Learning from Optimal: Energy Procurement Strategies for Data Centers

Sohaib Ahmad, Arielle Rosenthal, Mohammad H. Hajiesmaili, Ramesh K. Sitaraman
UMass Amherst
{sohaib,ariellerosen,hajiesmaili,ramesh}@cs.umass.edu

ABSTRACT

Environmental concerns and rising grid prices have motivated data center owners to invest in on-site renewable energy sources. However, these sources present challenges as they are unreliable and intermittent. In an effort to mitigate these issues, data centers are incorporating energy storage systems. This introduces the opportunity for electricity bill reduction, as energy storage can be used for power market arbitrage.

We present two supervised learning-based algorithms, `LearnBuy`, that learns the amount to purchase, and `LearnStore`, that learns the amount to store, to solve this energy procurement problem. These algorithms utilize the idea of "learning from optimal" by using the values generated by the offline optimization as a label for training. We test our algorithms on a general case, considering buying and selling back to the grid, and a special case, considering only buying from the grid. In the general case, `LearnStore` achieves a 10-16% reduction compared to baseline heuristics, whereas in the special case, `LearnBuy` achieves a 7% reduction compared to prior art.

CCS CONCEPTS

• Computing methodologies → Supervised learning; **• Hardware → Power and energy**; **• Applied computing →** *Data centers.*

ACM Reference Format:
Sohaib Ahmad, Arielle Rosenthal, Mohammad H. Hajiesmaili, Ramesh K. Sitaraman. 2019. Learning from Optimal: Energy Procurement Strategies for Data Centers. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems (e-Energy '19), June 25–28, 2019, Phoenix, AZ, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3307772.3328308

1 INTRODUCTION

The electricity bill is a substantial part of the operating cost of data centers and managing this cost has become critically important [10, 13, 14, 18, 20–22, 27, 28, 33] in the recent years. A promising direction to manage this cost is to procure the total energy of data centers from a variety of sources including the electric grid, renewables, and energy storage systems. A few notable examples include the Google data center in Belgium [5] and the Amazon data center in Virginia [3] with on-site solar farms, energy storage in a Google data center in Taiwan [1], and Tesla batteries used to power an Amazon data center in California [2]. The addition of renewables and storage systems presents a great opportunity for shifting energy usage over time to reduce the cost. *However, it also raises the question of how to orchestrate the energy procurement among different sources (i.e., grid, renewables, and storage) such that the electricity bill is minimized.*

Our paper is focused on the above challenge of energy procurement in a data center with on-site renewables and storage. Figure 1 depicts a typical energy procurement scenario in such a data center. At each time t, the total energy demand of the servers in the data center, $d_{\mathtt{tot}}(t)$, must be met by drawing energy from three sources: (i) $x_{\mathtt{d}}(t)$ units from the grid, (ii) $u(t)$ units from the renewable, and (iii) $b_{\mathtt{d}}(t)$ units by discharging the storage. To discharge the storage, one must have stored that energy in it at earlier time steps. Thus, at a time t, we could also choose to charge the storage by storing $x_{\mathtt{b}}(t)$ units from the grid. Further, the storage can store at most B units and may have further limitations on the charge and discharge rates. Finally, $b_{\mathtt{g}}(t)$ is the energy flow by discharging the storage to the grid, which captures the possibility of selling to the grid.

An *energy procurement* (EP) algorithm must decide what amounts to draw from the grid ($x(t)$) and the amounts to discharge ($b_{\mathtt{d}}(t) + b_{\mathtt{g}}(t)$) or store in the battery ($x_{\mathtt{b}}(t)$), given the current unit price of the grid energy ($p(t)$), amount of renewable generation ($u(t)$), and the total energy demand of the data center ($d_{\mathtt{tot}}(t)$). The objective of an EP algorithm is to minimize the total electricity cost of $\sum_t x(t)p(t) - \sum_t p(t)b_{\mathtt{g}}(t)$, where the first term is the cost of buying from the grid and the second term is the revenue from selling to the grid. In addition the algorithm should respect the constraints of the storage including charge/discharge rate and capacity limits. Designing energy procurement algorithms is difficult since the future values of the grid energy prices, the renewable generation, and the energy demand are unknown, highly variable, and unpredictable.

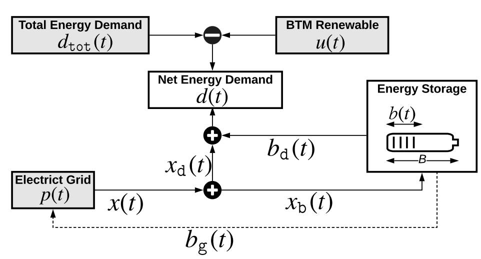


Figure 1: The energy procurement scenario with the possibility of selling back to the grid (shown by the dashed line)

Similar problems have been studied in the literature using empirical evaluations [13, 25, 34, 35], stochastic optimization [14, 17, 29, 30], and competitive design [10, 32]. All above approaches develop

e-Energy '19, June 25–28, 2019, Phoenix, AZ, USA

ACM ISBN 978-1-4503-6671-7/19/06…$15.00
https://doi.org/10.1145/3307772.3328308

simple and fixed decision rules based on simplified mathematical modeling of the underlying problem. Thus, they fail to capture general scenarios, e.g., negative energy pricing and selling back to the grid. In this paper, we pursue a new direction and study the general problem of energy procurement for data centers using a supervised learning approach.

We make the following key contributions.

1) Our primary contribution is an approach for energy procurement using the concept of *learning from the optimal* [8]. While prior work focused on designing fixed control rules based on simplified models, we devise a supervised learning approach that derives the optimal buy/sell/store decisions for training data consisting of historical energy prices, data center energy demands, and renewable generation. We then use these optimal decisions on the training inputs as labels to train our learning algorithms. Thus, our algorithms learn from a provably-optimal offline decision maker.

2) We define price-, demand-, storage-, and time-related features and devise two classes of learning algorithms: `LearnBuy` that attempts to learn how much energy to buy/sell to the grid at each time step; and `LearnStore` that attempts to learn what storage level should be maintained at each step. For both `LearnBuy` and `LearnStore`, we explore the effectiveness of different learning models such as decision trees, k-nearest neighbors, linear regression, and deep neural networks, and present results for the best model in each case.

3) To derive the real-world efficacy of our algorithms, we evaluate them using extensive data traces of electricity prices from the New York market (NYISO [24]), energy demands from multiple data centers of Akamai's CDN [23], and renewable production values from wind installations [4]. As the metric of evaluation, we use the notion of *normalized cost* which is ratio of the cost achieved by our algorithm with the cost achieved by the offline optimal algorithm for the same set of inputs.[1] We also consider two scenarios: a *general case* scenario where the data center is able to both buy and sell energy to the grid; and a *special case* scenario where the data center is only able buy from the grid and selling back is disallowed.

- **a)** For the general case scenario, `LearnStore` performs better than `LearnBuy` achieving a normalized cost of 1.13 (resp., 1.45) in the case where 5% (resp., 10%) of the storage can be charged/ discharged at each time step. In both cases, `LearnStore` performs 10.34% to 16.18% better than `PreDay`, an intuitive data-driven heuristic that uses the optimal decisions of the previous day to perform actions in the current day.
- **b)** For the special case scenario, `LearnBuy` performs better than `LearnStore` achieving a normalized cost of 1.16 (resp., 1.19) in the case where 5% (resp., 10%) of the storage can be charged/discharged. `LearnBuy` achieves a cost 3.20% to 3.67% better than `PreDay`. Unlike the general scenario where there are no known theoretically-validated algorithms, the special case scenario has had some recent literature. In particular, an online algorithm `BatManRate` [32] is known to have the smallest competitive ratio for the problem. However, `LearnBuy` is achieves a cost that is 7.18% to 7.26% smaller than `BatManRate`.

[1] Note that the cost ratio is always at least 1. However, no online algorithm may able to achieve a value of 1, since the offline optimal algorithm has the benefit of knowing the future values of all inputs.

2 THE ENERGY PROCUREMENT PROBLEM

An optimal energy procurement algorithm minimizes the energy procurement cost, i.e., purchased cost subtracted by the revenue of selling back to the grid, over the time horizon T. We can formulate the energy procurement problem as follows.

$$
\begin{aligned}
\text{EP}: \quad & \min \quad \sum_{t\in\mathcal{T}} p(t)(x(t) - b_{\text{g}}(t)) \\
& \text{s.t.}: \quad \forall t \in \mathcal{T}: \\
& \qquad x(t) = x_{\text{d}}(t) + x_{\text{b}}(t), && (1)\\
& \qquad d(t) = x_{\text{d}}(t) + b_{\text{d}}(t), && (2)\\
& \qquad b(t) = b(t-1) + x_{\text{b}}(t) - b_{\text{d}}(t) - b_{\text{g}}(t), && (3)\\
& \qquad 0 \le x_{\text{b}}(t) \le \min\{\rho_c, B - b(t-1)\}, && (4)\\
& \qquad 0 \le b_{\text{d}}(t) + b_{\text{g}}(t) \le \min\{\rho_d, b(t-1)\}, && (5)\\
& \qquad 0 \le b(t) \le B, && (6)\\
& \text{vars.}: \quad \{x(t), x_{\text{d}}(t), x_{\text{b}}(t), b_{\text{d}}(t), b_{\text{g}}(t), b(t)\} \in \mathbb{R}_{\ge 0}.
\end{aligned}
$$

The objective is to minimize the purchased cost from the electric grid, i.e., $p(t)x(t)$, and maximize the revenue from selling back to the grid, i.e., $p(t)b_g(t)$. Constraints (1)-(2) determine the procurement strategy. Constraint (3) dictates the evolution of the storage. By denoting B, ρ_c, and ρ_d as the capacity of storage, charging, and discharging rates, constraints (4)-(6) enforce the capacity and rate limits of the energy storage. Since EP is a linear program, it can be solved efficiently in an offline manner. The real-world practical setting, however, is online, since the future price $p(t)$, the total energy demand $d_{\texttt{tot}}(t)$, and the renewable generation $u(t)$ are not known a priori. As compared to the simplified problem introduced in [32], EP comes with two important generalizations: (1) in EP, we have an additional optimization variable that captures the possibility of selling back to the grid, i.e., $b_{\text{g}}(t)$; (2) in contrast to the problem studied in [32], EP captures a more general pricing model in which the real-time prices can be negative. In our experiments, we evaluate the performance of our proposed solutions as compared to the state-of-the-art algorithms for the special cases of no negative pricing and no selling back to the grid. For the general case, we compare our results with the simple baseline heuristic `PreDay`.

3 OUR MACHINE LEARNING APPROACH

The key challenge in applying a machine learning approach to the energy procurement problem is that an action, such as storing energy in the storage, and its potential reward, such as not having to purchase that energy from the grid at a higher price, can be significantly separated in time, i.e., the reward does not manifest until long after the action is taken. However, by solving our LP formulation EP offline, one can derive the optimal set of actions for any time sequence of demand, renewable generation, and energy prices. *Our main idea is to use supervised learning approach where we train our algorithm to learn from the optimal decisions made by EP.*

Feature Selection. To be able to use supervised learning, we choose a set of features as described below.

Price-related Features. Energy price is a significant determinant of the procurement strategy since we should buy and store energy when the price is low, and discharge and sell when the price is high.

Demand-related Features. Demand is an important factor as our energy procurement strategy must satisfy the demand at every step. We use the *current energy demand* which averages demand over the current 5-minute slot as a feature.

Storage-related Features. The amount of charge in the storage is a key determinant of energy procurement since we are more likely to buy and charge when the storage levels are low, and sell/discharge when the storage levels are high. We use the storage level at the beginning of the time step as a feature.

Time-related Features. Energy prices are set by matching supply and demand and may have diurnal patterns. Further, month of the year is indicative of seasons that could influence renewable generation. Therefore, we use both the *time of day*, with each day divided into 5-minute intervals, and *month of the year* as features.

Learning Algorithms. We present two classes of learning algorithms for energy procurement: learning how much to *buy*, and learning how much to *store*. We describe both approaches.

1) Learning to Buy Energy . Our algorithm `LearnBuy` computes the desired buy amount of $\hat{x}(t)$ at time t using the learned model that uses the features described earlier. Depending on whether $\hat{x}(t)$ is positive or negative we do the following.

(1) If $\hat{x}(t) \geq 0$, we need to buy energy from the grid. We compute the maximum amount that can bought from the grid: $MAX = d(t) + \min\{B - b(t-1), \rho_c\}$. We also compute the minimum amount that should be bought from the grid: $MIN = d(t) - \min\{b(t-1), \rho_d\}$. If $\hat{x}(t)$ is within the acceptable range, i.e., $MIN \leq \hat{x}(t) \leq MAX$, we buy $\hat{x}(t)$ from the grid. Otherwise, we buy $\min\{\max\{\hat{x}(t), MIN\}, MAX\}$ from the grid.
(2) If $\hat{x}(t) < 0$, we need to sell energy to the grid. The most you can draw from the battery for a sale is $MAX = \min\{b(t-1), \rho_d\}$, accounting for the maximum discharge rate and the amount in the battery. We sell $\min\{|\hat{x}(t)|, MAX\}$ to the grid.

2) Learning to Store Energy . `LearnStore` computes the desired storage level $\hat{b}(t)$ at time t using the learned model that uses the above features. Based on the value of $\hat{b}(t)$, it does the following.

(1) If $\hat{b}(t) \geq b(t-1)$, an additional amount of energy must be stored in the battery. The amount that the battery can be charged is $C = \min\{\hat{b}(t) - b(t-1), \rho_c\}$, accounting for the maximum charge rate. The algorithm buys the amount of $x(t) = d(t) + C$ from the grid that is required to cover both the demand and the increase in the storage level.
(2) If $\hat{b}(t) < b(t-1)$, some amount of energy must be removed from battery. The amount that can be removed from the battery is $D = \min\{b(t-1) - \hat{b}(t), \rho_d\}$, accounting for maximum discharge amount. We use the $\min\{d(t), D\}$ of the discharged amount to serve the demand, and sell the remaining amount of $D - \min\{d(t), D\}$ to the grid. Any demand that is not yet satisfied is served from the grid.

In the case that selling to the grid is not an option, the selling is skipped in step 2 and that amount is left in the battery. Note that the actual amount $b(t)$ at time t could be larger than the desired learned amount $\hat{b}(t)$ when restricted by the maximum discharge amount or when selling is not option.

Learning Models. For both `LearnBuy` and `LearnStore`, we experimented with different learning models, in particular, decision trees (DT), k-nearest neighbors (KNN), linear regression (LR), and deep neural networks (DNN). For DT, KNN, and LR, we used scikit-learn [26], which is a Python library containing standard implementations of the most common machine learning models. For DNN, we used TensorFlow [6] with Keras [11] on top. TensorFlow is a library for implementing neural networks, and Keras is a high level library that can use TensorFlow as the back-end to make neural network implementation easier. For the DNN model, we tested with a different number of layers, but we went with 4 feed-forward dense layers as it results in the best performance for our model. The loss function used is the mean squared error, and we used the initialization proposed by He et al. [16] for the initial weights of the neurons. We used ReLU as the activation function for all the layers, and the RMSprop optimizer with a learning rate of 0.001. For all the other models, we used the default parameters from scikit-learn.

4 EXPERIMENTAL EVALUATION

Data Traces. The energy consumption is gathered from Akamai data centers for a 1 month period, at 5 minute intervals, which is based on real-time settlement intervals of the US electricity markets. The Akamai data traces contain the workload of the server clusters at each time slot, hence, to obtain energy usage as a function of load, we use the standard linear model [7]. For our experiments, we report results for an Akamai data center in New York City and electricity prices from NYISO [24] for New York for 2016-2018.

In addition to the total energy usage of data center, we assume that at each location there is on-site renewable installation with 30% penetration, i.e., if renewable generates at its maximum capacity, it can satisfy 30% of the peak energy demand. We collected the renewable generation values from Eastern and Western data sets [4].

Training Methodology. The key idea of our training is to *learn from the optimal*. Our training inputs consist of the real-time energy prices, data center energy demand, and renewable data traces from wind. We derive the optimal decisions for the training inputs by solving EP. We run the offline optimal solution `OPT` on the input traces to derive the optimal amounts to buy or sell and the optimal battery level at that time step. We use two-thirds of the data (2016 and 2017) to train our learning models, and we report our results by testing on the remaining one-third (2018).

Evaluation. In all experiments, we evaluate our algorithms based on the *normalized cost* that they achieve, where the normalized cost is defined as the cost of our (online) algorithm divided by the optimal offline cost. The offline optimal cost is calculated using Gurobi optimization software [15] to solve EP by providing it the *entire* input to the problem. Thus, the offline optimal cost is a lower bound on the cost of any online algorithm. Moreover, no online algorithm may be able to achieve the optimal online cost. However, normalized cost is still a very useful way of viewing the costs in our setting.

We consider the following four different experimental scenarios:

(G5) General case where $\leq 5\%$ of the storage can be charged or discharged at each time step, i.e., $\rho_c = \rho_d = B/20$.

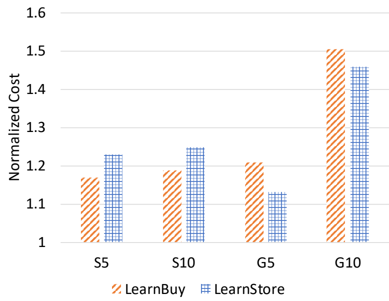


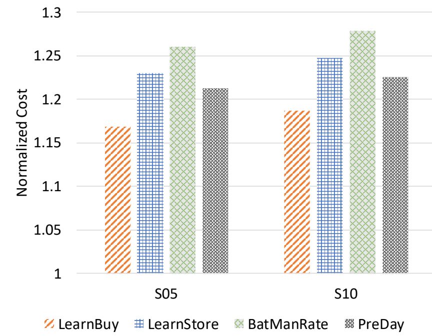
Figure 2: Comparing LearnBuy and LearnStore

- **(G10)** General case where $\leq 10\%$ of the storage can be charged or discharged at each time step, i.e., $\rho_c = \rho_d = B/10$.
- **(S5)** Special case of the procurement problem where $\leq 5\%$ of the storage can be charged or discharged at each time step, no energy can be sold to the grid, and no negative pricing.
- **(S10)** Special case of the procurement problem where $\leq 10\%$ of the storage can be charged or discharged at each time step, no energy can be sold to the grid, and no negative pricing.

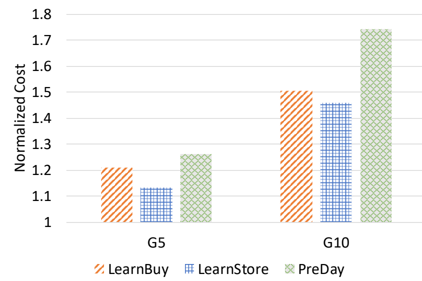
Note that BatManRate [32], works in the special case of just buying from the market without negative pricing. Hence, we construct S5 and S10 to enable comparison with the existing approach. To the best of our knowledge there is no existing work that tackles the general EP problem. Hence, we compare the results of our learning algorithms with the optimal offline values, and also PreDay that is a simple data-driven heuristic that uses the optimal values of the EP problem for the previous day to make energy procurement decisions for the current day. The PreDay algorithm needs an additional feasibility check since there is no guarantee that the optimal values for the previous day provide a feasible solution for the current day.

Comparing the learning algorithms. We compare the two natural learning approaches of learning to buy and learning to store. As shown in Figure 2, LearnStore does better than LearnBuy in the general procurement scenarios, but worse in the special scenarios.

Comparing our approach with prior art. Next, we compare LearnBuy and LearnStore with BatManRate that has the optimal theoretical competitive ratio for the special case scenario and is the best known algorithm for this scenario. We also compare it with the heuristic PreDay that is applicable in both scenarios. Note that there is no known online algorithm with a provable optimal competitive ratio for the general scenario. So, in the general scenario, we only compare the results with PreDay. As reported in Figure 3(a), LearnBuy outperforms PreDay and BatManRate by 3.67% and 7.26% (resp. 3.20% and 7.18%) in S5 (resp. S10). In the general scenario, LearnStore outperforms PreDay by 10.34% (resp. 16.18%) in G5 (resp. G10).



(a) Special Scenario



(b) General Scenario

Figure 3: Comparing the learning approach with prior art.

5 RELATED WORK

Similar problems have been studied in literature using empirical evaluations [13, 25, 34, 35]. These approaches require exact modeling and extensive training data, which is difficult to obtain due to multidimensional uncertainty in the problem. Another direction is stochastic optimization approaches [30], which propose optimal policies given the probabilistic modeling of uncertain inputs. Deviations from the stochastic models may severely degrade the overall performance. Lyapunov optimization [14, 17, 29] obtains optimal control policies over infinite horizon with i.i.d. assumptions for the inputs. Another approach is to leverage competitive design [9] for energy procurement algorithms [10, 32] The goal is to achieve a bounded competitive ratio for the worst-case input, which might be too conservative in reality. All of the above approaches design simple and fixed decision rules based on simplified models and mathematical modeling of the underlying problem. To the best of our knowledge, this work is the first that tackles the general problem of energy procurement for data centers without relying on any fixed decision making rules. There are several recent studies to apply different machine learning approaches in the different areas of energy efficiency and optimization in data centers [12, 19] and more broadly in energy systems [31]. Last, we note that the idea of learning from optimal has been recently used in other application domain such as caching [8]. Our work pursues a similar paradigm of learning form optimal, but for a different application scenario.

6 CONCLUSION

In this paper, we proposed learning-based algorithms for optimizing the energy procurement for data centers with on-site energy storage systems. We proposed and evaluated two algorithms, LearnBuy and LearnStore, that use the approach of learning from the optimal. Using extensive real-world data traces, we show that our algorithms achieve near-optimal cost and are significantly better than the state-of-the-art algorithms that rely on developing fixed rule policies.

Acknowledgments. This work is supported in part by NSF grants CNS-1413998 and CNS-1763617 and a Google Faculty Research Award.

REFERENCES

[1] 2013. Google Data Center in Changhua County, Taiwan. available at https://www.google.com/about/datacenters/inside/locations/changhua-county/. Accessed Feb 2019.

[2] 2015. Tesla's Powerpack proposes battery power for data centers. available at https://www.datacenterdynamics.com/analysis/teslas-powerpack-proposes-battery-power-for-data-centers/. Accessed Feb 2019.

[3] 2016. AMAZON SOLAR FARM US EAST. available at https://communityenergysolar.com/project/amazon-solar/. Accessed Feb 2019.

[4] 2018. Eastern and Western Data Sets. available at https://www.nrel.gov/grid/eastern-western-wind-data.html. Accessed Feb 2019.

[5] 2018. Google Data Center in St. Ghislain, Belgium. available at https://www.google.com/about/datacenters/inside/locations/st-ghislain/. Accessed Feb 2019.

[6] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.

[7] Luiz André Barroso and Urs Hölzle. 2007. The case for energy-proportional computing. *Computer* 12 (2007), 33–37.

[8] Daniel S Berger. 2018. Towards Lightweight and Robust Machine Learning for CDN Caching. In *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*. ACM, 134–140.

[9] A. Borodin and R El-Yaniv. 1998. *Online computation and competitive analysis*. Cambridge University Press.

[10] Chi-Kin Chau, Guanglin Zhang, and Minghua Chen. 2016. Cost minimizing online algorithms for energy storage management with worst-case guarantee. *IEEE Transactions on Smart Grid* 7, 6 (2016), 2691–2702.

[11] François Chollet et al. 2015. Keras. https://keras.io.

[12] Richard Evans and Jim Gao. 2016. DeepMind AI Reduces Google Data Centre Cooling Bill by 40%. *https://deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-40/* (2016).

[13] Sriram Govindan, Di Wang, Anand Sivasubramaniam, and Bhuvan Urgaonkar. 2013. Aggressive Datacenter Power Provisioning with Batteries. *ACM Transactions on Computing Systems* 31, 1 (2013), 2:1–2:31.

[14] Yuanxiong Guo and Yuguang Fang. 2013. Electricity cost saving strategy in data centers by using energy storage. *IEEE Transactions Parallel and Distributed Systems* 24, 6 (2013), 1149–1160.

[15] LLC Gurobi Optimization. 2018. Gurobi Optimizer Reference Manual. http://www.gurobi.com

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. 1026–1034.

[17] Longbo Huang, Jean Walrand, and Kannan Ramchandran. 2012. Optimal demand response with energy storage management. In *Proc. IEEE SmartGridComm*. 61–66.

[18] Anthony Kim, Vahid Liaghat, Junjie Qin, and Amin Saberi. 2016. Online Energy Storage Management: an Algorithmic Approach. In *Proc. APPROX*.

[19] Nevena Lazic, Craig Boutilier, Tyler Lu, Eehern Wong, Binz Roy, MK Ryu, and Greg Imwalle. 2018. Data center cooling using model-predictive control. In *Advances in Neural Information Processing Systems*. 3818–3827.

[20] Minghong Lin, Adam Wierman, Lachlan LH Andrew, and Eno Thereska. 2013. Dynamic right-sizing for power-proportional data centers. *IEEE/ACM Transactions on Networking* 21, 5 (2013), 1378–1391.

[21] Zhenhua Liu, Yuan Chen, Cullen Bash, Adam Wierman, Daniel Gmach, Zhikui Wang, Manish Marwah, and Chris Hyser. 2012. Renewable and cooling aware workload management for sustainable data centers. In *ACM SIGMETRICS Performance Evaluation Review*, Vol. 40. 175–186.

[22] Zhenhua Liu, Minghong Lin, Adam Wierman, Steven H Low, and Lachlan LH Andrew. 2011. Greening geographical load balancing. In *Proc. ACM SIGMETRICS*.

[23] Erik Nygren, Ramesh K Sitaraman, and Jennifer Sun. 2010. The akamai network: a platform for high-performance internet applications. *ACM SIGOPS Operating Systems Review* 44, 3 (2010), 2–19.

[24] NYISO [n.d.]. NYISO Electricity Market. available at http://www.nyiso.com.

[25] Darshan S Palasamudram, Ramesh K Sitaraman, Bhuvan Urgaonkar, and Rahul Urgaonkar. 2012. Using batteries to reduce the power costs of internet-scale distributed networks. In *Proc. of ACM SoCC*.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[27] Asfandyar Qureshi, Rick Weber, Hari Balakrishnan, John Guttag, and Bruce Maggs. 2009. Cutting the Electric Bill for Internet-scale Systems. In *Proc. ACM SIGCOMM*.

[28] L. Rao, X. Liu, and W. Liu. 2010. Minimizing electricity cost: Optimization of distributed Internet data centers in a multi-electricity-market environment. In *Proc. IEEE INFOCOM*.

[29] R. Urgaonkar, B. Urgaonkar, M.J. Neely, and A. Sivasubramaniam. 2011. Optimal power cost management using stored energy in data centers. In *Proc. ACM SIGMETRICS*.

[30] Peter M van de Ven, Nidhi Hegde, Laurent Massoulié, and Theodoros Salonidis. 2013. Optimal control of end-user energy storage. *IEEE Transactions on Smart Grid* 4, 2 (2013), 789–797.

[31] Hao Wang and Baosen Zhang. 2018. Energy storage arbitrage in real-time markets via reinforcement learning. In *2018 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 1–5.

[32] Lin Yang, Mohammad H Hajiesmaili, Ramesh Sitaraman, Enrique Mallada, Wing S Wong, and Adam Wierman. 2019. Online Inventory Management with Application to Energy Procurement in Data Centers. *arXiv preprint arXiv:1901.04372* (2019).

[33] L Yang, M H Hajiesmaili, H Yi, and M Chen. 2017. Hour-Ahead Offering Strategies in Electricity Market for Power Producers with Storage and Intermittent Supply. In *Proc. ACM SIGMETRICS*. 21–22.

[34] Wenli Zheng, Kai Ma, and Xiaorui Wang. 2014. Exploiting thermal energy storage to reduce data center capital and operating expenses. In *Proc. IEEE HPCA*. 132–141.

[35] Haihang Zhou, Jianguo Yao, Haibing Guan, and Xue Liu. 2015. Comprehensive understanding of operation cost reduction using energy storage for IDCs. In *Proc. IEEE INFOCOM*. 2623–2631.