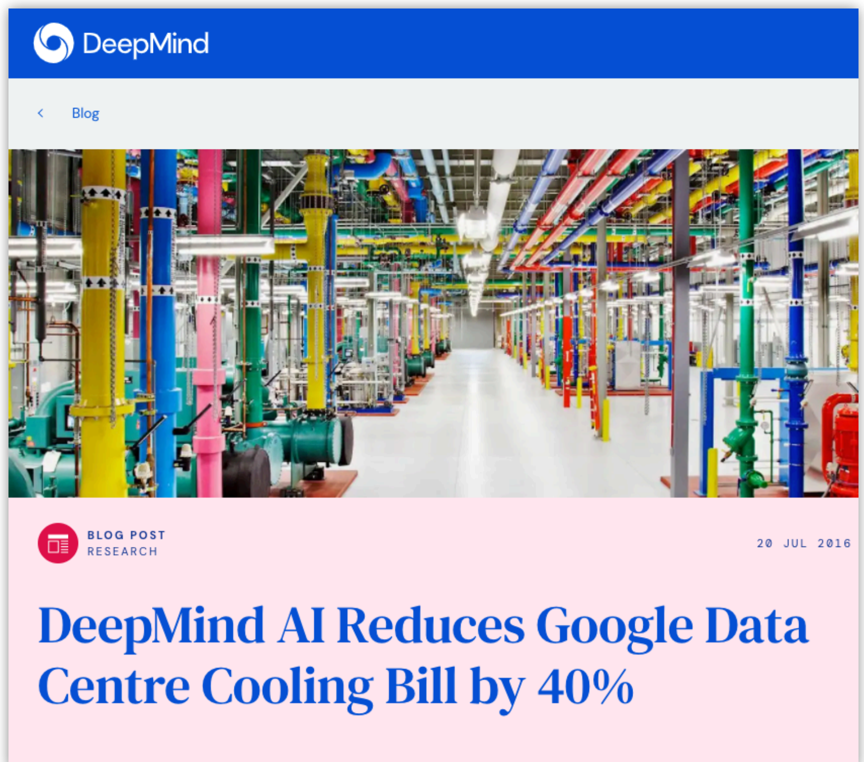
Robustifying machine-learned algorithms for efficient grid operation

Nico Christianson

Caltech

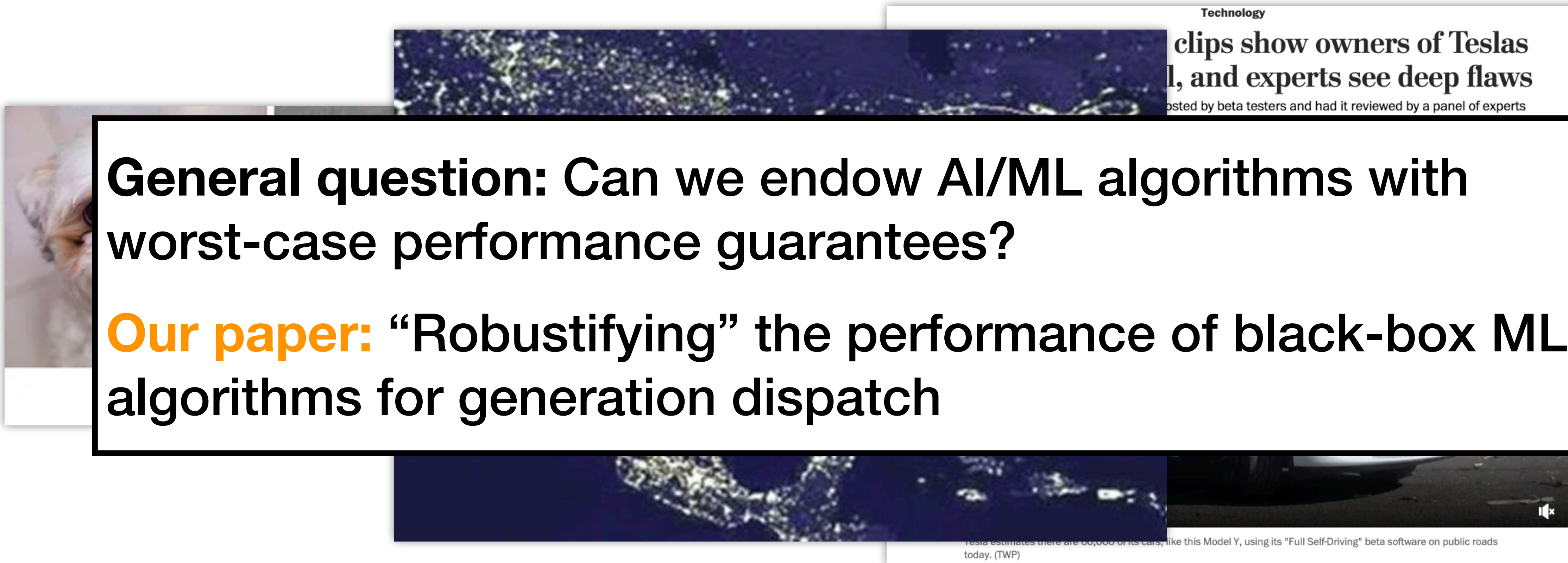
December 9, 2022

AI/ML achieves state-of-the-art performance in many domains:



But…

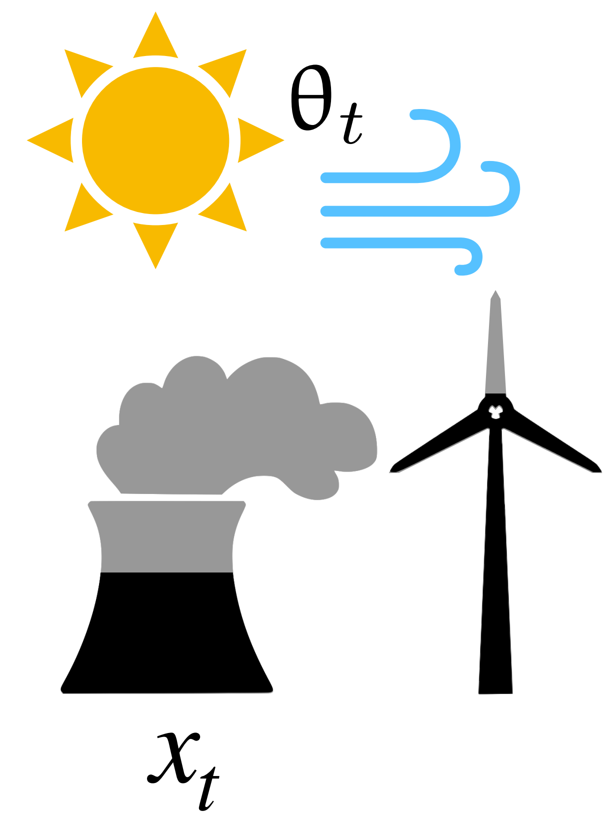
past performance does not guarantee future results!



General question: Can we endow AI/ML algorithms with worst-case performance guarantees?

Our paper: "Robustifying" the performance of black-box ML algorithms for generation dispatch

Problem setting: Generation dispatch with ramp costs



Generation operator faces a sequential problem:

1. At time t, observe ambient conditions θ_t (demand, wind, sun, temperature, etc.)
2. Choose dispatch level(s) $x_t \in X_t$
3. Pay fuel cost $f(x_t; \theta_t)$ and ramp cost $\|x_t - x_{t-1}\|$

Total cost: $\displaystyle\sum_{t=1}^{T} f(x_t; \theta_t) + \|x_t - x_{t-1}\|$

Often (potentially inaccurate) predictions $\hat{\theta}_{t+1|t}, \ldots, \hat{\theta}_{t+w|t}$ of future conditions are available

Dispatch algorithms in practice

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

(i.e., single-step economic dispatch)

$$\text{MPC}: \boldsymbol{\Theta} \mapsto \operatorname*{arg\,min}_{\substack{\mathbf{x}\in\mathbb{R}^d \\ \mathbf{y}_1,\ldots,\mathbf{y}_w\in\mathbb{R}^d}} f(\mathbf{x};\boldsymbol{\theta}_t) + \|\mathbf{x}-\mathbf{x}_{t-1}\| + \sum_{\tau=1}^{w} f(\mathbf{y}_\tau;\hat{\boldsymbol{\theta}}_{t+\tau|t}) + \|\mathbf{y}_\tau - \mathbf{y}_{\tau-1}\| =: \mathbf{x}_t$$

(model predictive control)

But… MPC will be intractable if $f(\,\cdot\,;\theta)$ is nonconvex!

Idea: train an ML algorithm (offline) to mimic MPC

But: ML algorithm generally won't come with worst-case guarantees

Wish to exploit its (likely) good performance while providing worst-case guarantees

Robustifying black-box ML algorithms

Idea: switch back and forth between ML and Greedy algorithms

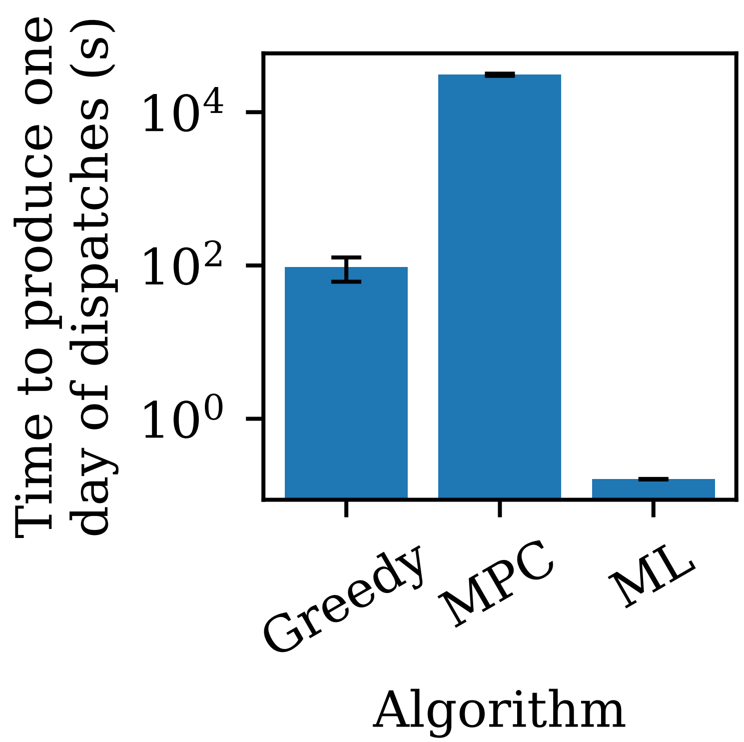
- Follow ML decisions until $\mathrm{Cost(ML)}$ surpasses some threshold h
- Increase h and switch to following the Greedy algorithm until $\mathrm{Cost(Greedy)}$ surpasses h
- Increase h, switch back to ML and repeat…

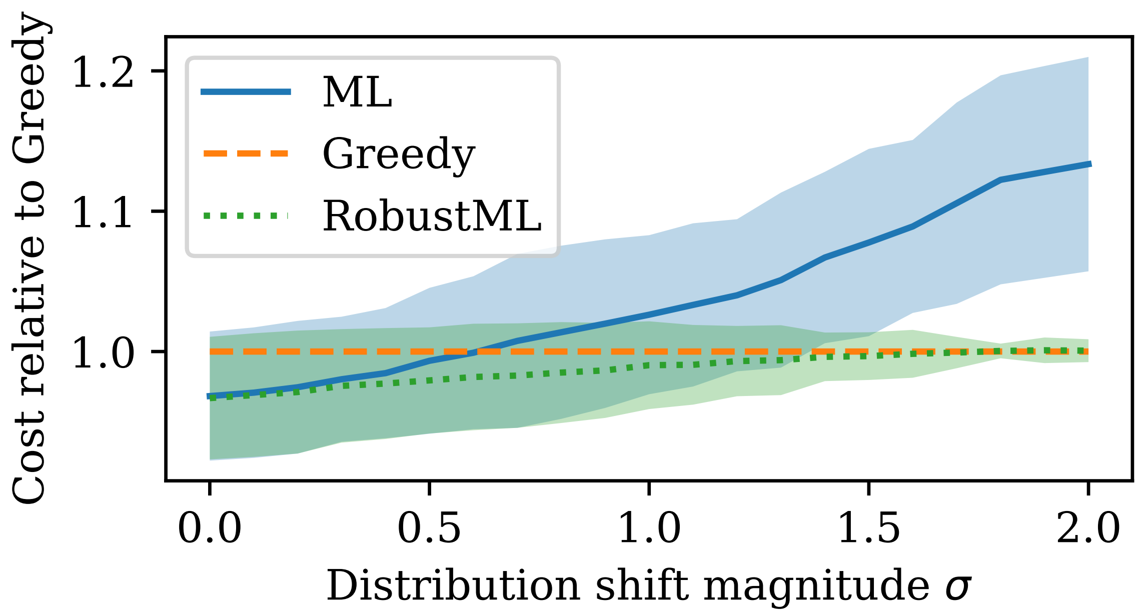
Theorem. For any $\epsilon > 0$, our algorithm **RobustML** achieves cost bounded by

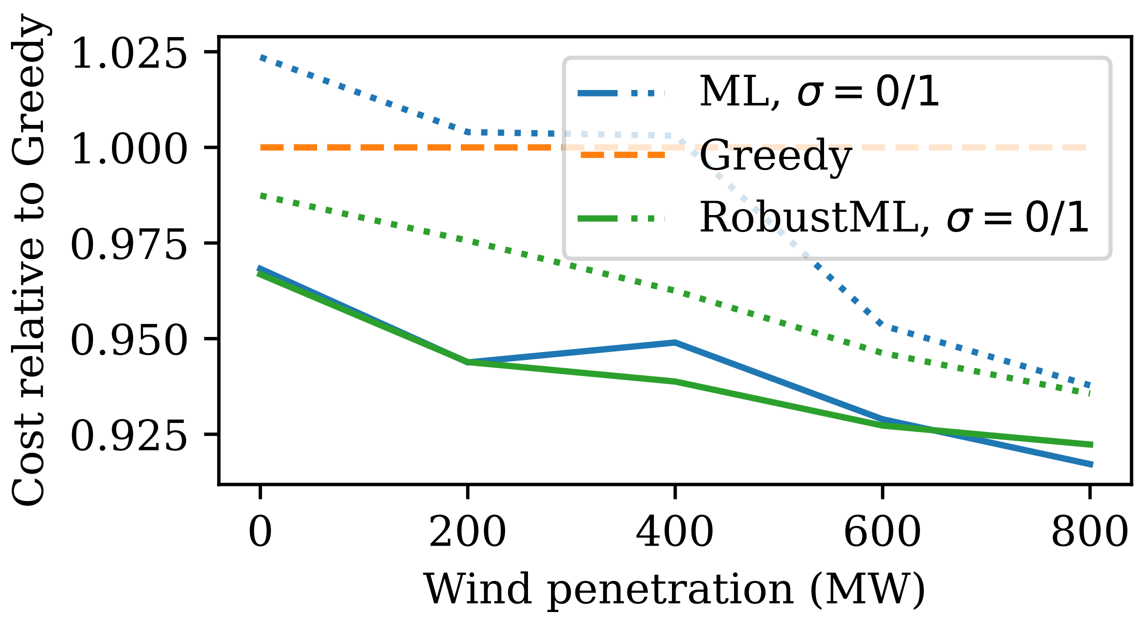
$$\min\{(1+\epsilon)\mathrm{Cost(ML)}, \mathcal{O}(\epsilon^{-1})\mathrm{Cost(Greedy)} + \mathcal{O}(D\epsilon^{-1})\}$$

Experiments: Grid Cogeneration

- Codispatch of electricity & steam on two thermal generators under increasing wind penetration
- Train ML to replicate behavior of MPC
- Combine with Greedy algorithm via RobustML







Thanks for listening!

Please feel free to reach out at nchristi@caltech.edu

Reference:

N. Christianson, C. Yeh, T. Li, M. Torabi Rad, A. Golmohammadi, and A. Wierman. *Robustifying machine-learned algorithms for efficient grid operation*. Workshop on Tackling Climate Change with Machine Learning at NeurIPS 2022.