



# Robust machine-learned algorithms for efficient grid operation

Nicolas Christianson<sup>1</sup>, Christopher Yeh<sup>1</sup>, Tongxin Li<sup>2</sup>, Mehdi Hosseini<sup>3</sup>, Mahdi Torabi Rad<sup>3</sup>, Azarang Golmohammadi<sup>3</sup> and Adam Wierman<sup>1</sup>

<sup>1</sup>Computing and Mathematical Sciences, California Institute of Technology, Pasadena, California, USA

<sup>2</sup>School of Data Science, The Chinese University of Hong Kong (Shenzhen), Shenzhen, Guangdong, China <sup>3</sup>Beyond Limits, Glendale, California, USA

Corresponding author: Nicolas Christianson; Email: nchristianson@caltech.edu

Received: 28 February 2023; Revised: 13 December 2023; Accepted: 09 August 2024

Keywords: energy systems; learning-augmented algorithms; machine learning robustness; online algorithms

#### Abstract

Increasing penetration of variable and intermittent renewable energy resources on the energy grid poses a challenge for reliable and efficient grid operation, necessitating the development of algorithms that are robust to this uncertainty. However, standard algorithms incorporating uncertainty for generation dispatch are computationally intractable when costs are nonconvex, and machine learning-based approaches lack worst-case guarantees on their performance. In this work, we propose a *learning-augmented* algorithm, ROBUSTML, that exploits the good average-case performance of a machine-learned algorithm for minimizing dispatch and ramping costs of dispatchable generation resources while providing provable worst-case guarantees on cost. We evaluate the algorithm on a realistic model of a combined cycle cogeneration plant, where it exhibits robustness to distribution shift while enabling improved efficiency as renewables penetration increases.

#### **Impact Statement**

We confront the challenge of designing efficient and robust algorithms for grid operation with the paradigm of learning-augmented algorithms, which leverage the excellent performance of machine learning while provably guaranteeing worst-case performance bounds and providing significant computational speedups over standard algorithms. Beyond the setting of generation dispatch, we propose that learning-augmented algorithms hold promise for safely utilizing the power of machine learning in a broader array of energy and sustainability-related control and optimization tasks.

# 1. Introduction

The need to reduce greenhouse gas emissions to mitigate the impacts of anthropogenic climate change is driving an energy transition characterized by large amounts of renewable generation resources being added

<sup>1 2</sup> This research article was awarded Open Data and Open Materials badges for transparent practices. See the Data Availability Statement for details.

<sup>©</sup> Beyond Limits, Inc. and the Author(s), 2025. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike licence (http://creativecommons.org/licenses/by-nc-sa/4.0), which permits noncommercial re-use, distribution, and reproduction in any medium, provided the same Creative Commons licence is used to distribute the re-used or adapted article and the original article is properly cited. The written permission of Cambridge University Press must be obtained prior to any commercial use.

to the grid. The variability of solar and wind energy and a relative scarcity of large-scale energy storage require the operation of dispatchable resources, such as fossil fuel generation, to balance out the fluctuations in renewable energy production and maintain reliable grid operation. However, conventional fossil fuel generators incur significant additional costs from the frequent cycling and ramping they must perform under high penetration of renewables, due both to decreased fuel efficiency and increased maintenance required from operating in this regime (Troy et al., 2010). Moreover, most dispatchable resources are limited in their ramp rate, and thus under high penetrations of renewables, such resources must be operated in a manner that anticipates system ramp needs, taking into account the high costs of frequent ramping while ensuring adequate supply to meet demand.

Operating generation optimally requires minimizing fuel costs while accounting for the intertemporal coupling of decisions, including both ramp costs and ramp limits. A natural approach for this problem is model predictive control (MPC), an algorithm that utilizes near-term forecasts of demand and other conditions to choose decisions that minimize aggregate cost over a fixed lookahead horizon (Morari & Lee, 1999). In addition to theoretical work confirming its good performance (e.g., Lin et al., 2021), MPC performs well in practice and has been deployed in a number of energy and sustainability-related domains, including control of wind turbines and solar photovoltaics (Lio et al., 2014; Sultana et al., 2017), smart buildings (Carli et al., 2020; Killian & Kozek, 2016), energy storage (Meng et al., 2015; Morstyn et al., 2018), and electric vehicle charging (Lee et al., 2018). Moreover, several regional power system operators in the US use MPC to settle the real-time electricity market (Ela & O'Malley, 2016), and it is widely understood that such lookahead algorithms will play an increasingly important role in enabling power systems to reliably absorb renewable energy volatility (Hua et al., 2019; Zhao et al., 2020).

However, MPC suffers computational complexity that grows exponentially with the lookahead horizon if the system model or costs are nonconvex, such as in the case of combined cycle generators (Bjelogrlic, 2000). In practice, it is thus typically necessary to use heuristic solvers that operate on a faster timescale but generally produce suboptimal decisions (Cao et al., 2017; Deng et al., 2015; Kelman et al., 2013). One promising avenue for overcoming the computational complexity of nonconvex MPC to enable improved performance in practice is the development of machine learning (ML) models that imitate its behavior, bypassing the need to solve an independent nonconvex optimization problem to generate each decision. This approach of "learning to control/optimize" has seen significant recent interest in the ML, control, and power systems communities (Fioretto et al., 2020; Kotary et al., 2021; Nellikkath & Chatzivasileiadis, 2022; Pan et al., 2022; Pon Kumar et al., 2018; Spielberg et al., 2017; Zhang et al., 2023). While several works develop approaches to guarantee constraint satisfaction for ML controllers (e.g., Zhang et al., 2023), none of these learning-based approaches come with a priori guarantees on their cost under distribution shift or on out-of-sample problem instances, jeopardizing their performance at deployment time. To counter this potential for poor performance and enable the reliable deployment of learned proxies for MPC in real-world settings, this work proposes an algorithm to robustify the behavior of such an ML proxy. We follow the paradigm of the emerging literature on learning-augmented algorithms (e.g., Lee et al., 2021; Lykouris and Vassilvtiskii, 2018; Purohit et al., 2018), specifically building upon the line of work (Antoniadis et al., 2020; Christianson, Handina, et al., 2022; Christianson et al., 2023; Li et al., 2022; Rutten et al., 2023) designing algorithms for online optimization with switching costs, a generalization of the dispatch problem, that can exploit the performance of an ML algorithm while providing worst-case guarantees on cost.

#### 1.1. Contributions

Our contributions are twofold. *First*, we propose a learning-augmented algorithm ROBUSTML for online optimization with switching costs that achieves the best deterministic performance bound in the setting of general nonconvex cost functions. Specifically, when provided with an ML algorithm for the problem as well as a heuristic baseline algorithm, for any desired  $\epsilon, \delta > 0$ , our algorithm achieves a cost at most  $(1 + \epsilon + \delta)$  times the cost incurred by the ML algorithm while maintaining a worst-case cost bound of  $\mathcal{O}(\frac{C}{\delta} + \frac{D}{\epsilon})$ , where C is the cost of the heuristic baseline and D is the diameter of the decision space. In other



**Figure 1.** Schematic of the cogeneration power plant model. The plant operator chooses how much electricity (yellow arrow) and steam (blue arrow) the three gas turbines (left cooling tower) produce, as well as how much steam is directed to the steam turbine (right cooling tower) to produce additional electricity. At each time t, ambient conditions (e.g., temperature) together with electricity and steam demand are represented by the vector  $\boldsymbol{\theta}_{1}$ , and electricity and steam dispatch decisions are represented by the vector  $\mathbf{x}_{1}$ .

words, ROBUSTML achieves performance that is close to that of ML, yet *robust* in comparison to the chosen baseline algorithm. This is the best-known tradeoff for deterministic algorithms, as all prior deterministic learning-augmented algorithms with bounded worst-case robustness paid at least three times the cost of the ML algorithm (Antoniadis et al., 2020).<sup>1</sup> *Second*, we empirically evaluate the performance of ROBUSTML on a realistic model of a combined cycle cogeneration power plant under increasing penetration of renewable energy. We find that an ML-based approach can improve computation time over MPC by three orders of magnitude, and while the performance of a pure ML approach suffers in response to distribution shift, our algorithm ROBUSTML takes advantage of the good performance of ML in the nominal case (improving over the baseline by 10–15%) while preserving robustness to distribution shift. Moreover, ROBUSTML can achieve better performance than both the pure ML approach and the baseline approach when there is increasing renewable penetration.

Our work has the potential for both direct and downstream impact on the problem of climate change. Our results suggest that using ROBUSTML for real-world grid operation could yield modest but tangible efficiency improvements, leading to reduced emissions. Moreover, ROBUSTML's robustness guarantees and lookahead use could enable greater penetration of renewables while maintaining grid reliability. More generally, we see great promise in using learning-augmented algorithms like ROBUSTML to achieve efficiency improvements without sacrificing robustness in other energy and sustainability-related domains where MPC is widely used (Carli et al., 2020; Killian & Kozek, 2016; Lee et al., 2018; Lio et al., 2014; Meng et al., 2015; Morstyn et al., 2018; Sultana et al., 2017).

# 2. Model and Preliminaries

We consider the problem of operating a combined cycle cogeneration power plant to meet both electricity and steam demand in the presence of exogenous variable renewable generation; see Figure 1 for a schematic. Specifically, we consider a plant with three gas turbines and a single steam turbine; we index the gas and steam turbines as  $\{1,2,3\}$  and  $\{4\}$ , respectively. At each time  $t \in \{1,...,T\}$ , the plant operator

<sup>&</sup>lt;sup>1</sup>Antoniadis et al. (2020) and Christianson et al. (2023) also consider *randomized* algorithms for online optimization with switching costs, obtaining bounds on expected algorithm performance. While these bounds can improve upon those of deterministic algorithms in the case of large D, we restrict our focus in this work to deterministic algorithms in order to obtain guarantees that hold with certainty, rather than in expectation.

observes an electricity demand (net of renewables)  $d_t \in \mathbb{R}_+$  and a steam demand  $q_t \in \mathbb{R}_+$ . In response, they choose energy dispatch setpoints  $p_t^{(i)} \in [\underline{p}_i, \overline{p}_i]$  and steam dispatch setpoints  $s_t^{(i)} \in [\underline{s}_i, \overline{s}_i]$  for all the turbines i = 1, ..., 4, where  $\underline{p}_i, \overline{p}_i$  are the lower and upper bounds, respectively, for the energy dispatch, and  $\underline{s}_i, \overline{s}_i$  are the lower and upper bounds, respectively, for the steam dispatch. Note that while the gas turbines i = 1, 2, 3 produce steam, and thus have positive steam dispatches, the steam turbine i = 4consumes steam, so  $s_t^{(4)} < 0$ . The plant operator's goal is to minimize the total cost of its dispatch decisions and its cost for ramping electricity generation while producing sufficient electricity and steam to meet demand. Formally, the plant operator faces the following constrained minimization problem:

$$\min_{\mathbf{p}_1,\ldots,\mathbf{p}_T \in [\underline{\mathbf{p}},\overline{\mathbf{p}}]} \sum_{t=1}^T f_t(\mathbf{p}_t,\mathbf{s}_t) + \alpha \|\mathbf{p}_t - \mathbf{p}_{t-1}\|_1$$

$$\mathbf{s}_1,\ldots,\mathbf{s}_T \in [\underline{\mathbf{s}},\overline{\mathbf{s}}]$$
(2.1)

s.t. 
$$\sum_{i=1}^{4} p_t^{(i)} = d_t$$
 for all  $t = 1, ..., T$  (2.2)

$$\sum_{i=1}^{4} s_t^{(i)} = q_t \quad \text{for all } t = 1, ..., T,$$
(2.3)

where  $\mathbf{p}_t = (p_t^{(i)})_{i=1}^4$ ,  $\mathbf{s}_t = (s_t^{(i)})_{i=1}^4$ ,  $f_t$  is a per-round fuel cost function that may depend on ambient conditions such as temperature, pressure, and humidity, and  $\alpha$  is a parameter determining the extent to which ramping energy generation is penalized. Importantly, the demands  $d_t$ ,  $q_t$  and the cost functions  $f_t$  are not all known in advance, so the plant operator cannot solve problem (2.1) all at once. Instead, the plant operator only knows the current timestep's problem parameters  $f_t$ ,  $d_t$ , and  $q_t$  exactly, though they may have (possibly noisy) access to predictions of these parameters in a short lookahead window.

The cogeneration dispatch problem we consider can be formulated as a specific instance of the more general problem of online optimization with switching costs (Chen et al., 2016). Abstractly, online optimization with switching costs can be considered as a game in which at each time  $t \in \{1, ..., T\}$ , a decision-maker receives a vector  $\theta_t \in \mathbb{R}^n$  parametrizing a cost function  $f(\cdot; \theta_t) : \mathbb{R}^d \to \mathbb{R}_+$ . In response, the decision-maker must choose a decision  $\mathbf{x}_t \in \mathbb{R}^d$  and incurs the *hitting* cost  $f(\mathbf{x}_t; \boldsymbol{\theta}_t)$  as well as the switching cost  $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|$  resulting from that decision, where  $\|\cdot\|$  is some norm. We assume that the decision  $\mathbf{x}_t$  does not impact future parameters  $\theta_{\tau}$  for  $\tau > t$ . In the context of the cogeneration dispatch application,  $\theta_t$  is a vector containing all ambient conditions such as temperature, pressure, humidity, and the electricity and steam demands  $d_t, q_t$  at time t,  $\mathbf{x}_t$  is a vector including the plant operator's energy and steam dispatch decisions at time t, and the function f maps ambient conditions and generator dispatches to a fuel cost while penalizing violation of constraints including capacity limits and supply-demand balance. The switching term  $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|$  represents the ramping cost.<sup>2</sup> The problem is *online*, meaning that the decision-maker only has access to the parameters  $\theta_1, \ldots, \theta_t$  that have been revealed by time t when making the decision  $\mathbf{x}_t$ . However, the decision-maker may have access to (possibly inaccurate) forecasts  $\hat{\theta}_{t+1|t}, \dots, \hat{\theta}_{t+w|t}$  of parameters within a lookahead window of length  $w \in \mathbb{N}$ , which can help them anticipate and reduce future switching costs. Such forecasts could be obtained using standard ML methods for predicting near-term weather or energy demand.

<sup>&</sup>lt;sup>2</sup> Note that while in (2.1) only ramping energy decisions yields a ramp cost, in this model all dispatch decisions in the vector  $\mathbf{x}_t$  are penalized by the ramp cost. However, since the norm  $\|\cdot\|$  can be arbitrary, we can model the cogeneration application by making the penalty for ramping the steam decisions arbitrarily small.

We consider two standard algorithms for the problem of online optimization with switching costs. The first, GREEDY, is a myopic algorithm that simply chooses the decision  $\mathbf{x}_t$  that minimizes  $f(\cdot; \boldsymbol{\theta}_t)$  at each time *t*. This algorithm has worst-case cost guarantees under mild assumptions on the structure of the cost function *f* (Zhang et al., 2021), and it resembles the single-stage dispatch algorithm widely used by power system operators. Its behavior is characterized formally as follows:

GREEDY: 
$$\boldsymbol{\theta}_t \mapsto \operatorname*{arg\,min}_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}; \boldsymbol{\theta}_t) =: \mathbf{x}_t.$$

That is, GREEDY can be viewed as a function that, when provided with parameter vector  $\theta_t \in \mathbb{R}^n$ , returns the minimizer of  $f(\cdot; \theta_t)$  as a dispatch decision. Note that, since we assume that  $f(\cdot; \theta_t)$  penalizes all constraints from the original cogeneration problem formulation, GREEDY will necessarily satisfy these constraints.

The second algorithm we consider is model predictive control (MPC). MPC solves a lookahead optimization problem using near-term predictions of parameters to choose a decision. Formally, at time *t*, given the perfectly known current parameter vector  $\theta_t$  and forecasts  $\hat{\theta}_{t+1|t}, ..., \hat{\theta}_{t+w|t}$  of parameters over the next *w* timesteps, MPC chooses its decision as the optimal solution of the following optimization problem:

$$MPC: \Theta \mapsto \underset{\mathbf{y}_{1}, \dots, \mathbf{y}_{w} \in \mathbb{R}^{d}}{\arg\min} f(\mathbf{x}; \boldsymbol{\theta}_{t}) + \sum_{\tau=1}^{w} f\left(\mathbf{y}_{\tau}; \widehat{\boldsymbol{\theta}}_{t+\tau|t}\right) + \|\mathbf{y}_{\tau} - \mathbf{y}_{\tau-1}\| =: \mathbf{x}_{t}$$
(2.4)

where  $\mathbf{y}_0 := \mathbf{x}$  and we have concatenated the parameter vectors  $(\boldsymbol{\theta}_t, \widehat{\boldsymbol{\theta}}_{t+1|t}, \dots, \widehat{\boldsymbol{\theta}}_{t+w|t})$  into a single entity  $\Theta \in (\mathbb{R}^n)^{w+1}$  for brevity. Note that only the minimizer  $\mathbf{x}$  corresponding to the decision made for time t is binding, i.e., the chosen decision  $\mathbf{x}_t$  is simply the optimal  $\mathbf{x}$  in the above optimization; all of the other variables  $\mathbf{y}_1, \dots, \mathbf{y}_w$  are ignored after the solution is obtained. Note that, similar to GREEDY, MPC produces decisions that satisfy the constraints of the original cogeneration problem.

As previously discussed, MPC can be computationally prohibitive if  $f(\cdot; \theta)$  is nonconvex. However, there has been a great deal of recent work on learning proxies for power systems-related constrained optimization problems that operate more quickly than traditional optimization solvers (Donti et al., 2020; Park & Van Hentenryck, 2023; Zhang et al., 2023). Following this approach, in our work, we train an ML model to approximate the input–output behavior of MPC in an unsupervised fashion. That is, given a dataset  $\mathcal{D} = \{\Theta_i\}_{i=1}^N$  of cost function parameters, we train a neural network ML :  $(\mathbb{R}^n)^{w+1} \rightarrow (\mathbb{R}^d)^{w+1}$  to minimize MPC's lookahead objective (2.4) on average over this dataset. We employ the method of Zhang et al. (2023) and include a gauge map in the output layer of the network to ensure that the network always produces decisions that satisfy the capacity limits and supply–demand balance constraints for both energy (2.2) and steam (2.3).

While this unsupervised training approach may yield low empirical error on the training set and the gauge map ensures constraint satisfaction, this does not guarantee that the ML algorithm will perform well on out-of-sample instances or under distribution shift. In the next section, we introduce an algorithm to address this challenge and provide provable cost guarantees for the learned proxy ML.

We conclude this section with some notation. For an algorithm ALG producing decisions  $\mathbf{x}_1, ..., \mathbf{x}_T$ , define  $A_{LG_t} := \mathbf{x}_t$  as ALG's decision at time *t*, and define  $C_{ALG}(s,t) := \sum_{\tau=s}^t f(\mathbf{x}_{\tau}; \boldsymbol{\theta}_{\tau}) + \|\mathbf{x}_{\tau} - \mathbf{x}_{\tau-1}\|$  as ALG's cost from time *s* through *t*. For brevity, we write the total cost as  $C_{ALG} := C_{ALG}(1,T)$ .

# 3. Algorithm

We propose in Algorithm 1 a novel algorithm, ROBUSTML, that "robustifies" the algorithm ML, giving a sequence of dispatch decisions with cost that can be bounded in terms of the costs incurred by both ML and GREEDY. ROBUSTML behaves as follows: it begins by following ML's decisions, but if ML surpasses a cost threshold and GREEDY is performing relatively well, ROBUSTML will switch to following GREEDY's decisions (line 8). However, if GREEDY begins to perform worse relative to ML, then ROBUSTML will

switch back to following ML (line 15). The specific thresholds for switching are determined by the parameters  $\epsilon, \delta > 0$ , which reflect the algorithm's confidence in ML. That is, when  $\epsilon$  and  $\delta$  are small (close to 0), ROBUSTML will spend more time following the decisions of ML; on the other hand, when  $\epsilon$ and  $\delta$  are large, ROBUSTML will spend more of its time following the decisions of GREEDY. Tuning of  $\epsilon$ and  $\delta$  thus enables the plant operator to choose the particular tradeoff they wish to achieve between exploiting the (typically excellent) performance of ML and obtaining worst-case performance guarantees comparable to that of GREEDY, even in cases of distribution shift.

Algorithm 1: The algorithm ROBUSTML( $\epsilon, \delta$ ). Note that all algorithms are assumed to begin in the same initial state  $\mathbf{x}_0$ , so  $\mathbf{a}_0 = \mathbf{r}_0 = \mathbf{x}_0$  (where  $\mathbf{a}_0 = ML_0$  and  $\mathbf{r}_0 = GREEDY_0$ ).

**Input:** Algorithms ML, GREEDY; hyperparameters  $\epsilon, \delta > 0$ , space diameter D **Output:** Decisions  $\mathbf{x}_1, \ldots, \mathbf{x}_T$  chosen online  $1 s \leftarrow 1$ **2**  $\mathbf{x}_1 \leftarrow \mathbf{a}_1 \coloneqq ML_1$ **3** for  $t = 2, 3, \ldots, T$  do Observe  $f_t$ ,  $\mathbf{a}_t := ML_t$ , and  $\mathbf{r}_t := GREEDY_t$ 4 // Case where the algorithm coincides with  $ML_{t-1}$ if  $\mathbf{x}_{t-1} = \mathbf{a}_{t-1}$  then 5 if  $C_{ML}(s,t) \geq \frac{2D}{\epsilon}$  and  $C_{GREEDY}(1,t) < \delta \cdot C_{ML}(1,t)$  then 6  $s \leftarrow t + 1$ 7  $\mathbf{x}_t \leftarrow \mathbf{r}_t$ 8 else 9  $\mathbf{x}_t \leftarrow \mathbf{a}_t$ 10 else // Case where the algorithm coincides with  $GREEDY_{t-1}$ 11 if  $C_{G_{REEDY}}(1,t) < \delta \cdot C_{ML}(1,t)$  then 12  $\mathbf{x}_t \leftarrow \mathbf{r}_t$ 13 14 else  $\mathbf{x}_t \leftarrow \mathbf{a}_t$ 15 16 end

Our main analytic result on the performance of ROBUSTML is the following bound on its cost. Note that we assume that the decision space has diameter D, so  $||ML_t - GREEDY_t|| \le D$  for all  $t \in \{1, ..., T\}$ .

Theorem 3.1. The algorithm ROBUSTML (Algorithm 1) achieves cost bounded as

$$C_{ROBUSTML} \le \min\left\{ (1+\epsilon+\delta)C_{ML}, \left(1+\frac{1+\epsilon}{\delta}\right)C_{GREEDY} + \left(1+\frac{2}{\epsilon}\right)D\right\}.$$

We present a proof of Theorem 3.1 in Section 6. In particular, the theorem tells us that by selecting  $\epsilon, \delta$  arbitrarily small, ROBUSTML can achieve performance arbitrarily close to that achieved by ML, at the cost of possibly worse performance relative to GREEDY. However, by selecting moderate  $\epsilon, \delta$ , it is possible to trade off the exploitation of ML with robustness with respect to GREEDY's performance. Moreover, rather than selecting fixed  $\epsilon, \delta$  a priori to achieve a particular, fixed tradeoff, it is possible to apply techniques from online learning to optimally tune these parameters in an adaptive fashion (see, e.g., Khodak et al., 2022; Li et al., 2022; Zeynali et al., 2021).

It is also important to note that the above result does not depend in any crucial way on the choice of the algorithm GREEDY. That is, the plant operator can choose *any* strategy as their baseline "robust" algorithm, including heuristic methods that are currently employed in practice, and the performance bound given by Theorem 3.1 on Algorithm 1 will hold, with  $C_{GREEDY}$  replaced by the cost of the operator's chosen strategy. Thus, Algorithm 1 gives a method for power plant operators to take advantage of



Figure 2. Number of seconds (mean and standard deviation) for each algorithm to produce a day's worth of dispatch decisions.



**Figure 3.** Cost of each algorithm under increasing noise  $\sigma$  on the lookahead predictions, normalized by *GREEDY's* cost. Curves indicate mean cost and shaded regions cover  $\pm$  one standard deviation. Several choices of parameters are shown for ROBUSTML:  $\epsilon = \delta = 0.1$  (top left),  $\epsilon = \delta = 0.4$  (top right),  $\epsilon = \delta = 0.7$  (bottom left),  $\epsilon = \delta = 1.0$  (bottom right).

machine learning's good performance while maintaining worst-case guarantees relative to their existing strategies, whatever they may be.

# 4. Experimental Results and Discussion

We investigate the performance of our algorithm for cogeneration dispatch on a modified form of the CogenEnv environment from SustainGym, a reinforcement learning benchmarks suite (Yeh et al., 2023); all of the code, models, and data employed in conducting experiments can be found at https://doi.org/10.5281/zenodo.13623809. This cogeneration plant has behavior modeled in a black-box fashion via neural networks, and thus the fuel cost is differentiable as a function of the dispatch decisions. We train an ML model to replicate the behavior of MPC with lookahead w = 6 in an unsupervised fashion as described in the previous section. The model is trained on a dataset of 100 days of ambient conditions (temperature, pressure, humidity) and municipal demands for energy and steam on a 15 minute basis. To solve for the GREEDY and MPC decisions, we use sequential least-squares programming in SciPy (Virtanen et al., 2020).

We begin by comparing the runtimes of GREEDY, MPC, and ML to produce a day's worth of dispatch decisions on an AWS EC2 instance with 4 virtual CPU cores and an NVIDIA T4 GPU (Figure 2). We find that while GREEDY can produce a day of dispatches in about 30 seconds, MPC with lookahead w = 6 takes



**Figure 4.** Cost of each algorithm under increasing wind penetration, normalized by GREEDY's cost. Curves indicate mean cost and shaded regions cover  $\pm$  one standard deviation. Several choices of parameters are shown for ROBUSTML:  $\epsilon = \delta = 0.1$  (top left),  $\epsilon = \delta = 0.4$  (top right),  $\epsilon = \delta = 0.7$  (bottom left),  $\epsilon = \delta = 1.0$  (bottom right).

nearly 100 times as long (~45 minutes). On the other hand, ML is three orders of magnitude faster than MPC, returning a day's worth of decisions in under two seconds on average. This is unsurprising, since while GREEDY and MPC require solving optimization problems to produce dispatch decisions, ML simply requires a neural network's forward pass.

We next examine the performance of ML, GREEDY, and ROBUSTML on the baseline system (no renewables) when the lookahead predictions are unreliable, as in the case when there is distribution shift. That is, we compare the setting of perfect predictions (i.e.,  $\hat{\theta}_{t+1|t} = \theta_{t+1}, ..., \hat{\theta}_{t+w|t} = \theta_{t+w}$ ) to settings with i.i.d. Gaussian noise and increasing standard deviation  $\sigma$  on the predictions (i.e.,  $\hat{\theta}_{t+1|t} = \theta_{t+1} + \mathbf{z}_{t+1|t}, ..., \hat{\theta}_{t+w|t} = \theta_{t+w} + \mathbf{z}_{t+w|t}$ , with  $\mathbf{z}_{t|t} \sim \mathcal{N}(\mathbf{0}, \sigma \mathbf{I})$ ). We show the results in Figure 3 for several choices of the ROBUSTML parameters  $\epsilon, \delta$ . In particular, we observe that while ML performs better than GREEDY when predictions are good ( $\sigma \leq 30$ ), its performance degrades significantly as the noise grows ( $\sigma \rightarrow 100$ ). Nonetheless, for the cases of larger  $\epsilon$  and  $\delta$  – specifically,  $\epsilon = \delta = 0.7$  and  $\epsilon = \delta = 1.0$  – ROBUSTML gracefully transitions between the good performance of ML for small  $\sigma$  to nearly matching or beating the performance of GREEDY in the large  $\sigma$  regime. Thus, even though the quality of predictions is unknown a priori, ROBUSTML exploits the good performance of ML when predictions are good while preserving robustness in the worst case. Notably, however, this is not the case when  $\epsilon = \delta$  is small, as can be seen in the top row of Figure 3; in this case, ROBUSTML stays too close to the ML algorithm, and its performance suffers as a result.

We further examine the performance of the algorithms under increasing penetration of wind energy, displaying the results in Figure 4. We find that the efficiency improvement of ML over GREEDY widens for moderate wind penetration (up to 200 MW), highlighting the value of using lookahead alongside ML to increase efficiency when renewable generation leads to more frequent ramp events. Notably, however, ML's performance gets closer to that of GREEDY when wind penetration nears 400 MW, possibly reflecting the challenge of adequately anticipating future ramp needs in a high-renewables environment even when equipped with moderate lookahead. Nonetheless, we find that when  $\epsilon = \delta = 1.0$ , ROBUSTML yields uniformly improved performance over both ML and GREEDY and thus enables better efficiency even in challenging high-renewables scenarios. While surprising, this improvement highlights that adaptive switching between the ML and GREEDY algorithms enables ROBUSTML to exploit each algorithm when it performs well, leading to better average performance than either algorithm.

#### 5. Conclusions

In this work, we proposed a learning-augmented algorithm, ROBUSTML, for the operation of cogeneration plants with ramp costs, finding that it effectively balances exploitation of the good performance of machine-learned algorithms with worst-case robustness in cases of distribution shift. In particular, ROBUSTML allows for taking advantage of lookahead predictions without suffering the computational complexity of MPC, and depending on how its parameters are tuned, it can yield performance that uniformly improves on both a pure ML approach and existing baseline algorithms used in practice.

Several interesting avenues remain open for future work. First, it would be interesting to apply ROBUSTML to larger energy systems with more generators to examine how performance gains scale on larger systems. Second, while ROBUSTML currently accounts for ramping costs in its performance guarantee, it does not ensure the satisfaction of strict ramp limits, which constrain the operation of many dispatchable generators in the real world. As such, extending the algorithm to allow for such strict limits would be a valuable step toward real-world algorithm deployment. Finally, while in this work, we trained the ML model to mimic MPC, there are numerous variants of MPC that are explicitly designed for greater robustness, e.g., Bemporad and Morari (1999), Christianson, Werner, et al. (2022), Langson et al., (2004), and Schwarm and Nikolaou (1999). It would be interesting to examine how such robustified MPC variants perform when input to a learning-augmented algorithm such as ROBUSTML and whether they can improve both average and worst-case performance.

#### 6. Proof of Theorem 3.1

We begin by showing  $C_{\text{ROBUSTML}} \leq (1 + \epsilon + \delta)C_{\text{ML}}$ . Note that the algorithm consists of *phases* in which ROBUSTML first coincides with ML, and then switches to following GREEDY, before switching back to ML, and so on. We will assume that ROBUSTML ends the instance coinciding with ML, so  $\mathbf{x}_T = \mathbf{a}_T$ ; the case in which ROBUSTML ends at  $\mathbf{r}_T$  is similar. Let  $t_i$  denote the timestep in which ROBUSTML switches from GREEDY back to ML for the *i*th time, with  $t_0 := 1$  since ROBUSTML always begins by following ML. Similarly, let  $m_i$  denote the timestep in which ROBUSTML switches from ML to GREEDY for the *i*th time. Clearly, we have  $1 = t_0 < m_1 < t_1 < \cdots < m_k < t_k \leq T$ , for some  $k \in \mathbb{N}$ . Even though ROBUSTML ends at ML, define  $m_{k+1} := T + 1$  for notational simplicity. Then the cost of ROBUSTML may be written as

$$C_{\text{ROBUSTML}} = \sum_{\tau=1}^{m_{1}-1} f_{\tau}(\mathbf{a}_{\tau}) + \|\mathbf{a}_{\tau} - \mathbf{a}_{\tau-1}\| + \sum_{i=1}^{k} \left( f_{m_{i}}(\mathbf{r}_{m_{i}}) + \|\mathbf{r}_{m_{i}} - \mathbf{a}_{m_{i}-1}\| + \sum_{\tau=m_{i}+1}^{t_{i}-1} f_{\tau}(\mathbf{r}_{\tau}) + \|\mathbf{r}_{\tau} - \mathbf{r}_{\tau-1}\| + f_{t_{i}}(\mathbf{a}_{t_{i}}) + \|\mathbf{a}_{t_{i}} - \mathbf{r}_{t_{i}-1}\| + \sum_{\tau=t_{i}+1}^{m_{i+1}-1} f_{\tau}(\mathbf{a}_{\tau}) + \|\mathbf{a}_{\tau} - \mathbf{a}_{\tau-1}\| \right) \leq C_{\text{ML}}(1, m_{1} - 1) + \sum_{i=1}^{k} (C_{\text{GREEDY}}(m_{i}, t_{i} - 1) + \|\mathbf{r}_{m_{i}-1} - \mathbf{a}_{m_{i}-1}\| + C_{\text{ML}}(t_{i}, m_{i+1} - 1) + \|\mathbf{a}_{t_{i}-1} - \mathbf{r}_{t_{i}-1}\|)$$

$$(6.1)$$

$$\leq C_{\rm ML}(1, m_1 - 1) + 2kD + \sum_{i=1}^{k} C_{\rm Greedy}(m_i, t_i - 1) + C_{\rm ML}(t_i, m_{i+1} - 1)$$
(6.2)

$$\leq (1+\epsilon)C_{\rm ML} + \sum_{i=1}^{k} C_{\rm Greedy}(m_i, t_i - 1)$$
(6.3)

$$\leq (1 + \epsilon + \delta) C_{\rm ML} \tag{6.4}$$

where (6.1) follows from the triangle equality on  $\|\mathbf{r}_{m_i} - \mathbf{a}_{m_i-1}\|$  and  $\|\mathbf{a}_{t_i} - \mathbf{r}_{t_i-1}\|$ , and (6.2) follows by the diameter bound. The inequality (6.3) follows by line 6 of the algorithm, which states that the algorithm will switch from following ML to following GREEDY at time *t* only if  $C_{ML}(s,t) \ge \frac{2D}{\epsilon}$ . Noting that at the start of any timestep *t*, *s* is exactly

$$s = \max_{i:m_i+1 \le t} m_i + 1$$

(with  $m_0 := 0$  for notational convenience), it follows that for each  $i \in [k]$ ,  $C_{ML}(m_{i-1}+1,m_i) \ge \frac{2D}{\epsilon}$ . Thus

$$2kD \leq \epsilon \sum_{i=1}^{k} C_{\mathrm{ML}}(m_{i-1}+1,m_i) = \epsilon \cdot C_{\mathrm{ML}}(1,m_k) \leq \epsilon \cdot C_{\mathrm{ML}}.$$

Finally, (6.4) follows from

$$\sum_{i=1}^{k} C_{\text{Greedy}}(m_i, t_i - 1) \leq C_{\text{Greedy}}(1, t_k - 1) < \delta \cdot C_{\text{ML}}(1, t_k - 1) \leq \delta \cdot C_{\text{ML}},$$

since by definition,  $\mathbf{x}_{t_k-1} = \mathbf{r}_{t_k-1}$ , which by line 12 of the algorithm means that  $C_{\text{GREEDY}}(1, t_k - 1) < \delta \cdot C_{\text{ML}}(1, t_k - 1)$ . Thus, we have proved the desired bound  $C_{\text{ROBUSTML}} \le (1 + \epsilon + \delta)C_{\text{ML}}$ .

We now turn to showing  $C_{\text{ROBUSTML}} \leq (1 + \frac{1+\epsilon}{\delta})C_{\text{GREEDY}} + (1 + \frac{2}{\epsilon})D$ . First suppose ROBUSTML finishes the instance coinciding with ML, so  $\mathbf{x}_T = \mathbf{a}_T$ . Let  $\tau \in \{0, ..., T-1\}$  denote the last time at which ROBUSTML coincided with GREEDY, or that  $\mathbf{x}_\tau = \mathbf{r}_\tau$ . Thus the cost can be bounded as

$$C_{\text{ROBUSTML}} = C_{\text{ROBUSTML}}(1, \tau + 1) + C_{\text{ROBUSTML}}(\tau + 2, T)$$
  

$$\leq (1 + \epsilon + \delta)C_{\text{ML}}(1, \tau + 1) + C_{\text{ML}}(\tau + 2, T)$$
(6.5)

$$\leq \max\left\{\left(1+\frac{1+\epsilon}{\delta}\right)C_{\text{Greedy}}(1,\tau+1)+\frac{2D}{\epsilon},\left(1+\frac{1+\epsilon}{\delta}\right)C_{\text{Greedy}}\right\}$$
(6.6)

$$\leq \left(1 + \frac{1+\epsilon}{\delta}\right) C_{\text{GREEDY}} + \frac{2D}{\epsilon}$$
(6.7)

where (6.5) follows via the previously proved inequality  $C_{\text{ROBUSTML}} \leq (1 + \epsilon + \delta)C_{\text{ML}}$ , and (6.6) follows by the fact (according to line 14 of the algorithm) that ROBUSTML switching from GREEDY to ML at time  $\tau + 1$ means that  $C_{\text{GREEDY}} \geq \delta \cdot C_{\text{ML}}(1, \tau + 1)$ , as well as from the following observation: since ROBUSTML coincides with ML between times  $\tau + 1$  and *T*, line 6 of the algorithm tells us that either  $C_{\text{ML}}(\tau + 2, T) \leq \frac{2D}{\epsilon}$  or  $C_{\text{GREEDY}} \geq \delta \cdot C_{\text{ML}}$ .

Finally, suppose ROBUSTML finishes the instance coinciding with GREEDY, so  $\mathbf{x}_T = \mathbf{r}_T$ . Let  $\sigma \in \{0, ..., T-1\}$  denote the last time at which ROBUSTML coincided with ML, or that  $\mathbf{x}_{\sigma} = \mathbf{a}_{\sigma}$ . By the previous case's inequality (6.7), we have

$$\begin{split} \mathbf{C}_{\text{ROBUSTML}} &= \mathbf{C}_{\text{ROBUSTML}}(1, \sigma) + \mathbf{C}_{\text{ROBUSTML}}(\sigma + 1, T) \\ &\leq \left(1 + \frac{1 + \epsilon}{\delta}\right) \mathbf{C}_{\text{GREEDY}}(1, \sigma) + \frac{2D}{\epsilon} + f_{\sigma+1}(\mathbf{r}_{\sigma+1}) + \|\mathbf{r}_{\sigma+1} - \mathbf{a}_{\sigma}\| + \mathbf{C}_{\text{GREEDY}}(\sigma + 2, T) \\ &\leq \left(1 + \frac{1 + \epsilon}{\delta}\right) \mathbf{C}_{\text{GREEDY}}(1, \sigma) + \frac{2D}{\epsilon} + D + \mathbf{C}_{\text{GREEDY}}(\sigma + 1, T) \\ &\leq \left(1 + \frac{1 + \epsilon}{\delta}\right) \mathbf{C}_{\text{GREEDY}} + \left(1 + \frac{2}{\epsilon}\right) D. \end{split}$$

Acknowledgments. The authors thank James Chen for helpful discussions.

Author contribution. Conceptualization: all authors; Methodology: N.C., C.Y.; Data curation: N.C., M.H., M.T.R., A.G.; Formal analysis: N.C.; Data visualization: N.C.; Writing—original draft: N.C.; Writing—review and editing: all authors.

**Competing interest.** Competing interests: during the completion of this work, M.H., M.T.R., and A.G. are or were employed at Beyond Limits, which funded a portion of this work.

Data availability statement. The code, models, and data used in conducting the experiments presented in this work are publicly available at https://doi.org/10.5281/zenodo.13623809.

Ethical standard. The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

**Funding statement.** This research was supported by NSF grants CNS-2146814, CPS-2136197, CNS-2106403, and NGSDI-2105648, Amazon AWS, the Caltech Resnick Sustainability Institute, and Beyond Limits. N.C. was supported by an NSF Graduate Research Fellowship (DGE-1745301). Tongxin Li was supported by the start-up funding UDF01002773 of CUHK-Shenzhen. The NSF, Amazon AWS, and the Caltech Resnick Sustainability Institute had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. Beyond Limits had a role in the study design, data collection, and preparation of the final manuscript.

# References

- Antoniadis, A., Coester, C., Elias, M., Polak, A., & Simon, B. (2020). Online metric algorithms with untrusted predictions [ISSN: 2640-3498]. In *Proceedings of the 37th International Conference on Machine Learning*, 345–355. Retrieved September 21, 2022, from https://proceedings.mlr.press/v119/antoniadis20a.html
- Bemporad, A., & Morari, M. (1999). Robust model predictive control: A survey. In A. Garulli & A. Tesi (Eds.), Robustness in identification and control (pp. 207–226). Springer London. https://doi.org/10.1007/BFb0109870
- Bjelogrlic, M. (2000). Inclusion of combined cycle plants into optimal resource scheduling. In 2000 Power Engineering Society Summer Meeting (Cat. No.00CH37134), 1, 189–194 vol. 1. https://doi.org/10.1109/PESS.2000.867469
- Cao, G., Lai, E. M.-K., & Alam, F. (2017). Gaussian process model predictive control of unknown non-linear systems. *IET Control Theory & Applications*, 11(5), 703–713. https://doi.org/10.1049/iet-cta.2016.1061
- Carli, R., Cavone, G., Ben Othman, S., & Dotoli, M. (2020). IoT Based Architecture for Model Predictive Control of HVAC Systems in Smart Buildings [Number: 3 Publisher: Multidisciplinary Digital Publishing Institute]. Sensors, 20(3), 781. https:// doi.org/10.3390/s20030781
- Chen, N., Comden, J., Liu, Z., Gandhi, A., & Wierman, A. (2016). Using Predictions in Online Optimization: Looking Forward with an Eye on the Past. In Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science, 193–206. https://doi.org/10.1145/2896377.2901464
- Christianson, N., Handina, T., & Wierman, A. (2022). Chasing Convex Bodies and Functions with Black-Box Advice [ISSN: 2640-3498]. In *Proceedings of Thirty Fifth Conference on Learning Theory*, 867–908. Retrieved September 12, 2022, from https://proceedings.mlr.press/v178/christianson22a.html
- Christianson, N., Shen, J., & Wierman, A. (2023). Optimal robustness-consistency tradeoffs for learning-augmented metrical task systems. In *International Conference on Artificial Intelligence and Statistics*. https://proceedings.mlr.press/v206/ christianson23a.html
- Christianson, N., Werner, L., Wierman, A., & Low, S. (2022). Dispatch-aware planning for feasible power system operation. *Electric Power Systems Research*, 212, 108597. https://doi.org/10.1016/j.epsr.2022.108597
- Deng, K., Sun, Y., Li, S., Lu, Y., Brouwer, J., Mehta, P. G., Zhou, M., & Chakraborty, A. (2015). Model Predictive Control of Central Chiller Plant With Thermal Energy Storage Via Dynamic Programming and Mixed-Integer Linear Programming. *IEEE Transactions on Automation Science and Engineering*, 12(2), 565–579. https://doi.org/10.1109/TASE.2014.2352280
- Donti, P. L., Rolnick, D., & Kolter, J. Z. (2020). DC3: A learning method for optimization with hard constraints. In International Conference on Learning Representations. https://openreview.net/forum?id=V1ZHVxJ6dSS
- Ela, E., & O'Malley, M. (2016). Scheduling and Pricing for Expected Ramp Capability in Real-Time Power Markets. IEEE Transactions on Power Systems, 31(3), 1681–1691. https://doi.org/10.1109/TPWRS.2015.2461535
- Fioretto, F., Mak, T. W., & Van Hentenryck, P. (2020). Predicting AC Optimal Power Flows: Combining Deep Learning and Lagrangian Dual Methods. Proceedings of the AAAI Conference on Artificial Intelligence, 34(01), 630–637. https://doi. org/10.1609/aaai.v34i01.5403
- Hua, B., Schiro, D. A., Zheng, T., Baldick, R., & Litvinov, E. (2019). Pricing in Multi-Interval Real-Time Markets. IEEE Transactions on Power Systems, 34(4), 2696–2705. https://doi.org/10.1109/TPWRS.2019.2891541
- Kelman, A., Ma, Y., & Borrelli, F. (2013). Analysis of local optima in predictive control for energy efficient buildings. *Journal of Building Performance Simulation*, 6(3), 236–255. https://doi.org/10.1080/19401493.2012.671959
- Khodak, M., Balcan, N., Talwalkar, A., & Vassilvitskii, S. (2022). Learning Predictions for Algorithms with Predictions. In Advances in Neural Information Processing Systems. https://openreview.net/forum?id=5OLcPQaYTVg
- Killian, M., & Kozek, M. (2016). Ten questions concerning model predictive control for energy efficient buildings. Building and Environment, 105, 403–412. https://doi.org/10.1016/j.buildenv.2016.05.034

- Kotary, J., Fioretto, F., & Van Hentenryck, P. (2021). Learning Hard Optimization Problems: A Data Generation Perspective. Advances in Neural Information Processing Systems, 34, 24981–24992. Retrieved September 12, 2022, from https:// proceedings.neurips.cc/paper/2021/hash/d1942a3ab01eb59220e2b3a46e7ef09d-Abstract.html
- Langson, W., Chryssochoos, I., Raković, S., & Mayne, D. (2004). Robust model predictive control using tubes. *Automatica*, 40 (1), 125–133. https://doi.org/10.1016/j.automatica.2003.08.009
- Lee, R., Maghakian, J., Hajiesmaili, M., Li, J., Sitaraman, R., & Liu, Z. (2021). Online Peak-Aware Energy Scheduling with Untrusted Advice. In Proceedings of the Twelfth ACM International Conference on Future Energy Systems, 107–123. https://doi. org/10.1145/3447555.3464860
- Lee, Z. J., Chang, D., Jin, C., Lee, G. S., Lee, R., Lee, T., & Low, S. H. (2018). Large-Scale Adaptive Electric Vehicle Charging. In 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), 1–7. https://doi.org/10.1109/SmartGridComm.2018.8587550
- Li, P.,Yang, J., & Ren, S. (2022). Expert-Calibrated Learning for Online Optimization with Switching Costs. In Abstract Proceedings of the 2022 ACM SIGMETRICS/IFIP PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems, 85–86. https://doi.org/10.1145/3489048.3530961
- Li, T., Yang, R., Qu, G., Shi, G., Yu, C., Wierman, A., & Low, S. (2022). Robustness and Consistency in Linear Quadratic Control with Untrusted Predictions. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 6(1), 1–35. https:// doi.org/10.1145/3508038
- Lin,Y., Hu,Y., Shi, G., Sun, H., Qu, G., & Wierman, A. (2021). Perturbation-based Regret Analysis of Predictive Control in Linear Time Varying Systems. Advances in Neural Information Processing Systems, 34, 5174–5185. Retrieved September 12, 2022, from https://papers.nips.cc/paper/2021/hash/298f587406c914fad5373bb689300433-Abstract.html
- Lio,W. H., Rossiter, J., & Jones, B. L. (2014). A review on applications of model predictive control to wind turbines. In 2014 UKACC International Conference on Control (CONTROL), 673–678. https://doi.org/10.1109/CONTROL.2014.6915220
- Lykouris, T., & Vassilvtiskii, S. (2018). Competitive Caching with Machine Learned Advice [ISSN: 2640-3498]. In Proceedings of the 35th International Conference on Machine Learning, 3296–3305. Retrieved January 25, 2022, from https://proceedings. mlr.press/v80/lykouris18a.html
- Meng, K., Dong, Z. Y., Xu, Z., & Weller, S. R. (2015). Cooperation-Driven Distributed Model Predictive Control for Energy Storage Systems. *IEEE Transactions on Smart Grid*, 6(6), 2583–2585. https://doi.org/10.1109/TSG.2015.2449760
- Morari, M., & Lee, J. H. (1999). Model predictive control: Past, present and future. Computers and Chemical Engineering, 16.
- Morstyn, T., Hredzak, B., Aguilera, R. P., & Agelidis, V. G. (2018). Model Predictive Control for Distributed Microgrid Battery Energy Storage Systems. *IEEE Transactions on Control Systems Technology*, 26(3), 1107–1114. https://doi.org/10.1109/ TCST.2017.2699159
- Nellikkath, R., & Chatzivasileiadis, S. (2022). Physics-Informed Neural Networks for AC Optimal Power Flow. *Electric Power Systems Research*, 212, 108412. https://doi.org/10.1016/j.epsr.2022.108412
- Pan, X., Chen, M., Zhao, T., & Low, S. H. (2022). DeepOPF: A Feasibility-Optimized Deep Neural Network Approach for AC Optimal Power Flow Problems [arXiv:2007.01002 [cs, eess]]. https://doi.org/10.48550/arXiv.2007.01002
- Park, S., & Van Hentenryck, P. (2023). Self-supervised primal-dual learning for constrained optimization. In Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence, 37, 4052–4060. https://doi. org/10.1609/aaai.v37i4.25520
- Pon Kumar, S. S., Tulsyan, A., Gopaluni, B., & Loewen, P. (2018). A Deep Learning Architecture for Predictive Control. *IFAC-PapersOnLine*, 51(18), 512–517. https://doi.org/10.1016/j.ifacol.2018.09.373
- Purohit, M., Svitkina, Z., & Kumar, R. (2018). Improving Online Algorithms via ML Predictions. In Advances in Neural Information Processing Systems, 31. Retrieved January 31, 2022, from https://proceedings.neurips.cc/paper/2018/hash/ 73a427badebe0e32caa2e1fc7530b7f3-Abstract.html
- Rutten, D., Christianson, N., Mukherjee, D., & Wierman, A. (2023). Smoothed Online Optimization with Unreliable Predictions. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 7(1), 1–36. https://doi.org/10.1145/3579442
- Schwarm, A. T., & Nikolaou, M. (1999). Chance-constrained model predictive control. AIChE Journal, 45(8), 1743–1752. https:// doi.org/10.1002/aic.690450811
- Spielberg, S., Gopaluni, R., & Loewen, P. (2017). Deep reinforcement learning approaches for process control. In 2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP), 201–206. https://doi.org/10.1109/ADCO-NIP.2017.7983780
- Sultana, W. R., Sahoo, S. K., Sukchai, S., Yamuna, S., & Venkatesh, D. (2017). A review on state of art development of model predictive control for renewable energy applications. *Renewable and Sustainable Energy Reviews*, 76, 391–406. https://doi. org/10.1016/j.rser.2017.03.058
- Troy, N., Denny, E., & O'Malley, M. (2010). Base-Load Cycling on a System With Significant Wind Penetration. IEEE Transactions on Power Systems, 25(2), 1088–1097. https://doi.org/10.1109/TPWRS.2009.2037326
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P.,Weckesser,W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2

- Yeh, C., Li, V., Datta, R., Arroyo, J., Christianson, N., Zhang, C., Chen, Y., Hosseini, M., Golmohammadi, A., Shi, Y., Yue, Y., & Wierman, A. (2023). SustainGym: Reinforcement learning environments for sustainable energy systems. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. https://openreview.net/forum?id= vZ9tA3o3hr
- Zeynali, A., Sun, B., Hajiesmaili, M., & Wierman, A. (2021). Data-driven Competitive Algorithms for Online Knapsack and Set Cover. In Proceedings of the AAAI Conference on Artificial Intelligence, 35(12), 10833–10841. https://doi.org/10.1609/aaai. v35i12.17294
- Zhang, L., Jiang, W., Lu, S., & Yang, T. (2021). Revisiting Smoothed Online Learning. Advances in Neural Information Processing Systems, 34, 13599–13612. Retrieved September 21, 2022, from https://proceedings.neurips.cc/paper/2021/hash/ 70fc5f043205720a49d973d280eb83e7-Abstract.html
- Zhang, L., Tabas, D., & Zhang, B. (2023). An Efficient Learning-Based Solver for Two-Stage DC Optimal Power Flow with Feasibility Guarantees. https://arxiv.org/abs/2304.01409
- Zhao, J., Zheng, T., & Litvinov, E. (2020). A Multi-Period Market Design for Markets With Intertemporal Constraints. IEEE Transactions on Power Systems, 35(4), 3015–3025. https://doi.org/10.1109/TPWRS.2019.2963022

Cite this article: Christianson N, Yeh C, Li T, Hosseini M, Torabi Rad M, Golmohammadi A and Wierman A (2025). Robust machine-learned algorithms for efficient grid operation. *Environmental Data Science*, 4: e24. doi:10.1017/eds.2024.28