



Decision-Focused Retraining of Forecast Models for Optimization Problems in Smart Energy Systems

Maximilian Beichter
maximilian.beichter@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Dorina Werling
dorina.werling@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Benedikt Heidrich
benedikt.heidrich@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Kaleb Phipps
kaleb.phipps@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Oliver Neumann
oliver.neumann@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Nils Friederich
nils.friederich@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Ralf Mikut
ralf.mikut@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Veit Hagenmeyer
veit.hagenmeyer@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

ABSTRACT

In order to enable the energy transition, a higher share of renewable energy sources is required in the electricity grid. However, the volatile nature of these renewable sources can lead to stability issues. Therefore, countermeasures must be integrated into modern electricity grids to maintain stability. However, many countermeasures rely on optimization problems on multiple grid levels to be successfully integrated. Furthermore, these optimization problems often require forecasts that are tailored to deliver value for the considered optimization problem. Nevertheless, existing applications of decision-focused learning to provide this value scale poorly for energy system optimization problems. Therefore, we propose a novel method called Decision-Focused Retraining that combines prediction-focused learning and decision-focused learning. In this method, an existing forecasting model is retrained to generate forecasts delivering increased value for the optimization problem. First, a prediction-focused learning approach with a suitable base loss is used to pre-train the forecasting model. Afterward, the model is fine-tuned by combining a global instance-independent surrogate NN with the prediction-focused base loss to optimize the forecasting model. We evaluate our approach on an exemplary optimization problem, the dispatchable feeder optimization problem, considering over 199 buildings, which leads to an improvement of at least 7.29%.

CCS CONCEPTS

• **Computing methodologies** → *Learning paradigms; Supervised learning*; • **Applied computing** → *Forecasting*.



This work is licensed under a Creative Commons Attribution International 4.0 License.

E-Energy '24, June 04–07, 2024, Singapore, Singapore
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0480-2/24/06
<https://doi.org/10.1145/3632775.3661952>

KEYWORDS

Decision-focused learning, energy systems, forecasting, predict then optimize, applied optimization

ACM Reference Format:

Maximilian Beichter, Dorina Werling, Benedikt Heidrich, Kaleb Phipps, Oliver Neumann, Nils Friederich, Ralf Mikut, and Veit Hagenmeyer. 2024. Decision-Focused Retraining of Forecast Models for Optimization Problems in Smart Energy Systems. In *The 15th ACM International Conference on Future and Sustainable Energy Systems (E-Energy '24)*, June 04–07, 2024, Singapore, Singapore. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3632775.3661952>

1 INTRODUCTION

To develop a low-carbon energy system, a higher share of distributed renewable energy sources must be integrated into the electricity grid. However, the volatility of these renewable energy sources can lead to stability issues. Thus, countermeasures to maintain grid stability are required. These countermeasures typically focus on directly integrating power electronics to counteract instability or including energy storage systems that increase flexibility and, thus, stability [22]. Furthermore, these countermeasures must often be managed through optimization problems to be successfully integrated into the energy system, especially when considering energy storage systems [22]. In addition, these optimization problems are integrated into different levels in the energy system, for example, on the grid level [5], in microgrids [25], or on building level [24] [3]. Therefore, these optimization problems must operate successfully and efficiently on all grid levels, which leads to two requirements.

First, many of these optimization problems require forecasts as inputs. Often, these forecasts are designed to maximize the precision of the predictions given a quality metric (forecast quality). However, maximizing the forecast quality may not always result in the most useful forecasts for the considered optimization problem [7, 16] and [26]. Thus, it may be useful to evaluate the forecast value, which is the practical utility cost impact on the considered optimization problem derived from a forecast. It goes beyond the usage of forecasting quality and considers the impact of forecasts

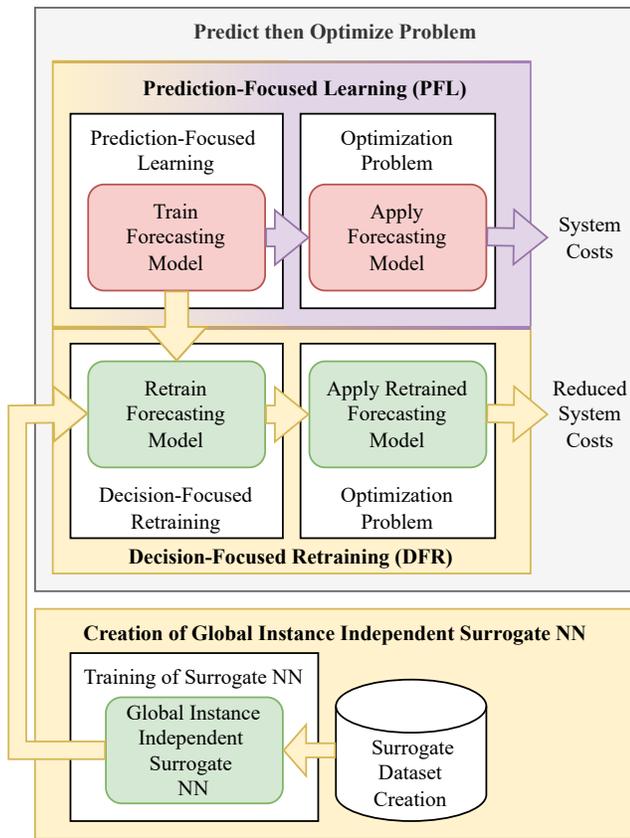


Figure 1: An overview of the traditional Prediction-Focused Learning (PFL) approach and our proposed Decision-Focused Retraining (DFR) method. DFR consists of four main steps to obtain the retrained forecasting model. First, a surrogate dataset is created. Second, this dataset is used to train a global instance-independent surrogate Neuronal Network (NN). Third, a PFL forecasting model is trained. Fourth, the PFL forecasting model is retrained by incorporating the forecast value into the loss through the trained surrogate NN. This retrained forecasting model reduces the system costs for the considered optimization problem.

in decision-making and planning. A forecast holds high value if it contributes positively to the decision process, enabling better actions and, thus, more profitable outcomes based on the optimization problem results.

Second, due to the high number of instances (e.g. buildings) in the lower disaggregated grid level, it is important that the optimization problem is efficiently scalable. Importantly, generating forecasts and considering the forecast value for the given optimization problem must also scale efficiently.

To address the first requirement, the difference between forecast quality and forecast value motivates two learning strategies for prediction models proposed in [14]. Prediction-Focused Learning (PFL) is designed to minimize a forecasting quality metric, and decision-focused learning aims to maximize the forecast value. However,

although previous work mostly considers both prediction-focused learning and decision-focused learning separately, existing research in literature fails to combine the advantage of both learning strategies in a data-driven context for non-convex optimization problems. Furthermore, previous work does not address the across various instances scalability issues that are crucial for energy systems.

Therefore, in the present paper, we propose a novel method, Decision-Focused Retraining (DFR), which combines both strategies: prediction-focused learning and retraining in decision-focused manner through global instance-independent surrogate NNs. Figure 1 presents an overview of the DFR method, which consists of four main steps. First, we create a surrogate dataset, before we train a global instance-independent surrogate NN in a second step. This global surrogate NN is trained to predict the costs of the optimization problem that are associated with a certain forecast. Thereby, global refers to the surrogate accepting all values of the forecasting space, and not only values very close to the ground truth value [20]. Furthermore, instance-independent implies that only one surrogate must be trained, and this surrogate can then be applied to all instances within the same domain. In an energy system, these instances are, for example, different buildings with similar characteristics. To achieve this, the surrogate NN considers the actual values, the forecasts, and further information about the building as inputs. Based on this information, it predicts the optimization problem costs, which correspond to the forecast value. Note that the surrogate network is differentiable by design as it is modeled by a NN architecture. In the third step, we use prediction-focused learning to train an instance-dependent forecasting model. This forecasting model is not yet adapted to the optimization problem. Thus, in the final step, this forecasting model is retrained using the surrogate model as a judge. In particular, we retrain the forecasting model to minimize the optimization problem costs that the surrogate NN predicts. The resulting retrained forecasting model generates forecasts with a higher value for the considered optimization problem and thus reduces the system costs.

The remainder of the paper is structured as follows: We present an overview of the related work and highlight our contribution in Section 2. In Section 3, we formally introduce the proposed decision-focused retraining approach. We then introduce an exemplary optimization problem, the dispatchable feeder [3] in Section 4, which we use to evaluate the DFR. In Section 5, we describe the applied experimental setup before presenting our evaluation results in Section 6. We discuss these results and key insights in Section 7 before concluding in Section 8.

2 RELATED WORK

The related work is divided into three subsections. The first subsection presents a general overview of decision-focused learning. In the second subsection, we analyze various applications of decision-focused learning in the energy system. Finally, the third subsection highlights our contributions.

2.1 Decision-Focused Learning Approaches

In decision-focused learning, a Machine Learning (ML) model is trained to maximize the forecast value instead of the commonly

used prediction-focused forecasting models that maximize the forecast quality. To achieve this, different approaches exist in the literature. The first approach is to integrate the optimization problem into the forecasting model and train this integrated model. However, this requires differentiation through the optimization problem, which is not straightforward. While there are analytical approaches tailored to specific classes of optimization problems, e.g. [2, 8, 30, 31] for strongly convex and quadratic optimization problems, the generalization to non-convex optimization problems is lacking.

The second type of approaches uses differentiable surrogate loss functions [4, 6, 9, 20, 21, 29, 32]. More precisely, while Elmachoub and Grigas [9] approximates the analytical description of the integrated model for linear optimization problems, multiple alternate approaches assume specific parametric loss function families and learn the respective parameters with respect to the forecast value [4, 6, 20, 21]. Such restrictive assumptions are not made by Zhang et al. [29], which empirically approximates the relation between forecast error and forecast value. However, for complex and non-convex optimization problems, the forecast value is not solely dependent on the forecast error.

Finally, the last approach is to simultaneously learn the forecast model and the surrogate model for the objective of the optimization problem based on the resulting forecast [32].

While the proposed approach does not require the integration of the optimization problem into the forecasting model, nor does the loss belong to a specific function family, it couples the training of the surrogate NN with the training of the forecast model. Thus, it requires fitting a surrogate model for each forecaster, which might decrease its scalability.

2.2 Decision-Focused Learning in Energy Systems

Various decision-focused learning papers show energy-relevant use cases, specifically for different energy-related downstream optimizations [8, 13–15, 19, 29–31]. These optimizations are often simplified compared to real-world optimization but already consider a wide variety of different tasks. Specifically, Kong et al. [13] using decision-focused learning applied to wind power bidding with a convex objective function, whilst Zhang et al. [30] and Zhang et al. [31] focus on day-ahead and real-time energy dispatch of a virtual power plant operator in charge of wind power in two stages, without accounting for battery storage. Furthermore, Zhang et al. [29] consider two energy-related use cases by solving both a day-ahead economic dispatch optimization problem to schedule the power output of generators and also the intraday power balance problem, which schedules battery storage. The approach from Zhang et al. [29] is performed on a 30-bus bar grid with three batteries and, therefore, considers the grid perspective. Mandi et al. [14] and Mulamba et al. [15] focus on an optimization problem that schedules computational tasks with respect to the energy pricing from Simonis et al. [23], whilst a probabilistic energy arbitrage task and a generator scheduling task are considered by Donti et al. [8]. Furthermore, the generator scheduling task is again addressed by Kong et al. [12], whilst a cost-oriented energy storage arbitrage task is dealt with by Sang et al. [19].

Whilst the identified literature shows that many applications have considered decision-focused learning in energy systems, these approaches are not easily scalable for multiple instances and are usually not suitable for non-convex multiple-level optimization problems, which are often found in energy systems.

2.3 Contribution of the Present Paper

Given the identified challenges in existing literature, the contribution of the present paper is three-fold:

First, this paper solves the scalability issue in classical decision-focused learning by using a globally instance-independent learned surrogate model and combining it with a prediction-focused loss. Second, to the best of the author's knowledge, this is the first paper that introduces exogenous features into the surrogate model. Using exogenous information in the surrogate model strengthens its ability to model the costs of the optimization problem as well as its ability to generalize on further unseen instances. Third, we apply the proposed novel decision-focused learning approach to a non-convex multiple-level optimization problem and demonstrate that globally instance-independent learned surrogate models can improve the forecast value, even for this complex class of optimization problems.

3 DECISION-FOCUSED RETRAINING (DFR)

In this section, we introduce Decision-Focused Retraining (DFR) of a NN regarding the forecast value. It consists of two parts, each with two steps. The first part generates a surrogate NN for the forecast value by first creating a surrogate dataset and then training the surrogate NN. The second part aims to generate a forecasting NN that considers the forecast value. Therefore, the forecasting NN is initially trained in a prediction-focused manner before being retrained using the surrogate NN to consider the forecast value. In the following, we describe each of these steps in detail.

Part 1: Generation of the Surrogate Neural Network for the Forecast Value

Step 1.1: Creation of the Surrogate Dataset. First, we create the dataset to train the surrogate NN for the forecast value. Such a dataset should consider all relevant aspects that affect the forecast value, such as forecasts with different characteristics and exogenous factors. The surrogate dataset $\mathcal{D}_{\text{surr}}$ can then be written as

$$\mathcal{D}_{\text{surr}} := \{(x_{\text{surr}}, y_{\text{surr}})_i\} = \{((\hat{y}, y, e), v(\hat{y}, y, e))_i\} \quad (1)$$

with Ground Truth (GT) y , forecast \hat{y} , exogenous factors e , the resulting forecast value $v(\hat{y}, y, e)$, and index $i = 1, \dots, N$ denoting the i 'th dataset tuple.

Step 1.2: Training of the Forecast Value Surrogate Neural Network. In the next step, we train the surrogate NN f_{surr} to forecast the forecast value using the surrogate dataset $\mathcal{D}_{\text{surr}}$ and loss function L_{surr} . The parameters of the trained surrogate NN are then

$$\theta_{\text{surr}}^* := \operatorname{argmin}_{\theta_{\text{surr}}} \sum_i L_{\text{surr}}(y_{\text{surr},i}, f_{\text{surr}}(x_{\text{surr},i}; \theta_{\text{surr}})) \quad (2)$$

and the output of the trained surrogate NN is

$$f_{\text{surr}}(x_{\text{surr},i}; \theta_{\text{surr}}^*) = \hat{y}_{\text{surr},i} \approx y_{\text{surr},i} = v(\hat{y}, y, e)_i. \quad (3)$$

Therefore, the differentiable function f_{surr} can be used as a proxy for the forecast value.

Part 2: Generation of the Forecasting Neural Network regarding the Forecast Value

Step 2.1: Training of the Forecasting Neural Network. The next step involves the prediction-focused training of a forecasting NN. In doing so, we utilize a dataset

$$\mathcal{D}_{\text{fc}} := \{(x_{\text{fc}}, y_{\text{fc}})_i\}, \quad (4)$$

with features $x_{\text{fc},i}$, ground truth historical values $y_{\text{fc},i}$, and index $i = 1, \dots, N$ denoting the i 'th dataset tuple. With this dataset and the loss function L_{base} , the forecasting NN is trained and optimal parameters

$$\theta_{\text{fc}}^* := \underset{\theta_{\text{fc}}}{\operatorname{argmin}} \sum_i L_{\text{base}}(y_{\text{fc},i}, f_{\text{fc}}(x_{\text{fc},i}; \theta_{\text{fc}})) \quad (5)$$

resulting in forecasts

$$f_{\text{fc}}(x_{\text{fc},i}; \theta_{\text{fc}}^*) = \hat{y}_{\text{fc},i} \quad (6)$$

are obtained.

Step 2.2: Retraining of the Forecasting Neural Network regarding the Forecast Value. The final step is the main contribution of the proposed method, where we retrain the forecasting NN to increase the forecast value (schematically in Figure 2). For this retraining, we use the trained surrogate NN f_{surr} with optimal parameters θ_{surr}^* , which are kept constant during retraining. Further, we use a third dataset

$$\mathcal{D}_{\text{re}} := \{(x_{\text{fc}}, (y_{\text{fc}}, e_{\text{fc}})), y_{\text{fc}}\}_i. \quad (7)$$

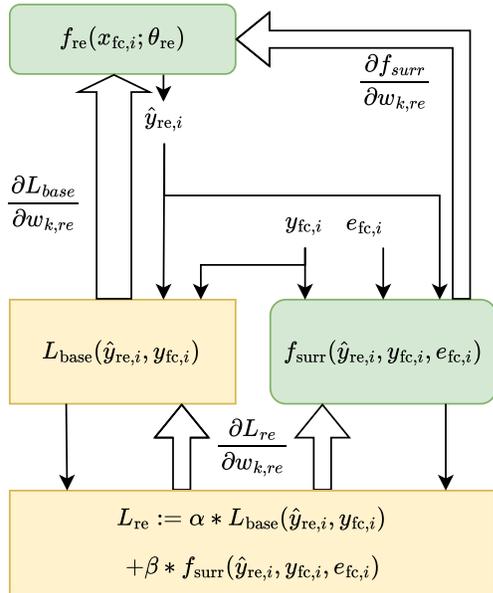


Figure 2: The retraining in Step 2.2 of the proposed method. The rounded green boxes are NNs. The corned yellow boxes display the loss functions used. The bold arrows symbolize the gradient flow, and the small arrows the data flow.

This dataset extends the training dataset used for the generation of the forecasting NN with additional exogenous information about the optimization problem. Therefore, $y_{\text{fc},i}$ remains the same as in the forecasting training dataset. The retraining consists of the following steps.

First, the retrained NN f_{re} is initialized with the parameters of the forecasting NN θ_{fc}^* , making the retrained NN equal to the forecasting NN.

Second, the parameters of the retrained NN are retrained¹ so that

$$\theta_{\text{re}}^* := \underset{\theta_{\text{re}}}{\operatorname{argmin}} \sum_i L_{\text{re}}(\hat{y}_{\text{re},i}, y_{\text{fc},i}, f_{\text{surr}}(\hat{y}_{\text{re},i}, y_{\text{fc},i}, e_{\text{fc},i}; \theta_{\text{surr}}^*)), \quad (8)$$

with

$$f_{\text{re}}(x_{\text{fc},i}; \theta_{\text{re}}) = \hat{y}_{\text{re},i} \quad (9)$$

and the retraining loss L_{re} as

$$L_{\text{re}} := \alpha \cdot L_{\text{base}}(\hat{y}_{\text{re},i}, y_{\text{fc},i}) + \beta \cdot f_{\text{surr}}(\hat{y}_{\text{re},i}, y_{\text{fc},i}, e_{\text{fc},i}) \quad (10)$$

with weights $\alpha \in (0, 1)$ and $\beta \in (0, 1)$. We further specify the weights as

$$\alpha = 1 - \frac{L_{\text{base}}}{L_{\text{base}} + f_{\text{surr}}}, \quad \beta = 1 - \frac{f_{\text{surr}}}{L_{\text{base}} + f_{\text{surr}}} \quad (11)$$

to keep the base loss's influence equal to the surrogate NN's influence. This is necessary to avoid convergence of the retrained NN into regions where the surrogate NN has not seen comparable data and therefore the surrogate does not model the forecast value sufficiently.

In more depth, the retraining can be formulated in terms of the weights of the forecasting NN $w_{k,\text{re}} \in \theta_{\text{re}}$ for the backpropagation. More precisely, we use the partial gradient of L_{re} regarding $w_{k,\text{re}}$, which can be written as

$$\frac{\partial L_{\text{re}}}{\partial w_{k,\text{re}}} = \underbrace{\alpha \frac{\partial L_{\text{base}}}{\partial w_{k,\text{re}}}}_{\text{Prediction-Focused}} + \underbrace{\beta \frac{\partial f_{\text{surr}}}{\partial w_{k,\text{re}}}}_{\text{Decision-focused}}, \quad (12)$$

as L_{base} and f_{surr} are differentiable. This gradient is used to update the weights $w_{k,\text{re}}$ during backpropagation via

$$w_{k,\text{re,new}} = w_{k,\text{re,old}} - \eta \frac{\partial L_{\text{re}}}{\partial w_{k,\text{re,old}}}, \quad (13)$$

with learning rate $\eta > 0$.

4 SELECTED OPTIMIZATION PROBLEM: THE DISPATCHABLE FEEDER

To evaluate the proposed DFR method, we apply it to an exemplary optimization problem, namely the dispatchable feeder [24]. The dispatchable feeder is a two-level non-convex optimization problem, and as a result, existing decision-focused methods cannot simply be applied. Furthermore, the dispatchable feeder optimization problem is often solved on a building level, which highlights the importance of scalability. Finally, previous work has shown the importance of finding high-value forecasts for the dispatchable feeder [26], which motivates the need for a decision-focused method. These

¹For simplicity the forecast value is introduced as cost. Therefore, in Equation (8) maximizing the forecast value for the downstream application is equivalent to minimizing the cost.

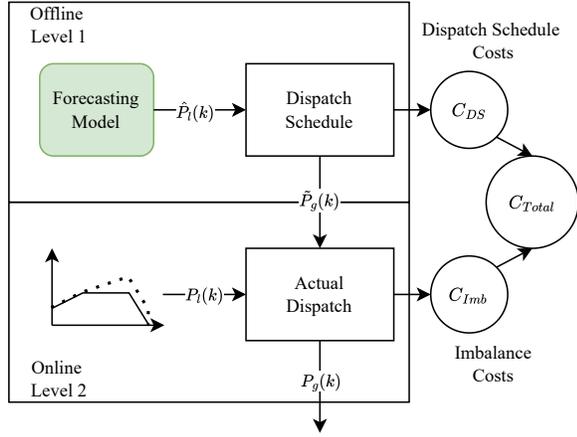


Figure 3: The two levels of the optimization problems (edged boxes) with their related costs (circles) and models (rounded boxes). Showing further the dataflow of the schedules and the forecasts.

three characteristics mean that the dispatchable feeder optimization problem is ideal for evaluating the proposed DFR method.

In the following, we first introduce the dispatchable feeder optimization problem. In formulating this optimization problem, we stay close to [26] and [27]. Afterward, we present the utilized forecast value for the dispatchable feeder.

4.1 Optimization Problem

A dispatchable feeder consists of two components [24] [3]. The first component is the inflexible and volatile prosumption of the residential building, which is defined as the residential load minus the PV power generation. The second component is a flexible power and energy-constrained residential battery. Using these components, the dispatchable feeder intelligently manages the flexible battery to counteract the uncertainty inherent in the uncertain consumption. In doing so, the operation of the dispatchable feeder consists of two steps illustrated schematically in Figure 3. First, a day-ahead dispatch schedule is computed based on prosumption forecasts. Second, the actual dispatch is calculated under the consideration of the actual prosumption with the aim to minimize the deviation of the dispatch schedule. This operation is formulated as a hierarchical two-level non-convex optimization problem. To formally model the optimization problem, the time operation is discretized into time intervals indexed by $k \in \mathbb{N}$ with interval length of $\Delta t \in \mathbb{R}$. The components of the dispatchable feeder are then modeled as follows. The flexible battery is modeled by its active power input $P_s(k) \in [\underline{P}_s, \overline{P}_s]$ and SoE $E_s(k) \in [\underline{E}_s, \overline{E}_s]$ with lower and upper bounds $\underline{P}_s, \overline{P}_s \in \mathbb{R}$ and $\underline{E}_s, \overline{E}_s \in \mathbb{R}_{\geq 0}$. Thereby, the battery's power input transforms into the SoE according to

$$E_s(k+1) = E_s(k) + \Delta t \cdot (P_s(k) - \mu P_s^+(k) + \mu P_s^-(k)) \quad (14)$$

with loss coefficient $0 \leq \mu \leq 1$ and positive and negative directions of the battery's power input $P_s^+(k) \geq 0$ and $P_s^-(k) \leq 0$. The interaction with the grid is solely modeled by the active power exchange

with the dispatchable feeder. Therefore, it consists of the sum of the two components of the dispatchable feeder, namely the battery's power input and the uncertain building's prosumption.

With this description, the two levels of the optimization problem can be formulated in the following.

First Level: Computation of Dispatch Schedule. The first level determines offline the Dispatch Schedule (DS) $\tilde{P}_g(k) \in \mathbb{R}$ regarding costs for the following day, considering deterministic forecasts of the prosumption $\hat{P}_l(k) \in \mathbb{R}$. Thereby, the cost function considers both self-consumption and peak shaving

$$C_{DS}(\tilde{P}_g^+(k), \tilde{P}_g^-(k)) = c_q^+ \cdot (\tilde{P}_g^+(k) \cdot \Delta t)^2 + c_1^+ \cdot \tilde{P}_g^+(k) \cdot \Delta t + c_q^- \cdot (\tilde{P}_g^-(k) \cdot \Delta t)^2 + c_1^- \cdot \tilde{P}_g^-(k) \cdot \Delta t \quad (15)$$

with positive and negative directions of the DS $\tilde{P}_g^+(k) \geq 0$ and $\tilde{P}_g^-(k) \leq 0$ and cost coefficients $c_q^+, c_1^+, c_q^-, c_1^- \in \mathbb{R}_{\geq 0}$. The first level optimization problem is then formulated as

$$\begin{aligned} \min_{\{X\}_{k \in \mathcal{K}}} \sum_{k \in \mathcal{K}} C_{DS}(\tilde{P}_g^+(k), \tilde{P}_g^-(k)) \\ \text{s.t. for all } k \in \mathcal{K} \\ (14) \\ E_s(k_0) = E_s^0 \\ \tilde{P}_g(k) = P_s(k) + \hat{P}_l(k) \\ \tilde{P}_g(k) = \tilde{P}_g^+(k) + \tilde{P}_g^-(k) \\ \tilde{P}_g^+(k) \geq 0 \\ \tilde{P}_g^-(k) \leq 0 \\ P_s(k) = P_s^+(k) + P_s^-(k) \\ P_s^+(k) \geq 0 \\ P_s^-(k) \leq 0 \\ 0 = P_s^+(k) \cdot P_s^-(k) \\ \underline{P}_s \leq P_s(k) \leq \overline{P}_s \\ \underline{E}_s \leq E_s(k) \leq \overline{E}_s \end{aligned} \quad (16)$$

with discrete scheduling horizon \mathcal{K} , decision vector $X(k) = (\tilde{P}_g(k), \tilde{P}_g^+(k), \tilde{P}_g^-(k), E_s(k+1), P_s(k), P_s^+(k), P_s^-(k))^T$, parameters $E_s^0, \underline{P}_s, \overline{P}_s, \underline{E}_s, \overline{E}_s$, and deterministic forecasts $\hat{P}_l(k)$. Note, it is necessary to know or estimate the SoE at the beginning of scheduling $k_0 \in \mathbb{N}$.

Second Level: Calculation of the Actual Dispatch. After the computation of the DS, the actual dispatch is calculated for every time interval based on the actual prosumption $P_l(k) \in \mathbb{R}$. Thereby, the aim is to minimize the deviation of the corresponding computed DS $\Delta P_g(k) \in \mathbb{R}$ considering the technical constraints of the battery.

The second-level optimization problem can be formulated as

$$\begin{aligned}
& \min_{X(k)} (\Delta P_g(k))^2 \\
& (14) \\
& E_s(k) = E_s^k \\
& P_g(k) = P_s(k) + P_l(k) \\
& P_g(k) = \tilde{P}_g(k) + \Delta P_g(k) \\
& P_s(k) = P_s^+(k) + P_s^-(k) \\
& P_s^+(k) \geq 0 \\
& P_s^-(k) \leq 0 \\
& 0 = P_s^+(k) \cdot P_s^-(k) \\
& \underline{P}_s \leq P_s(k) \leq \bar{P}_s \\
& \underline{E}_s \leq E_s(k) \leq \bar{E}_s
\end{aligned} \tag{17}$$

with decision vector $X(k) = (P_g(k), E_s(k+1), P_s(k), P_s^+(k), P_s^-(k))^T$, parameters $\tilde{P}_g(k), P_l(k), E_s^k, \underline{P}_s, \bar{P}_s, \underline{E}_s, \bar{E}_s$, and actual dispatch $P_g(k) \in \mathbb{R}$. Note, the SoE in $k \in \mathbb{N}$ is known.

4.2 Forecast Value

In order to effectively retrain the forecast model for generating suitable forecasts for the optimization problem, it is essential to establish a metric to evaluate the performance of the optimization problem in relation to the corresponding forecast. Therefore, for the dispatchable feeder optimization problem, we introduce the daily total costs as considered forecast value with lower daily total costs implying a higher forecast value. More precisely, the total costs consider both the DS costs of the first level optimization problem in Equation (15) and the imbalance costs resulting from the deviation from the DS after the calculation of the actual dispatch in the second level optimization problem, see Figure 3. These imbalance costs are defined as

$$C_{\text{Imb}}(\Delta P_g(k)) = c_q^\Delta \cdot |\Delta P_g(k) \cdot \Delta t|^2 + c_l^\Delta \cdot |\Delta P_g(k) \cdot \Delta t| \tag{18}$$

where $\Delta P_g(k)$ is the difference between the actual dispatch and the DS and $c_q^\Delta \in \mathbb{R}_{\geq 0}$ and $c_l^\Delta \in \mathbb{R}_{\geq 0}$ are weighting parameters. The total costs can then be written as

$$C_{\text{Total}}(\tilde{P}_g^+(k), \tilde{P}_g^-(k), \Delta P_g(k)) = C_{\text{DS}}(\tilde{P}_g^+(k), \tilde{P}_g^-(k)) + \alpha \cdot C_{\text{Imb}}(\Delta P_g(k)), \tag{19}$$

with imbalance costs factor $\alpha \in \mathbb{R}_{\geq 0}$. For the daily total costs, we aggregate the total costs for each day.

5 EXPERIMENTAL SETUP AND EVALUATION STRATEGY

This section is divided into five subsections. The first subsection presents the parameter specifications of the selected optimization problem, namely the dispatchable feeder (see Section 4). The second subsection describes the used data. The third subsection displays the loss functions utilized for both the NNs and the evaluation. The fourth subsection describes the evaluation metrics used. The

last subsection describes how our DFR method is applied to the dispatchable feeder optimization problem.²

5.1 Parameter Specifications of the Dispatchable Feeder Optimization Problem

Table 1 shows the parameter specifications of the dispatchable feeder optimization problem. Note that the parameter α is set to 10, which results in a high weighting of the imbalance costs and, thus, high costs for deviations from the dispatch schedule.

5.2 Used Data

We use the "Ausgrid - Solar home electricity data" set [17] for the evaluation. This dataset contains load and PV power generation time series from 300 residential buildings in Australia. The dataset spans three years, from 1st July 2010 to 30th June 2013, and is measured at a 30-minute resolution using a gross meter. The residential buildings are randomly selected from the Ausgrid electricity network, which uses the residential electricity tariff.

For our evaluation, we resample the data to an hourly resolution and calculate the prosumption by subtracting the PV power generation from the load. Further, we split the dataset into training, validation and test datasets as displayed in Figure 4. First, we use a distinct set of buildings for each of the two parts of our proposed DFR method in Section 3 to ensure independence. For the first part – the generation of the surrogate NN – we use the first 50 buildings for training and buildings 51 to 100 for validation³. For the second

²Code is publicly available via GitHub: <https://github.com/KIT-IAI/Decision-Focused-Retraining>

³Buildings 16 and 19 are omitted due to convergence issues of the optimization problem.

Table 1: Parameter specifications of the Equations (15-19).

Parameter	Value
Δt	1 (hour)
\mathcal{K}	$\{k_s, \dots, k_s + 29\}$ ¹
c_q^+	0.05 (€/kWh ²)
c_l^+	0.3 (€/kWh)
c_q^-	0.05 (€/kWh ²)
c_l^-	0.15 (€/kWh)
\underline{P}_s	-5 (kW)
\bar{P}_s	5 (kW)
\underline{E}_s	0 (kWh)
\bar{E}_s	13.5 (kWh)
μ	0.05
E_s^0	day 1: 6 (kWh)
α	10 as proposed in [18]
c_q^Δ	0.05 (€/kWh ²)
c_l^Δ	0.3 (€/kWh)

¹ $k_s \in \mathbb{N}$ is the index of the time interval starting at midnight timestamp and schedule calculation starts at noon timestamp of the "Ausgrid - Solar home electricity data" set [17]

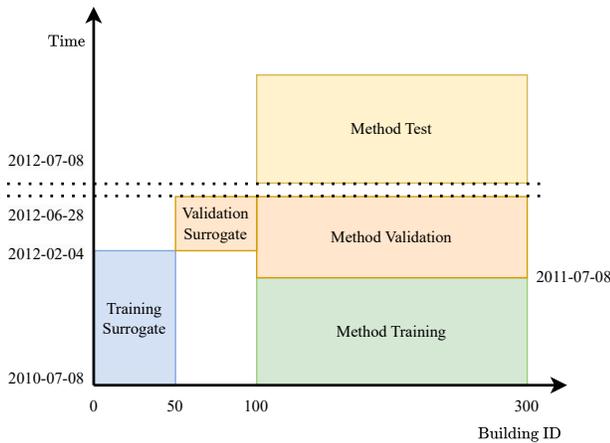


Figure 4: Displays the data split for the experimental setup, further it shows the data usage within the different parts of the experiment.

part – the generation of the forecasting NN – we use buildings 101 to 300 for training, validation, and testing⁴.

5.3 Loss Functions

We use the following loss functions for the NNs.

Pinball. The Pinball loss is typically used as a loss function in quantile regression. It evaluates how well the predicted value⁵ match the actual quantile. For the quantile q , it is defined as

$$\text{Pinball}(q) = \frac{1}{N} \sum_{i=1}^N \max((y_i - \hat{y}_i) \cdot q, (y_i - \hat{y}_i) \cdot (q - 1)). \quad (20)$$

MAE. The Mean Absolute Error (MAE) is a loss function that measures the average absolute difference between predicted and actual values. It is defined as

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|. \quad (21)$$

MSE. The Mean Squared Error (MSE) is a loss function that measures the average squared difference between predicted and actual values. This loss function is sensitive to outliers and is defined as

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2. \quad (22)$$

Huber. The Huber loss is a loss function that combines characteristics of both MAE and MSE. It is less sensitive to outliers compared to MSE and is defined as

⁴Building 157 is omitted because its PV power generation is compared to the other buildings extremely high. This results in negative average daily total costs, which are not covered by the total average skill score difference. However, even for this outlier building, the DFR method reduces the average daily total costs compared to PFL.

⁵ \hat{y} refers in this term to a quantile prediction and not to a direct estimate of y .

$$\text{Huber} = \frac{1}{N} \sum_{i=1}^N \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2 & \text{if } |y_i - \hat{y}_i| \leq 1 \\ (|y_i - \hat{y}_i| - \frac{1}{2}) & \text{otherwise.} \end{cases} \quad (23)$$

5.4 Evaluation Metrics

The following describes the evaluation metrics used, namely the costs, the average skill score difference and the p -value of the Wilcoxon signed-rank test.

Costs. We use three different costs, namely the average daily total costs, the average daily imbalance costs, and the average daily DS costs⁶. These costs are the respective daily costs from Section 4.2 averaged over all days and buildings under consideration.

Further, we use the building-specific average daily total costs for the average skill score difference and the p -value of the Wilcoxon signed-rank test. These costs are the daily total costs averaged over all considered days for each building denoted by $\bar{C}_{\text{Total},i}$ for building i .

In the case of multiple runs, the costs are calculated as the mean of every evaluation run for the building.

Skill Score. We use the average difference between the skill score of our DFR method and the skill score of the PFL method. The skill score thereby corresponds to the percentage improvement of a method's building-specific average daily total costs relative to a reference and is defined as

$$\text{Skill Score}_i(\text{Method}) = \left(1 - \frac{\bar{C}_{\text{Total},i}^{\text{Method}}}{\text{Reference}}\right) * 100. \quad (24)$$

In the evaluation, we consider the building-specific average daily total costs resulting from the usage of the actual prosumption as a reference. These total costs consist only of dispatch schedule costs and are the minimal total costs for the respective prosumption. However, these costs are unrealistic to achieve as they require a perfect forecast. In case of multiple runs, it is calculated as the mean of the skill score of every evaluation run for the building.

The average skill score difference is then the difference between the skill score of the DFR method and the skill score of the PFL method averaged over all buildings,

Average Skill Score Difference

$$= \frac{1}{N} \sum_{i=1}^N (\text{Skill Score}_i(\text{DFR}) - \text{Skill Score}_i(\text{PFL})). \quad (25)$$

A positive average skill score difference indicates an improvement in the average daily total costs of our DFR method.

Wilcoxon Signed-Rank Test. The Wilcoxon signed-rank test is a paired difference test and is considered as non-parametric alternative to the paired t-test [28]. It is utilized to assess the differences in medians between two paired samples. In our case, the paired samples are the building-specific average daily total costs of DFR and PFL ($\bar{C}_{\text{Total},i}^{\text{DFR}}, \bar{C}_{\text{Total},i}^{\text{PFL}}, i = 1, \dots, N$). The test statistic is defined as

⁶In the following, we use these costs without its unit (€).

$$T = \min \left(\sum_{i:D_i > 0} \text{rank}(|D_i|), \sum_{i:D_i < 0} \text{rank}(|D_i|) \right) \quad (26)$$

with $D_i = \bar{C}_{\text{Total},i}^{\text{DFR}} - \bar{C}_{\text{Total},i}^{\text{PFL}}$, $i = 1, \dots, N$. We use the one-sided H_0 hypothesis

$$H_0 : \tilde{C}_{\text{Total}}^{\text{DFR}} \geq \tilde{C}_{\text{Total}}^{\text{PFL}} \quad (27)$$

with medians of the building-specific average daily total costs $\tilde{C}_{\text{Total}}^{\text{DFR}}$ and $\tilde{C}_{\text{Total}}^{\text{PFL}}$.

5.5 Decision-Focused Retraining for the Dispatchable Feeder

This subsection details each step in the application of the proposed DFR method applied on the dispatchable feeder optimization problem. Furthermore, we introduce the chosen hyperparameters and discuss the evaluation strategy applied.

Part 1: Generation of the Surrogate Neural Network for the Forecast Value

Step 1.1: Creation of the Surrogate Dataset. To generate the surrogate NN training dataset, we use a synthetic set of buildings with diverse shares of load regarding the installed PV power generation based on the first 50 buildings. Thereby, in addition to the original dataset, we scaled their PV power generation with the factors 5 and 10 and the load with factors 1/5, 1/2, 2, and 5, leading to a data augmentation, as in [26]. Further, we limit the battery to 13.5 kWh. In addition, we consider different forecast properties by generating forecasts with multiple models using different loss functions $\{Pinball_x \text{ for } x \in \{0.10, 0.25, 0.75, 0.90\}, MAE, MSE, Huber\}$. The Pinball Losses are selected to ensure that systematic over- and underestimation are reflected in the dataset. Further, the MAE, MSE, and Huber directly estimate the presumption. For every building of this dataset, the simulation of the optimization problem generating the forecast value builds the foundation for the dataset to train the surrogate NN. This is carried out over the first 50 buildings of the dataset, and the non-scaled buildings 51 to 100 are used to validate.

As exogenous variables, we use statistical properties of the buildings, such as minimum, maximum, mean, and standard deviation of the midday hour presumption. This should influence the surrogate to prioritize important properties of the forecast regarding the statistical values. In addition, we use the State of Energy (SoE) at the beginning of the schedule which is uniformly drawn inside the range of the battery storage. This should ensure that every achievable battery state is considered. These exogenous variables are scaled with a min-max scaler before being input into the network.

Since we found a distribution within total costs in which many values lie within the same value range, we carried out a quantile transformation to a uniform distribution to simplify the forecast for the surrogate. As the absolute value is not relevant for the prediction and only the ratio of the values is relevant, it should simplify the prediction of this relation and, therefore, strengthen the surrogate quality regarding the gradient.

Step 1.2: Training of the Forecast Value Surrogate Neural Network. The surrogate networks are trained as an ensemble of multiple networks with the dataset generated in Step 1.1. The surrogate

NN architecture is a multilayer perceptron. The network takes the forecast and the GT and encodes them in two separate layers with 64 neurons and Scaled Exponential Linear Unit (SELU) activation [11]. Further, these layers are concatenated with the scaled statistical input features and the scaled SoE. These layers are followed by layers with (64,64,64,32,8) neurons using SELU activation. The last layer with one neuron uses sigmoid activation to ensure that the value is in (0, 1). A detailed overview of the surrogate network architecture is given in Table 3 in the Appendix. In our case, to smoothen the surrogate function and make it less sensitive to poor generalization, we used an ensemble of five members to approximate the function f_{surr} , all trained using the Adam optimizer [10] ($\alpha = 0.001$), batch size 256, L_1 and L_2 Regularization. Further, we applied early stopping ($patience = 20$, $\delta = 0.0001$) to the surrogate NN and set the maximal epochs to 100. In training, we use the MAE between the prediction of the transformed forecast value and the GT transformed forecast value.

Part 2: Generation of a Forecasting Neural Network regarding the Forecast Value

Step 2.1: Training of the Forecasting Neural Network. We train a forecasting model per building of the retraining evaluation dataset. The aim of this forecasting model is to forecast the upcoming 42 hours of presumption. Therefore, the forecasting model takes as input features the last 168 historical values of the presumption. Further, it takes additional calendar information (weekday, hour, month) sin- and cos-encoded and the one-hot encoded public holidays and work days for the next 42 hours. These features are normalized to have a mean of zero and a standard deviation of one. The architecture is a Multilayer Perceptron containing four fully connected layers (128,64,32,42 neurons). The first three use the ReLU [1] activation function, and the last uses a linear activation function to be able to forecast positive and negative presumption values. The architecture of the used forecasting model is given in Table 4 in the Appendix. Further, we use a batch size of 32, the Adam optimizer ($\alpha = 0.0001$) and applied early stopping ($patience = 20$, $\delta = 0.0001$) in training. This early stopping ensures the selection of a well-converged model, according to the forecasting metric.

Step 2.2: Retraining of the Forecasting Neural Network regarding the Forecast Value. In retraining, we use the models given in Steps 1.2 and 2.1. We use the surrogate networks trained in Step 2 and average their results. The retraining again uses the Adam optimizer ($\alpha = 0.0001$) and a batch size of 32. Further, we use early stopping ($patience = 20$, $\delta = 0.0001$). The weights are updated every batch before calculating the loss and applying backpropagation. The dataset of this step consists of the features already used for training the prediction-focused forecast, extended by the statistical building information of the midday hour, presumption, and a simulated SoE at schedule begin. The building is simulated using the MAE and MSE for SoE estimation over the training period. These decisions account for the use of the dispatchable feeder under the assumption of a realistic and unbiased forecast. Further, the scaler of Step 1.1 is used.

Evaluation Strategy for DFR

To evaluate this approach, we compare two models. The first is the resulting model of Step 2.1. Therefore, this is a forecasting model trained with traditional prediction-focused learning. The second model we compare is the forecasting model after Step 2.2, therefore after DFR. For both of these models, we simulated the outcome of the optimization problem using this forecasting model for schedule calculation and SoE estimation (Section 4). Using the test period we evaluate the forecast value within a realistic daily operation of the optimization problem. The Evaluation of Step 2.1 and Step 2.2 is performed over three different runs with different initialization and the same surrogate networks of Step 1.1 for the runs regarding the two best-suited losses, the Pinball 0.70 and Pinball 0.75 Loss. We use as base loss the Pinball loss with the quantiles $\{0.40, 0.50, 0.60, 0.70, 0.75, 0.80, 0.90\}$ to sample the effect of systematically under-, or overestimating, the MSE, we omit the MAE as it is schematically the same as forecasting with pinball 0.50 loss. We chose the evaluated base losses regarding the observations regarding the original data presented in [26].

6 RESULTS

In this section, we present the results of the evaluation of the DFR method applied to the dispatchable feeder optimization problem. For this, we first consider the average skill score difference before comparing the mean average total daily costs. We then consider the mean average daily cost difference for each of the components of the total costs separately. Finally, we analyze the significance of our results.

Average Skill Score Difference. The average skill score difference across all buildings on the retraining dataset is shown in Figure 5, with the values reported in Table 2. We first observe that the proposed DFR method results in a positive average skill score difference, i.e. an improvement, for all base losses considered. Additionally, we observe that the size of this improvement depends on the base loss considered. The largest improvement is achieved when using Pinball 0.90 as a base loss, with an average skill score difference of over 130%, whilst the smallest improvement is with Pinball 0.75, where the DFR method only results in an improvement of 7.29%.

Average Daily Total Costs. We compare the average total daily costs when using PFL and our proposed DFR method in Figure 6 and additionally report these values in Table 2. These results also highlight that the DFR method outperforms PFL for all considered base losses, as the total costs from the dispatchable feeder optimization problem are always lower. Again, the degree to which these costs are lowered depends on the base loss considered.

Average Daily Cost Difference. As described in Section 4, the forecast value of the dispatchable feeder problem relates to a combination of imbalance and dispatch schedule costs. Therefore, to better analyze the performance of our proposed DFR method we visualize all components of these costs in Figure 7. We first observe that the imbalance costs are reduced by applying DFR for every base loss considered. Furthermore, the degree of this reduction depends on the base loss considered, with the largest reduction occurring for Pinball 0.90. Second, we observe that for the base

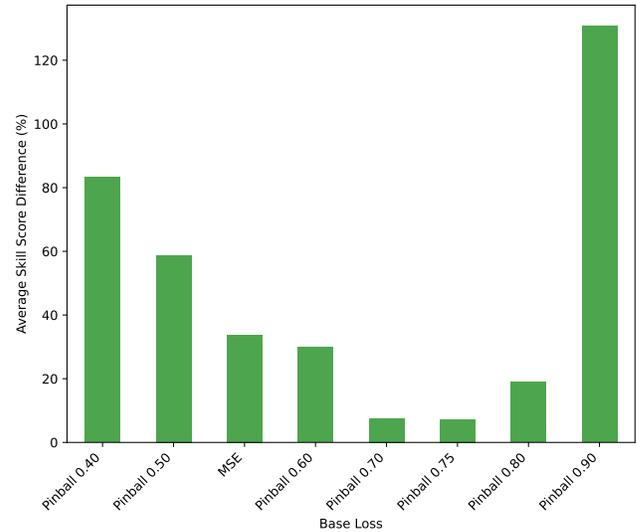


Figure 5: An overview of the average skill score difference in percent calculated across all buildings on the retraining dataset (see Figure 4). The improvements are larger for loss functions that are noticeably under- or overestimated.

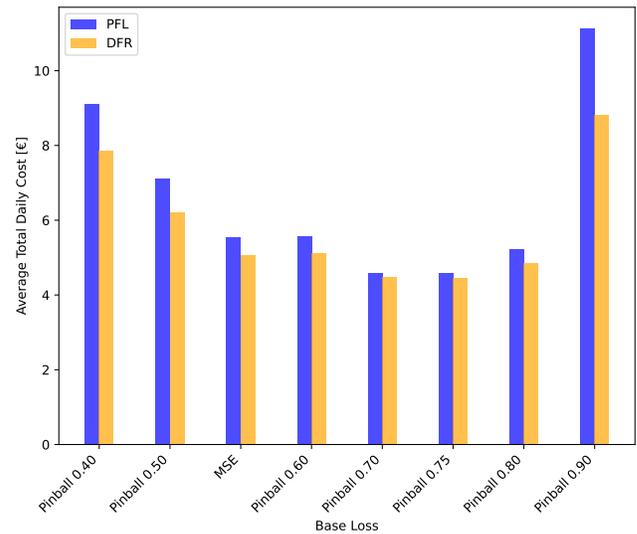


Figure 6: An overview of the average total daily costs for the different base losses using PFL (blue) and DFR (yellow) calculated on the test dataset (see Figure 4). The DFR method results in lower total costs for all considered losses.

loss functions Pinball 0.40, Pinball 0.50, MSE, Pinball 0.60 and Pinball 0.70 the mean dispatch schedule costs increase when applying DFR. Importantly, these increases are of a far smaller magnitude than the decrease in the imbalance costs, and therefore, the total costs are still noticeably reduced. Furthermore, we observe that for the Pinball loss of 0.75, 0.80, and 0.90, the mean dispatch schedule

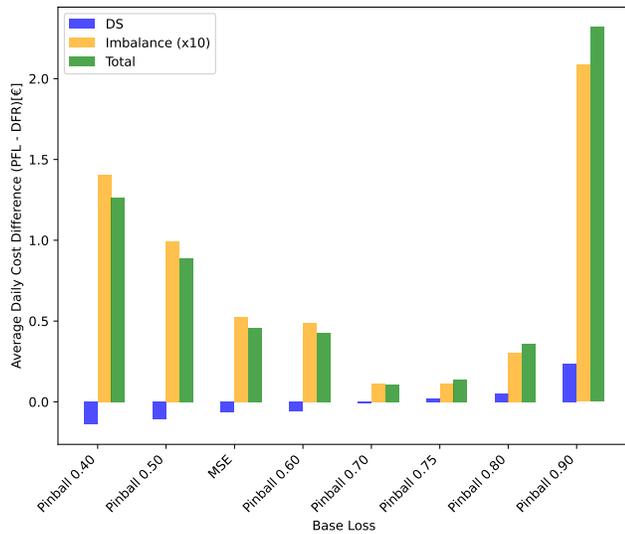


Figure 7: An overview of the average daily cost difference over the three different costs relevant for the dispatchable feeder, the dispatch schedule costs in blue, imbalance costs in yellow, and total costs in green. The costs are calculated for different base losses and the retraining as (PFL - DFR). Note that a positive value indicates an improvement in cost, while a negative indicates a deterioration. The numerical values are given in the Appendix (Table 5).

costs are also decreased by applying DFR. The DFR approach reduces imbalance and dispatch schedule costs for each of these loss functions.

Significance of Results. We report a summary of our results, including the P-Values for the applied Wilcoxon Signed-Rank Test in Table 2. This summary highlights that our proposed DFR approach results in reduced total costs and, as a result, a positive average skill score difference for all considered base losses. Furthermore, the reduction in costs is highly significant according to the Wilcoxon Signed-Rank Test for all base losses.

7 DISCUSSION

To discuss the results of our evaluation, we consider two main aspects. First, we discuss some key insights from our results. Second, we discuss the limitations and benefits of our proposed DFR method.

7.1 Key Insights

Given our results, we observe three major insights, which we discuss in more detail below.

DFR is Useful Regardless of the Base Loss. The results show that applying the proposed DFR method reduces total costs independent of the base loss. Importantly, the DFR results in reduced total costs even in cases where the selected base loss used in the prediction-focused model performs well, for example, Pinball 0.70. Therefore, we conclude that it always makes sense to apply DFR, even when the initial solution appears near-optimal. We also note that the

Table 2: An overview of the average daily costs for both the PFL and our proposed DFR approach and the P-Values reporting the significance of the Wilcoxon Signed-Rank Test and the average skill score difference for each considered loss function. The DFR method outperforms PFL for all considered loss functions.

	Average Daily C_{Total} PFL [€]	Average Daily C_{Total} DFR [€]	p -Value	Average Skill Score Diff. [%]
Pinball 0.40	9.12	7.85	1.08E-34	83.35
Pinball 0.50	7.11	6.22	1.86E-34	58.78
MSE	5.53	5.07	2.66E-33	33.81
Pinball 0.60	5.56	5.13	8.70E-31	29.88
Pinball 0.70	4.60	4.49	2.40E-14	7.53
Pinball 0.75	4.59	4.45	4.37E-02	7.29
Pinball 0.80	5.22	4.86	9.18E-09	18.96
Pinball 0.90	11.14	8.82	4.38E-34	130.77

degree of this improvement depends on the base loss considered, with the improvement generally smaller when the base loss already results in low total costs. However, this is expected since the closer the original result is to an optimal result, the less improvement possible.

Total Performance of the DFR is Still Dependent on the Base Loss. Whilst the DFR always results in a reduction in total costs compared to PFL, the total costs still depend on the initial base loss. For example, the largest improvement when using DFR is found when considering Pinball 0.90 as a base loss, but the resulting DFR total costs are still noticeably larger than the DFR total costs when other base losses. This result is not surprising since DFR can only correct a poor forecast via retraining, and this correction has boundaries. However, this highlights that the selection of the base loss is still important.

Applying DFR Offers Insights into Forecast Value for an Optimization Problem. By analyzing the separate components of the total cost difference for the considered optimization problem, namely the dispatch schedule costs and the imbalance costs, we are able to gain some insights into what kind of forecasts deliver value for the dispatchable feeder. For example, in a forecast that generally underestimates prosumption, e.g. Pinball 0.40, we observe a reduction in total cost, by decreasing imbalance cost and increasing dispatch schedule cost. This could be induced by an increase of the forecasted prosumption and, therefore, an increase in the dispatch schedule cost. This may result in a higher state of charge of the battery as more energy is consumed from the grid, and therefore, the battery can be better used in the scheduling task. This is observable until Pinball 0.70. If the overestimation is greater, the reverse effect is visible, as the battery has probably reached its upper limit.

Therefore, the DFR method may also be useful as a tool to help inform users as to the characteristics of forecasts that are valuable for certain optimization problems.

7.2 Limitations and Benefits

This section briefly discusses some of the key limitations and benefits of our proposed DFR method in the context of smart energy systems.

Limitations. First, we note that the proposed method is not a pure decision-focused learning method. It achieves optimal results for the optimization problem only when domain knowledge is used. But as we observe, when an unsuitable base loss is selected, DFR still contributes positively. A second limitation of our proposed method is that the surrogate NN is not constrained to parameterize a certain loss function, as in [4, 6, 20, 21]. As a result, the trained surrogate NN does not guarantee convexity. Therefore, the base loss is needed to ensure the forecast is in a feasible area. This makes the retraining process rather difficult and results in fluctuations within the training process. This leads to a small number of retraining epochs until the best tradeoff between the surrogate and the base loss is achieved, which is the current model selection criteria in the early stopping process.

Therefore, it would be interesting to investigate the effects of this surrogate NN in more detail by considering different architectures, different regularization techniques, different scenarios of learning rates and weightings, and investigating the impact of the parameterization of the loss function. Furthermore, our evaluation does not explicitly consider building-specific results, which are affected by outliers and building characteristics. In Future work, it would be interesting to analyze these effects in more detail.

Benefits. In addition to the positive results, our approach demonstrates several key benefits. Firstly, as a data-driven method, DFR can be applied to complex multiple-stage optimization problems, as we show through our evaluation of the dispatchable feeder optimization problem. Second, our proposed DFR method uses a global instance-independent surrogate NN. Independence from the instance of the optimization problem is advantageous, as this network can be trained once and used over many other instances afterward. This is specifically advantageous for energy systems, where multiple buildings with similar characteristics may all have independent optimization problems to solve. Furthermore, this independence implies that the computational effort to generate a surrogate dataset for training can be neglected, as the surrogate dataset generation only occurs once. However, the properties of the objects considered, for example, buildings, still play a role in surrogate training, and therefore, it would be interesting to investigate methods to increase this independence. Finally, the novel method is not specifically designed for a specific optimization problem and can, therefore, be applied to a range of optimization problems both within and outside the energy domain.

8 CONCLUSION

Many countermeasures to maintain stability in energy systems require that complex non-convex optimization problems are successfully solved. Often, these optimization problems require high-value forecasts as input, and many instances of the optimization problems must be solved. Therefore, the present paper introduces Decision-Focused Retraining (DFR), a retraining method combining decision-focused and prediction-focused learning advantages. Our method provides retraining through an instance-independent surrogate NN, leading to a scalable method that can deliver high-value forecasts to multiple instances of optimization problems. We evaluate our approach on the dispatchable feeder optimization problem,

showing that it results in an average reduction in average difference skill score of 7.29% across 199 buildings.

In light of these positive results, future work should consider the application of DFR. In addition to considering further optimization problems, future work should also investigate methods for improving the independence of the surrogate model.

ACKNOWLEDGMENTS

This project is funded by the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI, the Helmholtz Association under the Program "Energy System Design", and the German Research Foundation (DFG) as part of the Research Training Group 2153 "Energy Status Data: Informatics Methods for its Collection, Analysis and Exploitation". This work is supported by the Helmholtz Association Initiative and Networking Fund on the HAICORE@KIT partition.

REFERENCES

- [1] Abien Fred Agarap. 2018. Deep learning using rectified linear units (ReLU). *arXiv preprint arXiv:1803.08375* (2018).
- [2] Brandon Amos and J Zico Kolter. 2017. OptNet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*. PMLR, 136–145.
- [3] Riccardo Remo Appino, Jorge Ángel González Ordiano, Ralf Mikut, Timm Faulwasser, and Veit Hagenmeyer. 2018. On the use of probabilistic forecasts in scheduling of renewable energy sources coupled to storages. *Applied Energy* 210 (Jan. 2018), 1207–1218. <https://doi.org/10.1016/j.apenergy.2017.08.133>
- [4] Dishank Bansal, Ricky Chen, Mustafa Mukadam, and Brandon Amos. 2023. TaskMet: Task-Driven Metric Learning for Model Learning. (Dec. 2023).
- [5] Raymond H. Byrne, Tu A. Nguyen, David A. Copp, Babu R. Chalamala, and Imre Gyuk. 2018. Energy Management and Optimization Methods for Grid Energy Storage Systems. *IEEE Access* 6 (2018), 13231–13260. <https://doi.org/10.1109/ACCESS.2017.2741578>
- [6] Tsai-Hsuan Chung, Vahid Rostami, Hamsa Bastani, and Osbert Bastani. 2022. Decision-Aware Learning for Optimizing Health Supply Chains. *Extended Abstract presented at Machine Learning for Health (ML4H) Symposium* (Nov. 2022).
- [7] Mathieu David, John Boland, Luigi Cirocco, Philippe Lauret, and Cyril Voyant. 2021. Value of deterministic day-ahead forecasts of PV generation in PV+ Storage operation for the Australian electricity market. *Solar Energy* 224 (2021), 672–684.
- [8] Priya Donti, Brandon Amos, and J Zico Kolter. 2017. Task-based end-to-end model learning in stochastic optimization. *Advances in Neural Information Processing Systems* 30 (2017).
- [9] Adam N Elmachtoub and Paul Grigas. 2022. Smart "predict, then optimize". *Management Science* 68, 1 (2022), 9–26.
- [10] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [11] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-normalizing neural networks. *Advances in Neural Information Processing Systems* 30 (2017).
- [12] Lingkai Kong, Jiaming Cui, Yuchen Zhuang, Rui Feng, B Aditya Prakash, and Chao Zhang. 2022. End-to-end stochastic optimization with energy-based model. *Advances in Neural Information Processing Systems* 35 (2022), 11341–11354.
- [13] Lingkai Kong, Wenhao Mu, Jiaming Cui, Yuchen Zhuang, B Aditya Prakash, Bo Dai, and Chao Zhang. 2023. DF2: Distribution-Free Decision-Focused Learning. *arXiv preprint arXiv:2308.05889* (2023).
- [14] Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. 2023. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *arXiv preprint arXiv:2307.13565* (2023).
- [15] Maxime Mulamba, Jayanta Mandi, Michelangelo Diligenti, Michele Lombardi, Victor Bucarey, and Tias Guns. 2020. Contrastive losses and solution caching for predict-and-optimize. *arXiv preprint arXiv:2011.05354* (2020).
- [16] Dominik Putz, Michael Gumhalter, and Hans Auer. 2023. The true value of a forecast: Assessing the impact of accuracy on local energy communities. *Sustainable Energy, Grids and Networks* 33 (March 2023), 100983. <https://doi.org/10.1016/j.segan.2022.100983>
- [17] Elizabeth L. Ratnam, Steven R. Weller, Christopher M. Kellett, and Alan T. Murray. 2017. Residential load and rooftop PV generation: an Australian distribution network dataset. *International Journal of Sustainable Energy* 36, 8 (Sept. 2017), 787–806. <https://doi.org/10.1080/14786451.2015.1100196>

- [18] Riccardo Remo Appino, Jorge Angel Gonzalez Ordiano, Ralf Mikut, Veit Hagenmeyer, and Timm Faulwasser. 2018. Storage Scheduling with Stochastic Uncertainties: Feasibility and Cost of Imbalances. In *2018 Power Systems Computation Conference (PSCC)*. IEEE, Dublin, Ireland, 1–7. <https://doi.org/10.23919/PSCC.2018.8442529>
- [19] Linwei Sang, Yinliang Xu, Huan Long, Qinran Hu, and Hongbin Sun. 2022. Electricity Price Prediction for Energy Storage System Arbitrage: A Decision-Focused Approach. *IEEE Transactions on Smart Grid* 13, 4 (July 2022), 2822–2832. <https://doi.org/10.1109/TSG.2022.3166791>
- [20] Sanket Shah, Andrew Perrault, Bryan Wilder, and Milind Tambe. 2023. Leaving the Nest: Going Beyond Local Loss Functions for Predict-Then-Optimize. *arXiv preprint arXiv:2305.16830* (2023).
- [21] Sanket Shah, Kai Wang, Bryan Wilder, Andrew Perrault, and Milind Tambe. 2022. Decision-focused learning without decision-making: Learning locally optimized decision losses. *Advances in Neural Information Processing Systems* 35 (2022), 1320–1332.
- [22] Md. Nahid Haque Shazon, Nahid-Al-Masood, and Atik Jawad. 2022. Frequency control challenges and potential countermeasures in future low-inertia power systems: A review. *Energy Reports* 8 (Nov. 2022), 6191–6219. <https://doi.org/10.1016/j.egyrs.2022.04.063>
- [23] Helmut Simonis, Barry O’Sullivan, Deepak Mehta, Barry Hurley, and Milan De Cauwer. 2014. Energy-Cost Aware Scheduling/Forecasting Competition.
- [24] Fabrizio Sossan, Emil Namor, Rachid Cherkaoui, and Mario Paolone. 2016. Achieving the Dispatchability of Distribution Feeders Through Prosumers Data Driven Forecasting and Model Predictive Control of Electrochemical Storage. *IEEE Transactions on Sustainable Energy* 7, 4 (Oct. 2016), 1762–1777. <https://doi.org/10.1109/TSTE.2016.2600103>
- [25] Gokul Sidarth Thirunavukkarasu, Mehdi Seyedmahmoudian, Elmira Jamei, Ben Horan, Saad Mekhilef, and Alex Stojcevski. 2022. Role of optimization techniques in microgrid energy management systems—A review. *Energy Strategy Reviews* 43 (Sept. 2022), 100899. <https://doi.org/10.1016/j.esr.2022.100899>
- [26] Dorina Werling, Maximilian Beichter, Benedikt Heidrich, Kaleb Phipps, Ralf Mikut, and Veit Hagenmeyer. 2023. The Impact of Forecast Characteristics on the Forecast Value for the Dispatchable Feeder. In *Companion Proceedings of the 14th ACM International Conference on Future Energy Systems*. ACM, Orlando FL USA, 59–71. <https://doi.org/10.1145/3599733.3600251>
- [27] Dorina Werling, Maximilian Beichter, Benedikt Heidrich, Kaleb Phipps, Ralf Mikut, and Veit Hagenmeyer. 2024. Automating Value-Oriented Forecast Model Selection by Meta-learning: Application on a Dispatchable Feeder. In *Energy Informatics*, Bo Nørregaard Jørgensen, Luiz Carlos Pereira Da Silva, and Zheng Ma (Eds.). Vol. 14467. Springer Nature Switzerland, Cham, 95–116. https://doi.org/10.1007/978-3-031-48649-4_6 Series Title: Lecture Notes in Computer Science.
- [28] Frank Wilcoxon. 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin* 1, 6 (1945), 80–83. <https://doi.org/10.2307/3001968> Publisher: [International Biometric Society, Wiley].
- [29] Jialun Zhang, Yi Wang, and Gabriela Hug. 2022. Cost-oriented load forecasting. *Electric Power Systems Research* 205 (April 2022), 107723. <https://doi.org/10.1016/j.epsr.2021.107723>
- [30] Yufan Zhang, Mengshuo Jia, Honglin Wen, and Yuanyuan Shi. 2023. Value-oriented renewable energy forecasting for coordinated energy dispatch problems at two stages. *arXiv preprint arXiv:2309.00803* (2023).
- [31] Yufan Zhang, Honglin Wen, Yuexin Bian, and Yuanyuan Shi. 2023. Deriving Loss Function for Value-oriented Renewable Energy Forecasting. *arXiv preprint arXiv:2310.00571* (2023).
- [32] Arman Zharmagambetov, Brandon Amos, Aaron Ferber, Taoan Huang, Bistra Dilkina, and Yuandong Tian. 2023. Landscape surrogate: Learning decision losses for mathematical optimization under partial information. *arXiv preprint arXiv:2307.08964* (2023).

A APPENDICES

A.1 Architectures

Table 3: A detailed overview of the surrogate NN architecture. It is a multilayer perceptron consisting of multiple Fully Connected Layers (FCL).

Layer	Input Size	Output Size	Activation	Comments
Split x_0	42	42	-	Forecast
Split x_1	42	42	-	GT
Split x_2	5	5	-	SoE + Stats
FCL0 ₁	42 (x_0)	64	SELU	Forecast part of input
FCL0 ₂	42 (x_1)	64	SELU	GT part of input
Concatenate	133	133	-	Concatenation (FCL0 ₁ , FCL0 ₂ , x_2)
FCL1	133	64	SELU	-
FCL2	64	64	SELU	-
FCL3	64	64	SELU	-
FCL4	64	32	SELU	-
FCL5	32	8	SELU	-
FCL6	8	1	Sigmoid	Output layer between 0 and 1

Table 4: Detailed overview of the architecture of the NN for forecasting.

Layer	Input Size	Output Size	Activation
FCL1	504	128	ReLU
FCL2	128	64	ReLU
FCL3	64	32	ReLU
FCL4	32	42	Linear

A.2 Absolute Results of the Differences

Table 5: Results of the differences (PFL- DFR) of the three costs C_{DS} , C_{Imb} and C_{Total} . Denote imbalance costs are scaled by the imbalance factor. The results of this table represent the numerical values of the graphical representation in Figure 7.

	Average Daily C_{DS} [€] Diff.	Average Daily $C_{Imb}(x10)$ [€] Diff.	Average Daily C_{Total} [€] Diff.
Pinball 0.40	-0.140	1.404	1.264
Pinball 0.50	-0.108	0.996	0.888
MSE	-0.066	0.523	0.458
Pinball 0.60	-0.060	0.489	0.429
Pinball 0.70	-0.007	0.113	0.107
Pinball 0.75	0.023	0.115	0.138
Pinball 0.80	0.053	0.305	0.358
Pinball 0.90	0.234	2.085	2.320