

Nonparametric Kernel Regression for Coordinated Energy Storage Peak Shaving with Stacked Services

Emily Logan, Ning Qi, Bolun Xu
Department of Earth and Environmental Engineering
Columbia University
New York, NY 10027, USA
{el3350, nq2176, bx2177}@columbia.edu

Abstract—Developing effective control strategies for behind-the-meter energy storage to coordinate peak shaving and stacked services is essential for reducing electricity costs and extending battery lifetime in commercial buildings. This work proposes an end-to-end, two-stage framework for coordinating peak shaving and energy arbitrage with a theoretical decomposition guarantee. In the first stage, a non-parametric kernel regression model constructs state-of-charge trajectory bounds from historical data that satisfy peak-shaving requirements. The second stage utilizes the remaining capacity for energy arbitrage via a transfer learning method. Case studies using New York City commercial building demand data show that our method achieves a 1.3 times improvement in performance over the state-of-the-art forecast-based method, achieving cost savings and effective peak management without relying on predictions.

Index Terms—Peak shaving, energy storage, stacked service, two-stage optimization, kernel regression

I. INTRODUCTION

The rapid growth of electricity demand, along with limited grid infrastructure, has intensified the need for effective demand-side management. Large commercial and industrial consumers are facing significant electricity costs, including both real-time electricity charges and monthly peak demand charges, with the latter being the more substantial. Behind-the-meter battery energy storage systems (BESS) have thus become an attractive solution for peak shaving and electricity cost reduction, while also mitigating grid stress [1], [2].

Peak shaving is challenging, as it requires decision-making under uncertainty over monthly horizons. Many existing methods rely on demand and price forecasting and uncertainty models such as regression [3], neural-networks [4], and chance-constrained models [5]. However, reliable forecasting or uncertainty models at a monthly resolution remain computationally intensive [6]–[8]. The complexity increases significantly when peak shaving is integrated with other grid services, such as energy arbitrage and frequency regulation [9], requiring careful coordination to balance the competing objectives. Other approaches, including dynamic programming [10] and portfolio optimization [11] have also been employed to optimize stacked storage services, but these methods primarily focus on daily peak shaving and struggle with computational burden when adapting to real-time uncertainties or extending to monthly horizons. This limits their practicality for smaller businesses and non-profits, where costs may exceed the potential savings.

This work was partly supported by the Department of Energy under Grant No. DE-EE0011385 and by the National Science Foundation under award ECCS-2239046.

We propose an end-to-end, two-stage framework to coordinate peak shaving with stacked services such as energy arbitrage. Unlike conventional methods that rely on forecasting or computationally intensive optimization, we propose a lightweight online optimization method to jointly predict a dynamic peak demand target and state-of-charge (SoC) reserve trajectory for peak shaving. This enables real-time adaptive control under evolving demand conditions, resulting in faster and more efficient battery operation. The key contributions of this work are summarized as follows:

- 1) **Decoupled Control Formulation:** We show that the stacked service control problem can be reformulated as a two-stage optimization by determining the SoC reserve required for peak shaving first, and then the remaining battery capacity is subsequently allocated to other services.
- 2) **Kernel Regression-Based Prediction:** We develop a data-driven kernel regression algorithm that jointly predicts peak-shaving targets and SoC reserve requirements directly from historical demand and battery characteristics.
- 3) **Joint Peak Shaving and Arbitrage** We integrate the kernel regression predictions with an arbitrage optimization algorithm, enabling decoupled management of peak shaving and energy arbitrage, reducing total electricity costs and mitigating unnecessary battery cycling.
- 4) **Validating Performance through Simulation** The proposed framework is evaluated using 6 years of electricity demand data from a 35-story commercial building in New York City, demonstrating improved peak shaving and cost savings compared to a state-of-the-art method.

The remainder of the paper is organized as follows. Section II introduces the formulation, and Section III details the training and the real-time control algorithm. Section IV shows the simulation and Section V concludes the paper.

II. PROBLEM FORMULATION AND DECOMPOSITION

A. Peak Shaving and Arbitrage Formulation

We consider a commercial building with a co-located, behind-the-meter BESS operating under a time-varying electricity tariff. The objective is to minimize the overall electricity cost by leveraging both energy arbitrage and peak shaving strategies. The deterministic version of the joint peak shaving and arbitrage problem is listed as follows:

$$F_{\text{comb}}(x) = \min_{p,d,q} \sum_{t=1}^T [cd_t + \lambda_t(D_t - d_t + q_t)] + \kappa p \quad (1a)$$

$$\text{s.t. } p \geq D_t - d_t + q_t, \quad \forall t = 1, \dots, T \quad (1b)$$

$$e_{t+1} = e_t - d_t/\eta + \eta q_t, \quad \forall t = 1, \dots, T \quad (1c)$$

$$0 \leq d_t, q_t \leq \bar{P}, \quad \forall t = 1, \dots, T \quad (1d)$$

$$\underline{E} \leq e_{t+1} \leq \bar{E}, \quad \forall t = 1, \dots, T \quad (1e)$$

The decision variables include the discharge energy per timestep (d_t), charge energy per timestep (q_t), and peak demand within the optimization horizon (p). The battery's maximum energy capacity and energy capacity per timestep are denoted by $\bar{E}$ and $\bar{P}$, respectively. $\underline{E}$ denotes the minimum SoC bound and η denotes the efficiency for charging and discharging. The objective function in (1a) minimizes total cost by balancing peak demand charges (κ), real-time energy price (λ), and battery degradation cost (c). The model includes constraints for battery behavior and peak shaving. (1b) defines p as the maximum net demand. SoC dynamics are updated by (1c), and (1d) and (1e) enforce power and energy limits.

B. Decomposition of Stacked Services

The deterministic formulation of the problem in (1) is impractical to solve in practice, as both demand and electricity prices are uncertain, while the peak demand charge is typically settled on a monthly basis. Solving the problem over such a long uncertainty horizon is therefore computationally intractable. To address this challenge, based on the fact that the peak penalty κ is much larger than the energy cost coefficients c and λ_t in practice, (1) can be effectively decomposed into two sequential subproblems. This decomposition first prioritizes minimizing peak demand and then performs energy arbitrage over the resulting feasible set.

Peak Shaving. In the first stage, we minimize the peak demand and determine the associated trajectory of the SoC reserve necessary for peak shaving. In comparison to the objective in (1), we introduce a small tie-breaking term $\delta \sum_{t=1}^T e_{t+1}$ in the objective. This term keeps SoC levels low but does not affect the peak value when δ is sufficiently small. Solving (2) thus yields an optimal peak demand p^* and SoC profile e^* .

$$F_{\text{PS}}(x; \delta) = \min_{p,d,q,e} p + \delta \sum_{t=1}^T e_{t+1} \quad (2)$$

s.t. Constraints (1b)–(1e)

Arbitrage. Given the Stage 1 solution (p^* , e^*), Stage 2 maximizes the revenue from energy arbitrage while ensuring that arbitrage decisions respect both the peak shaving constraint $p \leq p^*$ and the SoC trajectory established in Stage 1.

$$F_{\text{arb}}(x) = \min_{d,q,e} \sum_{t=1}^T [cd_t - \lambda_t(d_t - q_t)] \quad (3a)$$

$$\text{s.t. } p^* \geq D_t - d_t + q_t, \quad \forall t = 1, \dots, T \quad (3b)$$

$$e_{t+1} \geq e_{t+1}^*, \quad \forall t = 1, \dots, T \quad (3c)$$

(1c) – (1e).

We now present the following proposition proving that the decomposition is equivalent to the original problem:

Proposition 1. *Given that δ is sufficiently small and $\kappa \gg \lambda_t, c$, solving the two-stage optimization in (2) and (3) is equivalent to solving the combined peak shaving and arbitrage problem in (1).*

Proof. We first restate the combined problem for convenience:

$$F_{\text{comb}}(x) = \sum_{t=1}^T \lambda_t D_t + F_{\text{arb}}(x) + \kappa F_{\text{PS}}(x).$$

Let $x = (p, \{d_t, q_t, e_{t+1}\}_{t=1}^T)$ denote the decision variables, and let S be the feasible region defined by (1b)–(1e), where $g(x) \leq 0$ collects all inequality constraints and $h(x) = 0$ the equality constraint.

The Lagrangian of the combined problem is

$$\mathcal{L}_{\text{comb}}(x, \mu, \nu) = F_{\text{arb}}(x) + \kappa F_{\text{PS}}(x) + \mu^T g(x) + \nu^T h(x), \quad (4)$$

with $\mu \geq 0$ and ν denoting the dual variables. The KKT stationarity condition is

$$\nabla_x F_{\text{arb}}(x) + \kappa \nabla_x F_{\text{PS}}(x) + \nabla g(x)^T \mu + \nabla h(x)^T \nu = 0, \quad (5)$$

together with primal and dual feasibility and complementary slackness.

To analyze the limit $\kappa \gg \lambda_t, c$, define rescaled dual variables $\tilde{\mu} = \mu/\kappa$ and $\tilde{\nu} = \nu/\kappa$. Substituting into (5) and dividing by κ yields

$$\nabla_x F_{\text{arb}}(x)/\kappa + \nabla_x F_{\text{PS}}(x) + \nabla g(x)^T \tilde{\mu} + \nabla h(x)^T \tilde{\nu} = 0. \quad (6)$$

Taking the limit as $\kappa \rightarrow \infty$ removes the first term, giving

$$\nabla_x F_{\text{PS}}(x) + \nabla g(x)^T \tilde{\mu} + \nabla h(x)^T \tilde{\nu} = 0, \quad (7)$$

which are precisely the KKT conditions of the peak shaving problem. Thus, for $\kappa \gg \lambda_t, c$ and $\delta \rightarrow 0^+$, the solution to (1) coincides with the peak shaving optimum: $p_{\text{comb}}^* = p^*$ and $e_{\text{comb}}^* = e^*$.

Since the combined solution minimizes $F_{\text{PS}}(x)$ under these conditions, it lies in the optimal peak shaving set $S^* = \{x \in S \mid F_{\text{PS}}(x) = F_{\text{PS}}^*\}$. Restricting the combined problem to S^* , we have:

$$\min_{x \in S^*} [F_{\text{arb}}(x) + \kappa F_{\text{PS}}(x)].$$

As $F_{\text{PS}}(x) = F_{\text{PS}}^*$ for all $x \in S^*$, the constant term κF_{PS}^* can be dropped, leaving $\min_{x \in S^*} F_{\text{arb}}(x)$, identical to the arbitrage formulation (3) over the feasible set characterized by $p \leq p^*$ and $e_{t+1} \geq e_{t+1}^*$ for all t .

Therefore, when κ is sufficiently large, solving the combined problem (1) is equivalent to the two-stage procedure of first solving the peak shaving problem (2) to obtain (p^* , e^*), and then optimizing arbitrage over S^* . $\square$

In this work, our primary contribution is the development and evaluation of a peak shaving strategy, including its integration with other stacked services. Energy arbitrage is handled using an established non-anticipatory policy from [12].

III. LEARNING AND CONTROL METHOD

We now present our main methods for deriving a non-anticipatory algorithm to solve the joint peak shaving and arbitrage problem using the decomposition approach. Our general approach is as follows: 1) we use historical demand data to generate the SoC reserve target and peak shaving for peak shaving, conditioned on monthly peak shaving targets; 2) we use a kernel regression algorithm with a look-back demand window to jointly predict the SoC reserve target and the peak shaving target; 3) we input the predicted SoC reserve target into a real-time peak shaving and arbitrage controller.

A. Kernel Regression for SoC Reserve Prediction

The kernel regression model predicts the minimum SoC reserve required for peak shaving at each timestep by comparing the current demand pattern to historically observed demand trajectories. We first use historical demand data and perform (2) to generate the hindsight-optimal SoC reserve target e_s^{hist} and the net demand for each timestep. From the latter, we extract the maximum value within each day, which we define as the hindsight-optimal peak shaving target p_s^{hist} . Note that we use s to represent the time index in the training data and t in the real-time control to reduce confusion of the time index.

Training data generation. Let the full historical demand record be $D^{\text{hist}} = \{D_1^{\text{hist}}, \dots, D_N^{\text{hist}}\}$, where N is the total number of historical timesteps. Let T be the length of the sliding demand windows, then demand look-back vectors are constructed as $\mathbf{D}_s^{\text{hist}} = [D_{s-T+1}^{\text{hist}}, \dots, D_s^{\text{hist}}]$, where each window endpoint s is associated with a corresponding next minimum SoC reserve e_{s+1}^{hist} and peak shaving target p_{s+1}^{hist} obtained from the prior training results. Hence each training data entry can be represented as

$$\{e_{s+1}^{\text{hist}}, p_{s+1}^{\text{hist}} \mid \mathbf{X}_s^{\text{hist}} = [\mathbf{D}_s^{\text{hist}}, t_{\sin,s}, t_{\cos,s}]\} \quad (8)$$

where $t_{\sin,t}$ and $t_{\cos,t}$ are sin and cosine vectors with a daily period to capture the time-of-the-day feature.

Prediction Feature. In prediction, we assemble the input feature similarly to $\mathbf{X}_s^{\text{hist}}$. Let the observed demand sequence up to the current timestep t be $D = \{D_1, \dots, D_t\}$, and at each timestep $t \geq T$, the most recent sequence of T demand values is represented as $\mathbf{D}_t = [D_{t-T+1}, \dots, D_t]$. The prediction feature vector is thus assembled as

$$\mathbf{X}_t = [\mathbf{D}_t, t_{\sin,t}, t_{\cos,t}], \quad (9)$$

α -Confidence Kernel Regression. For each current feature vector $\mathbf{X}_t$, the set of K most similar historical vectors, denoted $\text{KNN}(t)$, is identified using Euclidean distance. The similarity between $\mathbf{X}_t$ and each neighbor $\mathbf{X}_s^{\text{hist}}$ is quantified using a Gaussian kernel, and the resulting weights are normalized across the K nearest neighbors so that they sum to one:

$$w_{t,s} = \frac{\exp(-\|\mathbf{X}_t - \mathbf{X}_s^{\text{hist}}\|_2^2 / (2T\sigma^2))}{\sum_{j \in \text{KNN}(t)} \exp(-\|\mathbf{X}_t - \mathbf{X}_j^{\text{hist}}\|_2^2 / (2T\sigma^2))}. \quad (10)$$

where $s \in \text{KNN}(t)$, σ is the kernel bandwidth parameter that controls how rapidly the influence of historical windows decays with dissimilarity to the current feature profile.

The resulting weights $w_{t,s}$ are used to compute the predicted required SoC reserve at each timestep. We introduce a confidence-level parameter $\alpha \in (0, 1)$ to provide a tunable conservativeness ensuring the predicted SoC reserve is greater than α ratio of the training data. The predicted SoC for the next time step $t + 1$ is therefore

$$\hat{e}_{t+1}^\alpha = e_k^{\text{hist}} + \frac{\alpha - W_{t,k}}{W_{t,k} - W_{t,k-1}}(e_k^{\text{hist}} - e_{k-1}^{\text{hist}}) \quad (11a)$$

where

$$W_{t,k-1} \leq \alpha \leq W_{t,k} \quad (11b)$$

$$W_{t,k} = \sum_{i=1}^k w_{t,i} \quad (11c)$$

$$k \text{ is the ascending rank of } e_{s+1}^{\text{hist}}, s \in \text{KNN}(t) \quad (11d)$$

The historical SoC reserve values of the K nearest neighbors are sorted in ascending order, and their corresponding weights are summed. The SoC value at which the cumulative weight first reaches α is selected as the predicted SoC reserve, allowing the estimate to be adjusted to be more conservative or aggressive depending on operational preferences.

Unlike the SoC reserve, the peak shaving target p_{t+1}^{pred} is computed as a simple weighted average:

$$p_{t+1}^{\text{pred}} = w_{t,s} p_s^{\text{hist}}, \quad s \in \text{KNN}(t). \quad (12)$$

No confidence-level adjustment is applied here because the real-time controller described in Section III-B dynamically adapts peak-shaving targets based on the observed net demand, ensuring conservativeness is enforced through the adaptive control rather than through the kernel regression itself.

B. Real-Time Control

Based on the predicted SoC reserve estimates and daily net demand targets, a real-time controller determines the battery charge schedule, discharge schedule, and SoC trajectory to achieve peak shaving while ensuring feasible operation. This SoC trajectory establishes the energy that must be reserved for peak shaving at each time step. The remaining battery capacity can then be allocated to additional services, such as arbitrage. The control logic operates as described by Algorithm 1.

C. Parameter Search

The kernel regression model relies on three key hyperparameters—the kernel bandwidth σ , the look-back window T , and the number of nearest neighbors K , which directly influence the performance of the combined peak shaving and arbitrage strategy. Performance is evaluated using two metrics: (1) *total cost savings*, defined as the reduction in electricity cost relative to a no-storage baseline, and (2) *annual battery cycles*, calculated as the total discharge energy divided by the nominal battery capacity $\bar{E}$, indicating utilization and wear. To systematically identify effective hyperparameters, we implement an automated, tiered search that adaptively tunes each in sequence. The look-back window T is optimized first through a coarse-to-fine search to find the most informative time horizon. Next, the kernel bandwidth σ is tuned using a logarithmic-scale exploration followed by linear refinement to strike a balance between accuracy and smoothness. Finally,

Algorithm 1: Real-Time Peak Shaving and Stacked Services Control

Input: $D_t, e_t, \hat{e}_{t+1}^\alpha, p_{t-1}, p_{t+1}^{pred}, (\bar{P}, \bar{E}, \eta), q_{t,arb}, d_{t,arb}$ [12].

Output: $q_t, d_t, e_{t+1}, p_{t+1}, D_t^{net}$.

Stage 1: Peak Shaving

$\Delta e_t = \hat{e}_{t+1}^\alpha - e_t; \quad p_t = \max\{p_{t-1}, p_{t+1}^{pred}\}$

if $D_t > p_t$ **then**

$d_{t,PS} = \min(\max(-\Delta e_t \eta, D_t - p_t), \bar{P}, e_t \eta);$
 $q_{t,PS} = 0$

else if $e_t < \hat{e}_{t+1}^\alpha$ **and** $D_t < p_t$ **then**

$q_{t,PS} = \min(\Delta e_t / \eta, p_t - D_t, \bar{P}, (\bar{E} - e_t) / \eta);$
 $d_{t,PS} = 0$

else

$q_{t,PS} = d_{t,PS} = 0$

$e_{t+1,PS} = e_t - d_{t,PS} / \eta + q_{t,PS} \eta; \quad p_{t+1} = \max\{p_t, D_t^{net}\}$

Stage 2: Arbitrage

$q_{t,max} = \min(p_{t+1} - D_t, \bar{P}, (\bar{E} - e_t) / \eta);$

$d_{t,max} = \min((e_t - e_{t+1,PS}) \eta, \bar{P})$

if $q_{t,PS} > 0$ **then**

$q_t = \min(q_{t,arb} + q_{t,PS}, q_{t,max}); \quad d_t = 0$

else if $d_{t,PS} > 0$ **then**

$d_t = \min(d_{t,arb} + d_{t,PS}, d_{t,max}); \quad q_t = 0$

else

$q_t = \min(q_{t,arb}, q_{t,max}); \quad d_t = \min(d_{t,arb}, d_{t,max})$

$e_{t+1} = e_t - d_t / \eta + q_t \eta; \quad D_t^{net} = D_t - d_t + q_t$

the number of nearest neighbors K is adjusted through broad and refined searches to maximize performance.

IV. SIMULATION AND RESULTS

We use a real-world case study based on a 35-story, 800,000 ft², Class A commercial building in New York City. The demand data for this building, provided by Nantum AI¹, is shown in Fig. 1. Peak demand charges follow Con Edison's standard rate, which applies a charge of \$42.80/kW to the monthly peak demand from June 1 to September 30 and \$33.50/kW during other months, plus a flat monthly customer charge of \$71. The monthly peak demand is defined as the average of the two highest consecutive 15-minute intervals of demand². Arbitrage is performed using real-time electricity prices from the New York Independent System Operator³. To account for seasonal patterns in demand and pricing, the kernel regression model considers only historical data from the same season (either June–September or the other months) when predicting SoC reserve and net demand targets.

Historical building data from 2019–2023 is used for model training, and the data from 2024 is reserved for testing. The battery duration ($\bar{E}/\bar{P}$) is 2.2 hours, with an initial state of charge (SoC) of 50% (e_0), a minimum SoC ($\underline{E}$) of 20% of $\bar{E}$, and a round-trip efficiency (η) of 90%. Parameters are chosen by the automatic algorithm described in Section III-C. All simulations are conducted at a 5-minute resolution on a 10-core Apple M4 processor, and all optimization problems are solved

¹https://github.com/emlog9/peak_shaving_kernel_regression

²<https://www.coned.com/en/accounts-billing/your-bill/time-of-use>

³<https://www.nyiso.com/energy-market-operational-data>

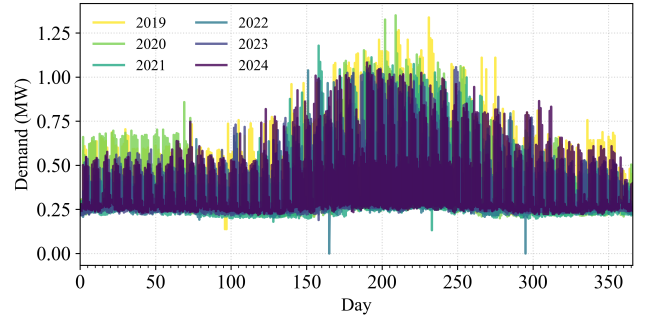


Fig. 1. Building electricity demand from 2019–2024. Line colors indicate years. Yearly standard deviations are: 2019: 0.21 MW, 2020: 0.16 MW, 2021: 0.16 MW, 2022: 0.16 MW, 2023: 0.18 MW, and 2024: 0.19 MW.

using Gurobi 12.0.2. Model performance is evaluated based on total cost savings compared to the no-storage scenario and battery cycling. The results for peak shaving and arbitrage are compared against two benchmarks: (i) the deterministic case outlined in (1), limited to an average of one cycle per day, and (ii) a demand-forecasting controller based on a recursive XGBoost model with a **mean absolute error (MAE) of 0.03 MW**. This model utilizes calendar features, temperature, and lagged demand to generate day-ahead demand forecasts, which are then input into the peak shaving optimization from (2). From there, the real-time controller determines the charge and discharge schedule. As with the kernel regression framework, these results are subsequently combined with an arbitrage schedule from [12] in a two-stage approach. Simulations are performed for battery power ratings ranging from 0.1 MW to 1.0 MW.

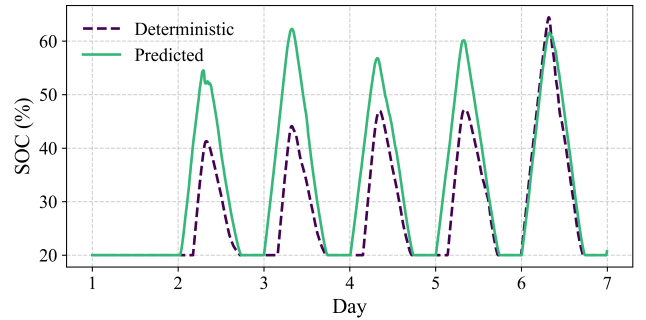


Fig. 2. Comparison of the kernel-regression predicted and deterministic minimum SoC reserve trajectories for a 1 MW battery.

Fig. 2 compares the predicted SoC reserve trajectory to the deterministic SoC trajectory for a representative week. The predicted SoC generally tracks the timing and magnitude of the actual SoC peaks, though it tends to slightly overestimate the reserve. This behavior aligns with the design of the confidence level parameter (α), which accounts for uncertainty by allowing for more conservative SoC predictions.

Fig. 3 shows the peak demand across the three controller types in comparison to the peak demand with no battery storage. Overall, the kernel regression controller effectively shaves the peak demand, outperforming the demand forecasting model in most months.

To evaluate the economic impact of the proposed control

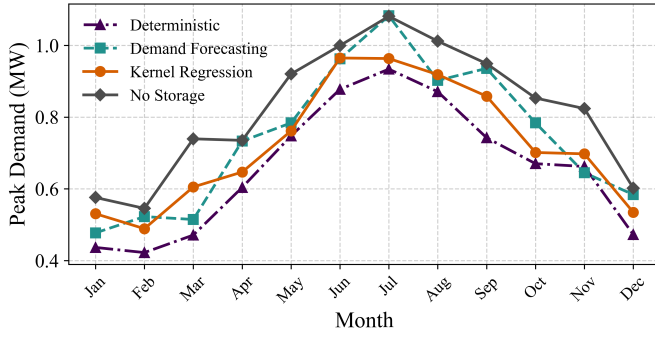


Fig. 3. Peak demand (MW) by month for perfect foresight, kernel regression-based, and demand forecasting control algorithms for a 0.6 MW battery.

strategy, we analyze annual cost savings relative to a scenario with no storage, where the total electricity cost is \$512.5k, as a function of battery power capacity (MW). This metric captures the economic benefit of combined peak shaving and energy arbitrage given real-time electricity prices and demand patterns. Fig. 4 compares the results for the kernel regression-based controller with the deterministic and demand-forecasting baseline cases.

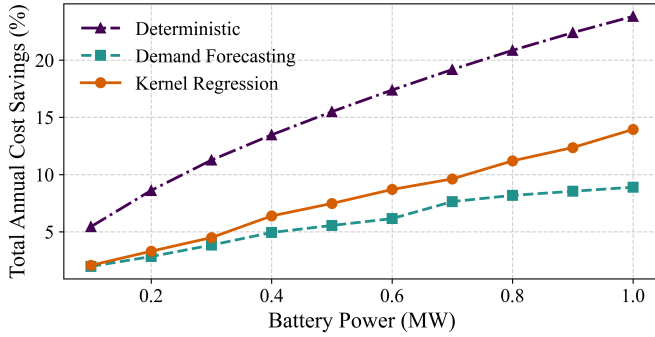


Fig. 4. Total annual cost savings (%) vs. battery size for perfect foresight, kernel regression-based, and demand forecasting control algorithms.

Larger batteries achieve higher savings while the increasing trend saturates. Across the battery sizes evaluated in this study, the kernel regression model captures between 38.1% and 58.5% of the deterministic cost savings, outperforming the demand forecasting controller for each battery size, which achieves only 33.0% to 39.8% of the deterministic cost savings. It is worth noting that the deterministic perfect foresight assumes one month of perfect demand and price forecast (based on a monthly utility bill settlement). Hence, the performance of the perfect forecast is drastically better.

Battery cycling is an important metric for understanding battery utilization and wear. The deterministic controller operates near its cycle limit of one cycle per day on average, ranging from 346 to 366 cycles annually across the range of battery power capacities (0.1 MW to 1 MW). This behavior indicates that with perfect foresight, maximizing cost savings and fully exploiting peak shaving and arbitrage opportunities requires aggressive daily cycling. In contrast, the kernel regression controller exhibits more variable cycling, ranging from approximately 76 to 126 cycles per year across battery sizes. This

variation arises because the controller relies on predicted SoC trajectories rather than perfect knowledge of future demand and prices. Underestimation or overestimation of the SoC can cause missed opportunities to charge or discharge at optimal times. The demand forecasting model cycles even less, ranging from 57 to 95 annual cycles.

V. CONCLUSION

This work presents an end-to-end framework for behind-the-meter peak shaving and stacked energy services in commercial buildings. The method employs a two-stage decomposition, where a nonparametric kernel regression model generates a state-of-charge reserve trajectory for real-time peak shaving control without requiring explicit demand forecasts. The remaining battery capacity is then allocated to energy arbitrage via a non-anticipatory control strategy. A real-world case study demonstrates that the proposed approach outperforms a standard demand forecasting baseline, capturing 38.1%-58.5% of the cost savings achieved by a perfect-foresight benchmark. The results highlight the method's practicality and economic effectiveness for real-time battery management. For future work, we aim to enhance the kernel regression model and integrate additional services such as frequency regulation.

REFERENCES

- [1] J. Li, G. Mu, J. Zhang, C. Li, G. Yan, H. Zhang, and G. Chen, "Dynamic economic evaluation of hundred megawatt-scale electrochemical energy storage for auxiliary peak shaving," *Protection and Control of Modern Power Systems*, vol. 8, no. 3, pp. 1–18, 2023.
- [2] B. T. Mirtetz and N. D. Laws, "Impacts of dispatch strategies and forecast errors on the economics of behind-the-meter pv-battery systems," National Renewable Energy Laboratory (NREL), Golden, CO, USA, Tech. Rep. NREL/CP-7A40-86194, July 2023.
- [3] R. T. de Salis, A. Clarke, Z. Wang, J. Moyne, and D. Tilbury, "Energy storage control for peak shaving in a single building," *IEEE PES General Meeting*, pp. 1–5, 2014.
- [4] X. Zhang, T. K. Chau, Y. H. Chow, T. Fernando, and H. H.-C. Iu, "A novel sequence to sequence data modelling based cnn-lstm algorithm for three years ahead monthly peak load forecasting," *IEEE Transactions on Power Systems*, vol. 39, no. 1, pp. 1932–1947, 2023.
- [5] N. Qi, P. Pinson, M. R. Almassalkhi, L. Cheng, and Y. Zhuang, "Chance-constrained generic energy storage operations under decision-dependent uncertainty," *IEEE Transactions on Sustainable Energy*, vol. 14, no. 4, pp. 2234–2248, 2023.
- [6] M. Kaut, "Handling of long-term storage in multi-horizon stochastic programs," *Computational Management Science*, vol. 21, no. 1, p. 27, 2024.
- [7] C. Cortes-Aguirre, Y.-A. Chen, A. Ghosh, J. Kleissl, and A. Khurram, "Economic mpc with an online reference trajectory for battery scheduling considering demand charge management," *IEEE Transactions on Smart Grid*, 2025.
- [8] Y. Mo, Q. Lin, M. Chen, and S.-Z. J. Qin, "Optimal online algorithms for peak-demand reduction maximization with energy storage," in *Proceedings of the twelfth ACM international conference on future energy systems*, 2021, pp. 73–83.
- [9] Y. Shi, B. Xu, D. Wang, and B. Zhang, "Using battery storage for peak shaving and frequency regulation: Joint optimization for superlinear gains," *IEEE transactions on power systems*, vol. 33, no. 3, pp. 2882–2894, 2017.
- [10] J. Engels, B. Claessens, and G. Deconinck, "Optimal combination of frequency control and peak shaving with battery storage systems," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3270–3279, 2020.
- [11] N. Qi, L. Cheng, H. Li, Y. Zhao, and H. Tian, "Portfolio optimization of generic energy storage-based virtual power plant under decision-dependent uncertainties," *Journal of energy storage*, vol. 63, p. 107000, 2023.
- [12] Y. Baker, N. Zheng, and B. Xu, "Transferable energy storage bidder," *IEEE Transactions on Power Systems*, vol. 39, no. 2, pp. 4117–4126, 2023.