	2.182	−0.800	−0.206	1.076	0.295	1.110	4.113
PTO RNN	9,910.42	38.23	6,134.94	2.083	−1.469	−0.062	1.485	0.256	0.977	5.643
dDFL NN	−618,150.00	7979.13	914,762.13	1.951	−29.061	−2.258	1.548	1.826	7.287	6.113
dDFL LSTM	−10,122.57	231.89	17,324.98	1.218	0.798	−0.666	0.033	1.389	3.725	0.041
dDFL RNN	−10,684.98	238.66	17,747.69	1.237	0.800	−0.371	0.033	1.235	3.066	0.041
ELM	−11,458.23	277.51	22,343.32	1.444	0.800	−0.091	0.044	1.122	2.440	0.041
Proposed DFL	11,009.00	39.34	6,652.51	2.216	−0.700	−0.264	1.031	0.314	1.163	3.884

less spread in power purchased than quadratic loss ($\beta = 2$), suggesting greater stability in decision-making. Pretraining epochs primarily affect RNN models, where higher pretraining epochs with quadratic loss lead to reduced power purchased. Other end-to-end learning processes incurs in excessive market penalties due to suboptimal bidding as Table 4 showcase. The proposed DFL model achieves an imbalance cost of 6652.51€, comparable to the best PTO model. This, in addition with better coefficient of determination and lower error metrics, lead to improved market performance.

3.6.3. Degradation analysis

Fig. 10 presents mean daily degradation for BESS in reserve market participation. For linear losses ($\beta = 1$), LSTM and RNN models trained with DFL consistently show lower daily values of degradation than their PTO counterparts. This pattern is not observed for NN models with low regularisation ($\gamma \leq 10$), which focus solely on maximising profits. For quadratic losses ($\beta = 2$), LSTM and RNN models follow similar trends, while NN models exhibit significantly higher degradation than PTO models. Pretraining epochs have little impact on degradation values for DFL models for linear losses ($\beta = 1$) and it shows mixed values in spread for quadratic losses ($\beta = 2$). DFL induces marginally higher degradation (2.216 kWh) than PTO RNN (2.083 kWh) but remains sustainable compared to direct DFL (1.218–1.951 kWh). Other end-to-end models, despite its poor economic performance, does not excessively degrade the battery, suggesting that its negative profits are not caused by aggressive dispatching but rather by suboptimal bidding. This reflects DFL's trade-off: prioritising profitability without drastic degradation.

3.6.4. Battery size and RE share analysis

Battery capacity and renewable energy share influence bidding performance in reserve markets. To quantify this effect, we evaluate the Mean Daily Profit Ratio (MDPR), which measures the relative deviation of a model's achieved profits from an ideal oracle benchmark. A higher MDPR indicates a greater shortfall from oracle profits, while lower values suggest better alignment with the oracle's decisions.

We train the DFL, PTO NN, and ELMR methods for 4, 20, 100, and 500 MW batteries (1-hour duration) and analyse MDPR across storage sizes and renewable penetration levels. As shown in Fig. 11(a), ELMR exhibits high variability, particularly for larger batteries, indicating poor scalability. In contrast, DFL and PTO maintain stable MDPR distributions, with DFL consistently aligning more closely with oracle profits, demonstrating superior adaptability to different storage capacities. Fig. 11(b) highlights the impact of renewable energy share. ELMR shows erratic performance, especially for RE shares > 10%, reflecting its sensitivity to market volatility. Meanwhile, DFL maintains stable performance across all RE levels, reinforcing its robustness in fluctuating market conditions.

4. Conclusion

This paper presents a methodology to train regressor models for BESS reserve market participation using a decision-oriented learning process. The methodology is compared to traditional PTO decision-making process, where the models are trained to minimise the MSE of

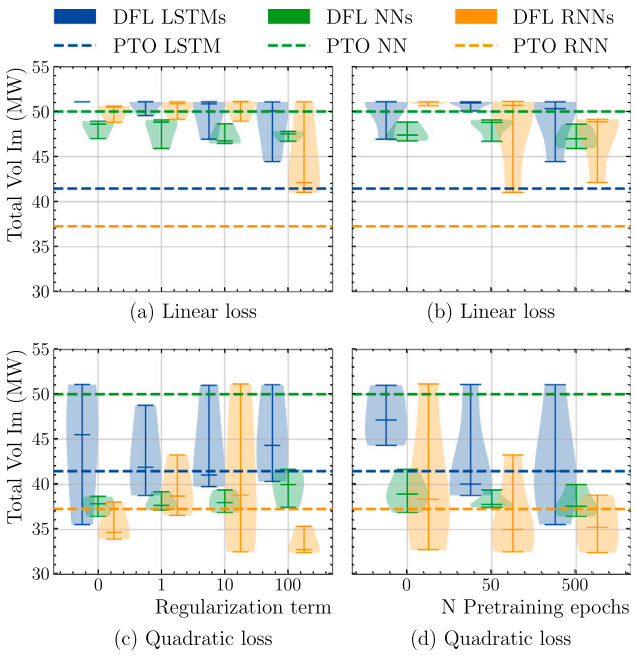


Fig. 9. Mean power purchased in the imbalance market by the BESS using PTO (dashed lines) and DFL (violin plots). Power purchased is shown by regularisation term for linear (a) and quadratic (c) loss functions, and by pretraining epochs for linear (b) and quadratic (d) loss functions.

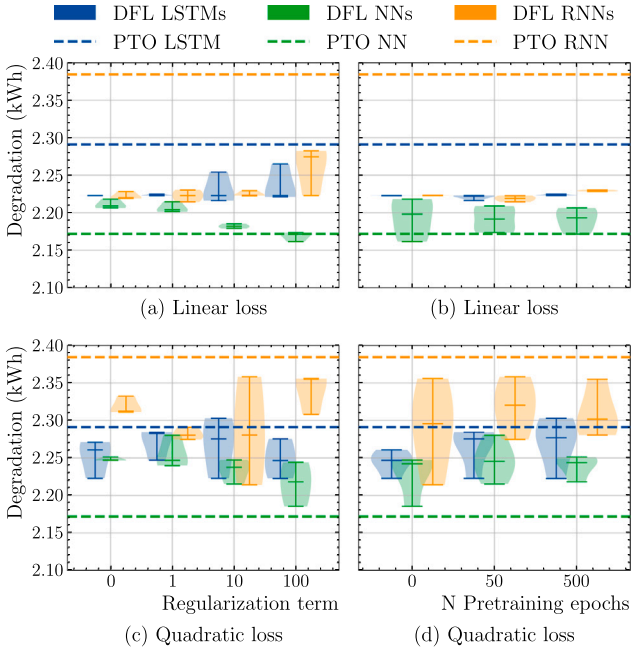


Fig. 10. BESS mean degradation for a day of operation in reserve markets using PTO (dashed lines) and DFL (violin plots). Degradation is shown by regularisation term for linear (a) and quadratic (c) loss functions, and by pretraining epochs for linear (b) and quadratic (d) loss functions. Oracle BESS mean daily degradation: 2.45 kWh.

the predictions, and other end-to-end learning frameworks which directly predict bidding decisions. The results show that the DFL methodology consistently outperforms PTO and other end-to-end learning frameworks in terms of true profits, with an increase of up to 9.47% for this case with respect to PTO . The DFL methodology also shows more stable results in terms of the power purchased in the imbalance market and the mean degradation of the BESS. In terms of the hyper-parameter

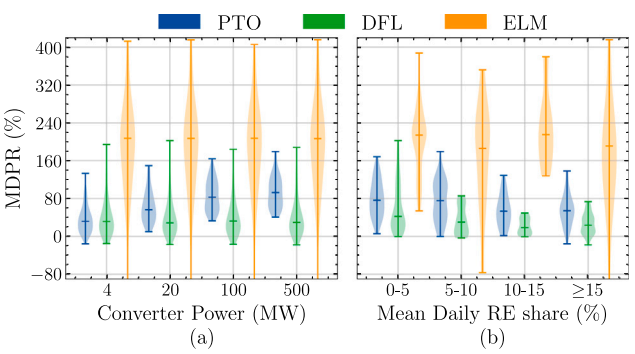


Fig. 11. Mean Daily Profit Ratio (MDPR) (%) distributions for different values of (a) battery size and (b) Mean Daily Share of Renewable Energy.

configuration of the DFL approach, the regularisation term γ, the order of the loss of profits β, and the number of pretraining epochs N_{pre} have a significant impact on the performance of the models, being the first two the most relevant regarding the profits obtained in the market, and the last one the most relevant regarding the accuracy of the predictions. We show that the DFL methodology can be a valuable tool to reflect the complexities of dynamic market conditions and the decision-making process of BESSs in reserve markets, providing consistently better performing regressor models compared to traditional PTO methodologies. However, this work is limited to Belgian reserve markets and a BESSs. Future research could address the challenge of stacking multiple services for batteries and incorporating distributed energy storage systems or aggregation techniques, enabling participation in multiple markets and extending the applicability of this framework.

CRediT authorship contribution statement

Ángel Paredes: Writing – original draft, Software, Methodology, Data curation, Conceptualization. **Jean-François Toubeau:** Writing – review & editing, Methodology. **José A. Aguado:** Writing – review & editing. **François Vallée:** Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Angel Paredes Parrilla reports financial support was provided by Spain Ministry of Education Vocational Training and Sports. Jose Antonio Aguado Sanchez reports financial support was provided by Spain Ministry of Science and Innovation. Jose Antonio Aguado Sanchez reports financial support was provided by Horizon Europe. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] G. Angenendt, M. Merten, S. Zurmühlen, D.U. Sauer, Evaluation of the effects of frequency restoration reserves market participation with photovoltaic battery energy storage systems and power-to-heat coupling, Appl. Energy 260 (2020) 114186, http://dx.doi.org/10.1016/j.apenergy.2019.114186.

[2] J. Kruse, B. Schäfer, D. Witthaut, Secondary control activation analysed and predicted with explainable AI, Electr. Power Syst. Res. 212 (July) (2022) 108489, http://dx.doi.org/10.1016/j.epsr.2022.108489.

[3] A.V. Vykhodtsev, D. Jang, Q. Wang, W. Rosehart, H. Zareipour, A review of modelling approaches to characterize lithium-ion battery energy storage systems in techno-economic analyses of power systems, Renew. Sustain. Energy Rev. 166 (June) (2022) 112584, http://dx.doi.org/10.1016/j.rser.2022.112584.
[4] M. Bulut, E. Özcan, How to build a state-of-the-art battery energy storage market? Challenges, opportunities, and future directions, J. Energy Storage 86 (2024) http://dx.doi.org/10.1016/j.est.2024.111174.
[5] M.S. Ibrahim, W. Dong, Q. Yang, Machine learning driven smart electric power systems: Current trends and new perspectives, Appl. Energy 272 (May) (2020) 115237, http://dx.doi.org/10.1016/j.apenergy.2020.115237.
[6] J. Cardo-Miota, E. Pérez, H. Beltran, Deep learning-based forecasting of the automatic frequency reserve restoration band price in the iberian electricity market, Sustain. Energy, Grids Networks 35 (2023) 101110, http://dx.doi.org/10.1016/j.segan.2023.101110.
[7] M. Merten, F. Rücker, I. Schoeneberger, D.U. Sauer, Automatic frequency restoration reserve market prediction: Methodology and comparison of various approaches, Appl. Energy 268 (December 2019) (2020) 114978, http://dx.doi.org/10.1016/j.apenergy.2020.114978.
[8] J.F. Toubeau, J. Bottieau, Z. De Greeve, F. Vallee, K. Bruninx, Data-driven scheduling of energy storage in day-ahead energy and reserve markets with probabilistic guarantees on real-time delivery, IEEE Trans. Power Syst. 36 (4) (2021) 2815–2828, http://dx.doi.org/10.1109/TPWRS.2020.3046710.
[9] J.I. Pérez-Díaz, I. Guisández, M. Chazarra, A. Helseth, Medium-term scheduling of a hydropower plant participating as a price-maker in the automatic frequency restoration reserve market, Electr. Power Syst. Res. 185 (August 2019) (2020) 106399, http://dx.doi.org/10.1016/j.epsr.2020.106399.
[10] Á. Paredes, J. Aguado, C. Essayeh, Y. Xia, I. Savelli, T. Morstyn, Stacking revenues from flexible DERs in multi-scale markets using tri-level optimization, IEEE Trans. Power Syst. (2023) 1–13, http://dx.doi.org/10.1109/TPWRS.2023.3286178.
[11] R. Smets, K. Bruninx, J. Bottieau, J.-F. Toubeau, E. Delarue, Strategic implicit balancing with energy storage systems via stochastic model predictive control, IEEE Trans. Energy Mark. Policy Regul. 1 (4) (2023) 373–385, http://dx.doi.org/10.1109/tempr.2023.3267552.
[12] R. Kumar, M.J. Wenzel, M.J. Ellis, M.N. Elbsat, K.H. Drees, V.M. Zavala, A stochastic model predictive control framework for stationary battery systems, IEEE Trans. Power Syst. 33 (4) (2018) 4397–4406, http://dx.doi.org/10.1109/TPWRS.2017.2789118.
[13] J. Cao, D. Harrold, Z. Fan, T. Morstyn, D. Healey, K. Li, Deep reinforcement learning-based energy storage arbitrage with accurate lithium-ion battery degradation model, IEEE Trans. Smart Grid 11 (5) (2020) 4513–4521, http://dx.doi.org/10.1109/TSG.2020.2986333.
[14] Q. Tang, H. Guo, Q. Chen, Multi-market bidding behavior analysis of energy storage system based on inverse reinforcement learning, IEEE Trans. Power Syst. 37 (6) (2022) 4819–4831, http://dx.doi.org/10.1109/TPWRS.2022.3150518.
[15] B. Huang, J. Wang, Deep-reinforcement-learning-based capacity scheduling for PV-battery storage system, IEEE Trans. Smart Grid 12 (3) (2021) 2272–2283, http://dx.doi.org/10.1109/TSG.2020.3047890.
[16] S. soroush Karimi madahi, G. Gokhale, M.-S. Verwee, B. Claessens, C. Develder, Control policy correction framework for reinforcement learning-based energy arbitrage strategies, in: The 15th ACM International Conference on Future and Sustainable Energy Systems, ACM, New York, NY, USA, 2024, pp. 123–133, http://dx.doi.org/10.1145/3632775.3661948.
[17] J. Li, C. Wang, Y. Zhang, H. Wang, Temporal-aware deep reinforcement learning for energy storage bidding in energy and contingency reserve markets, IEEE Trans. Energy Mark. Policy Regul. 2 (3) (2024) 1–15, http://dx.doi.org/10.1109/tempr.2024.3372656.
[18] T. Carriere, G. Kariniotakis, An integrated approach for value-oriented energy forecasting and data-driven decision-making application to renewable energy trading, IEEE Trans. Smart Grid 10 (6) (2019) 6933–6944, http://dx.doi.org/10.1109/TSG.2019.2914379.
[19] C. Zhao, C. Wan, Y. Song, Operating reserve quantification using prediction intervals of wind power: An integrated probabilistic forecasting and decision methodology, IEEE Trans. Power Syst. 36 (4) (2021) 3701–3714, http://dx.doi.org/10.1109/TPWRS.2021.3053847.
[20] C. Lu, W. Jiang, C. Wu, Effective end-to-end learning framework for economic dispatch, IEEE Trans. Netw. Sci. Eng. 9 (4) (2022) 2673–2683, http://dx.doi.org/10.1109/TNSE.2022.3168845.
[21] A.N. Elmachtoub, P. Grigas, Smart predict, then optimize, Manag. Sci. 68 (1) (2022) 9–26, http://dx.doi.org/10.1287/mnsc.2020.3922.
[22] H. Zhang, R. Li, M. Sun, T. Fei, Adaptive decision-objective loss for forecast-then-optimize in power systems, 2023, URL https://arxiv.org/abs/2312.13501.
[23] D. Wahdany, C. Schmitt, J.L. Cremer, More than accuracy: end-to-end wind power forecasting that optimises the energy system, Electr. Power Syst. Res. 221 (April) (2023) 109384, http://dx.doi.org/10.1016/j.epsr.2023.109384.
[24] A. Stratigakos, S. Camal, A. Michiorri, G. Kariniotakis, Prescriptive trees for integrated forecasting and optimization applied in trading of renewable energy, IEEE Trans. Power Syst. 37 (6) (2022) 4696–4708, http://dx.doi.org/10.1109/TPWRS.2022.3152667.
[25] P. Ellinas, V. Kekatos, G. Tsaousoglou, Decision-focused learning under decision dependent uncertainty for power systems with price-responsive demand, Electr. Power Syst. Res. 235 (101036723) (2024) 110665, http://dx.doi.org/10.1016/j.epsr.2024.110665.
[26] L. Sang, Y. Xu, H. Long, Q. Hu, H. Sun, Electricity price prediction for energy storage system arbitrage: A decision-focused approach, IEEE Trans. Smart Grid 13 (4) (2022) 2822–2832, http://dx.doi.org/10.1109/TSG.2022.3166791, arXiv:2305.00362.
[27] D. Naware, A. Mitra, Data-driven technology applications in planning, demand-side management, and cybersecurity for smart household community, IEEE Trans. Artif. Intell. 5 (10) (2024) 4868–4883, http://dx.doi.org/10.1109/TAI.2024.3417389.
[28] R. Li, H. Zhang, M. Sun, F. Teng, C. Wan, S. Pineda, G. Kariniotakis, Decision-oriented learning for future power system decision-making under uncertainty, 2024, URL https://arxiv.org/abs/2401.03680.
[29] European Union Agency for the Cooperation of Energy Regulators (ACER), Implementation framework for the European platform for the exchange of balancing energy from frequency restoration reserves with automatic activation, 2022, pp. 1–34, URL https://eepublicdownloads.entsoe.eu/clean-documents/nc-tasks/220921_ACER%20Decision%2015-2022%20on%20the%20Amendment%20of%20the%20aFRRIF%20-%20Annex%20II.pdf.
[30] Á. Paredes, J.A. Aguado, Revenue stacking of BESSs in wholesale and aFRR markets with delivery guarantees, Electr. Power Syst. Res. 234 (2024) 110633, http://dx.doi.org/10.1016/j.epsr.2024.110633.
[31] Y. Zhou, C. Essayeh, T. Morstyn, Aggregated feasible Active Power Region for distributed energy resources with a distributionally robust joint probabilistic guarantee, IEEE Trans. Power Syst. PP (2024) 1–15, http://dx.doi.org/10.1109/TPWRS.2024.3392622.
[32] K. Garifi, K. Baker, D. Christensen, B. Touri, Convex relaxation of grid-connected energy storage system models with complementarity constraints in DC OPF, IEEE Trans. Smart Grid 11 (5) (2020) 4070–4079, http://dx.doi.org/10.1109/TSG.2020.2987785.
[33] N. Collath, B. Tepe, S. Englberger, A. Jossen, H. Hesse, Aging aware operation of lithium-ion battery energy storage systems: A review, J. Energy Storage 55 (PC) (2022) 105634, http://dx.doi.org/10.1016/j.est.2022.105634.
[34] H.W. Kuhn, A.W. Tucker, Nonlinear programming, in: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950, University of California Press, Berkeley and Los Angeles, 1951, pp. 481–492.
[35] A. Bojańczyk, Complexity of solving linear systems in different models of computation, SIAM J. Numer. Anal. 21 (3) (1984) 591–603, http://dx.doi.org/10.1137/0721041.
[36] A. Agrawal, S. Barratt, S. Boyd, Learning convex optimization models, IEEE/CAA J. Autom. Sin. 8 (2021) 1355–1364, http://dx.doi.org/10.1109/JAS.2021.1004075.
[37] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, Z. Kolter, Differentiable Convex Optimization Layers, 2019, URL https://arxiv.org/abs/1910.12430.
[38] ENTSO-E, ENTSO-E Transparency Platform, 2023, URL https://transparency.entsoe.eu/transmission-domain/r2/dayAheadPrices/show?areaType=BZN.
[39] ELIA Group, ELIA Grid Data, 2023, URL www.elia.be/en/grid-data/data-download.
[40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, 2019, URL https://arxiv.org/abs/1912.01703.
[41] University of Málaga, Supercomputing and bioinformatics center, 2024, URL https://www.scbi.uma.es/web/.
[42] J. Bergstra, D. Yamins, D. Cox, Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures, in: S. Dasgupta, D. McAllester (Eds.), Proceedings of the 30th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 28, PMLR, Atlanta, Georgia, USA, 2013, pp. 115–123, URL https://proceedings.mlr.press/v28/bergstra13.html.
[43] The Apache Software Foundation, SparkR: R front end for 'apache spark', 2024, URL https://www.apache.org, https://spark.apache.org.