Power Modeling for Effective Datacenter Planning and Compute Management

Ana Radovanovic, Bokan Chen, Saurav Talukdar, Binz Roy, Alexandre Duarte, and Mahya Shahbazi, *Member, IEEE*

***Abstract*—Over the past decade, there has been a global growth in datacenter capacity, power consumption and the associated costs. Accurate mapping of datacenter resource usage (CPU, RAM, etc.) and hardware configurations (servers, accelerators, etc.) to its power consumption is necessary for efficient long-term infrastructure planning and real-time compute load management. This paper presents two types of statistical power models that relate CPU usage of Google's Power Distribution Units (PDUs, commonly referred to as power domains) to their power consumption. The models are deployed in production and are used for cost- and carbon-aware load management, power provisioning and infrastructure rightsizing. They are simple, interpretable and exhibit uniformly high prediction accuracy in modeling power domains with large diversity of hardware configurations and workload types across Google fleet. A multi-year validation of the deployed models demonstrate that they can predict power with less than 5% Mean Absolute Percent Error (MAPE) for more than 95% diverse PDUs across Google fleet. This performance matches the best reported accuracies coming from studies that focus on specific workload types, hardware platforms and, typically, more complex statistical models.**

***Index Terms*—Datacenter power modeling, statistical power models, datacenter power efficiency.**

NOMENCLATURE

r	The index indicating PDU's operating regime.
d	The index indicating day.
$\mathcal{T}$	The set of 288 5-minute time periods within a day.
$\mathcal{C}$	The set of PDUs in a cluster.
cpu_{-}^{PDU}	Minimum PDU-level CPU usage.
cpu_{+}^{PDU}	Maximum PDU-level CPU usage.
$\hat{\cdot}$	The predicted value of a variable.
$\bar{\cdot}$	The average value of a variable.
u_{CPU}^{PDU}	PDU-level CPU usage.
λ	Segment length in the Per-PDU model.
α^r	Intercept of the Per-PDU model in operation regime r.
β^r	Slope of the Per-PDU model in operating regime r.
u_{POW}^{PDU}	PDU power consumption.
u_{CPU}^{m}	CPU usage of machine m.
u_{POW}^{m}	Power consumption of machine m.
C^m	Hardware configuration of machine m.
P_{-}^{m}	Idle power of machine m.
P_{+}^{m}	Maximum power usage of machine m.
l_d^m	Binary label indicating whether machine m is dedicated to a specific workload.
$u_{POW}^{M,PDU}$	Power usage of all machines in PDU.
$u_{POW}^{O,PDU}$	The overhead power of PDU.
N_{F_i}	Platform count of platform family F_i.
$u_{CPU}^{F_i}$	CPU usage of platform family F_i.
P_{N}^{PDU}	Maximum networking power in PDU.
P_{C}^{PDU}	Maximum cooling power in PDU.
A^{PDU}	Power architecture of PDU.
P_{-}^{PDU}	Total idle power of all machines in PDU.
P_{+}^{PDU}	Total maximum power of all machines in PDU.

Manuscript received January 13, 2021; revised June 8, 2021 and October 12, 2021; accepted October 19, 2021. Date of publication November 3, 2021; date of current version February 21, 2022. This work was supported by the Google. Paper no. TSG-00066-2021. *(Corresponding author: Bokan Chen.)*

The authors are with the Department of Technical Infrastructure, Google Inc., Mountain View, CA 94043 USA (e-mail: anaradovanovic@google.com; bokanchen@google.com; stalukdar@google.com; binzroy@google.com; alexandredu@google.com; mahyash@google.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSG.2021.3125275.

Digital Object Identifier 10.1109/TSG.2021.3125275

I. INTRODUCTION

DATACENTERS are warehouse-sized computing systems that operate around the clock to support large-scale Internet services worldwide, while enabling fast growth of the IT industry and transformation of the economy. The demand for computing resources and datacenter power worldwide has been continuously growing, contributing to approximately 1% of the total electricity usage [1]. Given the rapid growth of datacenter workload globally (more than sixfold in a decade) [1], novel methodologies for improving datacenter power and energy efficiency can have a considerable economic, environmental and performance impact [2]. The focus of this paper is on the development of power models that predict power consumption of power domains in Google's datacenters as a function of their resource usage and hardware characteristics, which are used for efficient datacenter planning and compute load management.

There exists a body of literature focused on provisioning power consumption at component, circuit, server, PDU[1] (also referred to as power domain) and datacenter levels [2]. While component (e.g., processor) level power models are typically

[1]PDU stands for Power Distribution Unit, multiple of which comprise a datacenter campus. PDUs are metered and encapsulate all servers, networking equipment and data storage.

designed to capture relationships between control signals and component states at different time scales [2]–[5], they are usually designed for specific types of hardware and are hard to use for power provisioning in hyperscale datacenters that are highly heterogeneous in workload types, platform families, and platform level control mechanisms [6]. Statistical models (sometimes referred to as software-based models) [2], [7] have proven effective in modeling either individual subsystems of a server such as CPU, memory, disk and network, or a virtual machine, server, server cluster [8], or whole datacenters.

In this paper, we discuss two types of statistical power models deployed at Google which relate a power domain's CPU usage to its power consumption at 5-minute granularity. The first type, named Per-PDU model, is piecewise-linear in CPU usage and retrained daily for each PDU. The second type, named Unified model, is used to 1) predict a single machine power, which is then aggregated to obtain the PDU power estimate (Unified Machine Model), or 2) directly predict PDU power (Unified PDU Model). The Unified models utilize Random Forest regression to capture nonlinear dependence between machine/PDU power consumption, CPU usage, machine/PDU hardware characteristics, and machine-workload sharing properties. Both types of models implicitly incorporate the effects of task scheduling and CPU/voltage frequency scaling.

Each model shows advantages in certain use cases. For example, the Per-PDU model's predictive performance across all CPU/power usage regimes and its piecewise-linearity is exploited within Google's Carbon-Intelligent Computing System [9] to evaluate the impact of changes in datacenter clusters' CPU usage on their power consumption as a result of shifting workloads in time and across locations. The Per-PDU model has also been used for estimation and real-life validation of a datacenter's potential to reduce power consumption during grid-level demand response events. On the other hand, the Unified Machine Model is used to set CPU utilization limits and ensure that the power capping [10] thresholds are not exceeded. Our evaluation has shown that the Unified models are best suited for quantifying the impact of large perturbations in compute usage due to load migrations, maintenance, or infrastructure upgrades, as well as long-term capacity planning for future datacenters.

The previous studies on statistical power models primarily focused on a single machine or a group of machines, and have been limited in the number of machine configurations, workload types and targeted use cases ([7] and references therein). Modeling approaches involving a single server and a few-server models for specific platforms and workloads (e.g., streaming media, synthetic workloads, SPECpower database applications, etc.) report a wide range of prediction errors, from 2% to 26% [7]. At a datacenter level, modeling of power consumption has mainly been addressed via stylized models [11]–[13] with little or no performance validation. The best previously reported prediction error for a cluster ([8]) comprised of thousands of servers serving Webmail and Websearch workloads, as well as data processing pipelines using Mapreduce [14] is 1% [10]. This model was previously used at Google to estimate PDU power consumption, but has been substituted by the new generation of models discussed in this paper. We refer to it as the "benchmark model", and quantitatively demonstrate the superior performance of the new models by conducting validations across all PDUs within the Google datacenter fleet (Section III).

To summarize, the key contributions of the discussed work, when compared to the state-of-the-art, include:

- **Use-case-driven design:** The proposed power model design is driven by a wide range of targeted use cases, which include: 1) real-time estimation of datacenter power consumption and its electricity-based carbon footprint [15], [16]; 2) near-term (intraday and day-ahead) cost-and-carbon-aware workload management (including software-controlled power capping [10] and grid-level demand response [15], [17]–[19]); and 3) power provisioning and rightsizing of future machine upgrades given monthly, quarterly, or yearly projected resource usage trends [10].
- **Largest scale validation:** To the best of our knowledge, the paper discusses the largest-scope power modeling system deployed for their training and validation, which spans all power domains across the Google datacenter fleet with their heterogeneous hardware configurations, workload types and resource utilization regimes.
- **High accuracy:** A multi-year validation of the discussed models has consistently demonstrated high prediction accuracy throughout the dynamic range of power domains' utilizations. While the deployed models use only basic hardware and CPU usage characteristics, the predictive performance matches the best reported results in the related literature [10], [20], [21].

The rest of the paper is organized as follows. Section II describes the overall power architecture of a datacenter, discusses the deployed system used for daily training and validation, and reviews the Per-PDU and Unified models in more detail. The performance of the proposed models for different use cases is discussed in Section III. Finally, Section IV concludes the paper.

II. Datacenter Infrastructure and Power Modeling

In this section, we 1) describe a typical Google datacenter power architecture, 2) briefly discuss its workload classification and management, 3) introduce a system used for data collection, training and validation of the deployed power models, and 4) include a detailed overview of the two types of power models, with a brief discussion on their complexity and scalability.

A. Power Architecture

Figure 1 shows a simplified view of power architecture of a typical Google datacenter [22]. Every datacenter is connected to the electric grid via several medium-voltage feeders. Each medium-voltage distribution line is transformed to supply low-voltage PDUs. PDUs are typically connected to multiple BUS ducts. The BUS ducts supply power to the IT equipment (i.e., compute, storage and networking racks), fan coils (used for

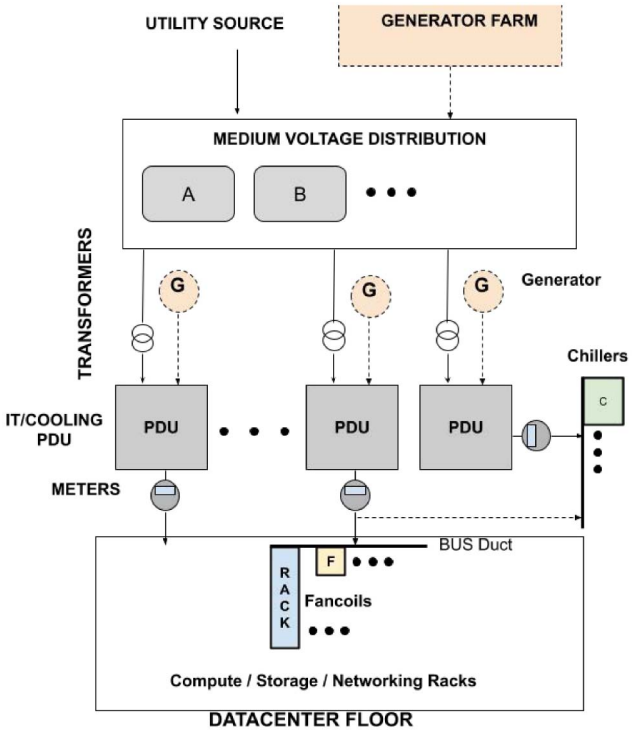

Fig. 1. A simplified representation of datacenter power distribution hierarchy.

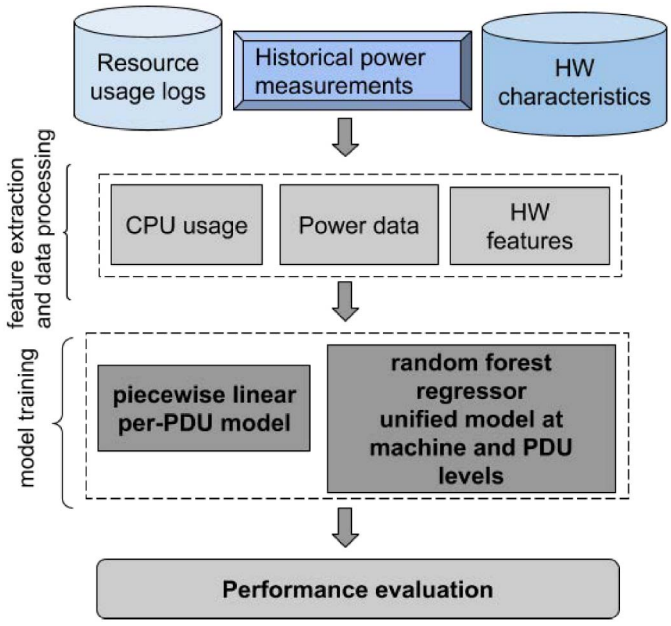

Fig. 2. Data processing and training pipeline.

cooling) on the datacenter floor and, in some cases, off-floor cooling equipment (e.g., chillers).

A single PDU typically has a few thousand machines and a handful of PDUs comprise a cluster. The PDUs in each cluster belong to a single job-scheduling domain, i.e., a common real-time scheduler that assigns computing tasks to its feasible machines. Generator backup is available to keep the datacenter running in the event of power outage. Depending on the architecture, the PDUs in a cluster are either connected to a separate backup generator or to a common backup generator that supports the medium-voltage line.

All PDUs are metered and provide power measurements which are used to train and validate the models discussed in the subsequent sections. It is observed that datacenter power can be accurately expressed as the sum of its individual PDU measurements inflated by a few percent overhead to account for unmetered auxiliary loads such as office HVAC.

B. Datacenter Workloads and Their Management

Machines at Google are set up to run any application, connected via high bandwidth switches within a campus and via a global backbone network connecting datacenters. Datacenter hardware is controlled and administered by specialized software that can handle massive scale. To the extent possible, hardware controls, job scheduling, etc., are abstracted away from users.

Compute jobs at Google are managed by a distributed cluster-level operating system (known as Borg [23]). These jobs can be roughly split into two categories: 1) indefinitely running servers, and 2) batch processes (e.g., data processing pipelines using MapReduce or Flume [14], [24]). Jobs can consist of several tasks (sometimes thousands), both for reasons of reliability and because a single process can't usually handle all traffic. The cluster operating system is responsible for task allocation across machines within a cluster, which includes starting a job, finding machines for its tasks (i.e., task scheduling), allocating requested resources (CPU/RAM/disk) on machines, and instructing the machines to start executing the tasks. A machine is either dedicated to a specific workload (e.g., Search), or shared among various workloads.

The PDU-level CPU usage of serving jobs (category 1) in Google datacenters have predictable daily patterns. On the other hand, the total PDU-level CPU usage of batch processes has considerable intraday variability and their intraday profiles are hard to accurately predict [9]. As a result, the total PDU-level CPU usage typically varies significantly within a day with fluctuations larger than 10% for more than half of the PDUs across the Google fleet.

C. Power Modeling Pipeline

The power modeling pipeline is used for data extraction, processing, training and validation of two types of power models, Per-PDU and Unified, that map CPU usage to power consumption at 5-minute granularity.

The pipeline consists of components with specific functionality such as (Figure 2):

1) Collection of usage and machine/PDU hardware configuration data,
2) Data processing for detection of outliers and their removal,
3) Power models training, and
4) Performance evaluation.

The collected usage data includes: 1) power consumption and CPU usage of every single machine across the fleet at 5-minute granularity, collected through built-in power sensors, reported by the machine OS, and logged and maintained

in a Google-internal database; 2) PDU power at 5-minute granularity, collected by power meters for each PDU and stored in a separate Google-internal database. The gathered configuration data includes both machine and PDU configuration information as discussed in more detail in Section II-D2.

The data processing is conducted before the power models are trained with the goal to automatically detect and remove outliers from the data set. It has been observed that the outliers mainly correspond to atypical regimes in CPU usage and power measurements caused by rare maintenance events, power meter malfunctions, losses in stored data, etc. Rare and short-term losses in the collected usage data are handled using linear interpolation of the time series. Other heuristics used to automatically detect and remove outliers include:

- Smoothing using Exponentially Weighted Moving Average (EWMA) in cases where CPU usage changes more than 30% from one 5-minute interval to the next (which is rare);
- Removing data instances for which changes in power measurements cannot physically match the corresponding change in CPU usage from one 5-minute interval to the next. For example, data instances with $\Delta|\text{power usage}| > 20\Delta|\text{CPU usage}|$ for any two consecutive 5-minute measurements are excluded. This type of anomaly is typically caused by erroneous power meter readings;
- Filtering out data instances with PDU power measurements smaller than 80% of its daily median. It is observed that these power drops are extremely rare and coincide with 1-2 hour-long maintenance events.

The power models are trained after the data processing step.

D. Power Models

As discussed above, PDUs contain servers, cooling and networking equipment. The telemetry system available in each PDU enables collection of power consumption data at 5-minute granularity and, therefore, a supervised learning framework for modeling PDU power consumption.

While datacenter infrastructure planning and real-time workload management are typically driven by trends in workload resource usage (CPU, RAM, disk usage), the analysis in this paper demonstrates that PDU power consumption can be accurately estimated using only its CPU usage. This conclusion holds irrespective of the diversity in machine types, e.g., compute, storage, accelerators (TPU or GPU), etc., which might appear surprising. However, it is well known that storage machines have a narrow dynamic range of power usage [2], which translates into a small impact on power fluctuations when aggregated at a PDU level. Our empirical analysis demonstrates similar conclusions with regards to accelerators. The proposed models demonstrate high power prediction accuracy irrespective of PDUs' diversity in machine types and workloads.

The following subsections discuss the Per-PDU and Unified models in more detail.

1) Per-PDU Power Model: To enable power efficiency and carbon-aware workload management as in [9], we use a light-weight, piecewise-linear Per-PDU model. The piecewise-linearity of the model allows for easier and tractable integration of the power usage sensitivity within the datacenter power-aware load management optimization problems.

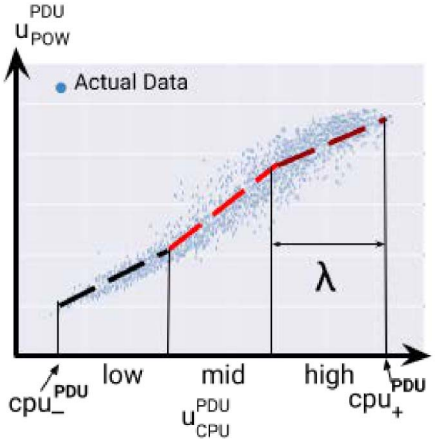

Fig. 3. Structure of the piecewise linear model.

Extensive analysis of 5-minute average PDU power consumption u_{POW}^{PDU}, as a function of its average CPU usage u_{CPU}^{PDU}, indicates three distinct utilization regimes (Figure 3). Furthermore, PDU power consumption monotonically increases with its CPU usage. Based on this observation, the linear models are trained for *low*, *medium* and *high* CPU usage regimes, as defined below:

$$\begin{cases} low, & \text{if } u_{CPU}^{PDU} \le cpu_{-}^{PDU} + \lambda \\ medium, & \text{if } u_{CPU}^{PDU} \in \left[cpu_{-}^{PDU} + \lambda,\ cpu_{-}^{PDU} + 2\lambda\right) \\ high, & \text{if } u_{CPU}^{PDU} \ge cpu_{-}^{PDU} + 2\lambda. \end{cases}$$

PDU minimum and maximum CPU usage, i.e., $cpu_{-}^{PDU} \equiv \min u_{CPU}^{PDU}$ and $cpu_{+}^{PDU} \equiv \max u_{CPU}^{PDU}$ are measured historically. Note that in the previous expressions we omit the reference to time for more clarity. The three segments are assumed to be of equal lengths, defined as $\lambda = \frac{cpu_{+}^{PDU} - cpu_{-}^{PDU}}{3}$. Piecewise-linear models in the context of machine and cluster power estimation were previously proposed in [21], where the Multivariate Adaptive Regression Splines (MARS) [25] was used to automatically learn the regime switching points (called knots). The Per-PDU model is defined using the three equal-width segments corresponding to low, medium and high usage regimes, while achieving continuity (i.e., change in PDU power as a function of a change in its CPU usage is bounded by the platform-aware constant), monotonicity (i.e., PDU power is monotonically increasing in its CPU usage) and desirable accuracy for the use cases of interest.

For each PDU, linear models are trained for each usage regime to estimate the power consumption of a given PDU as

$$\hat{u}_{POW}^{PDU} = \alpha^r + \beta^r u_{CPU}^{PDU}. \tag{1}$$

The intercept and slope corresponding to each regime, i.e., α^r, β^r, $r \in \{low, medium, high\}$ are computed so that the weighted sum of squared errors is minimized. The model parameters in (1) are constant for each day and updated daily. The daily training uses the most recent 7 days of PDU power

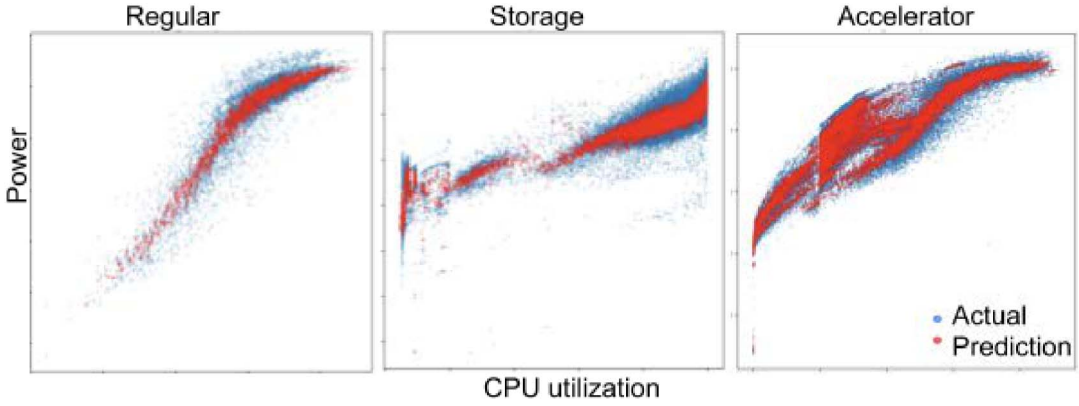

Fig. 4. Nonlinear relationship between machine power and CPU utilization for compute, storage and accelerator machine types. Actual power is presented in blue, while the Unified Machine Model's predictions are in red.

and CPU usage data and is done to adapt the model in case of hardware changes (deployment/decommission of servers). In addition, the training instances are weighted based on their recency, i.e., higher weights are assigned to more recent measurements to ensure proper adaptation to systematic changes in the fleet. In particular, the pipeline uses $\frac{1}{1+d}$ to weigh data instances from d days ago.

The predictive performance of the Per-PDU model is evaluated daily using power/CPU usage data from the next day (Section III). It is observed that the Per-PDU model can adapt to sporadic changes in workload properties and hardware infrastructure in $\sim$ 2 days. Outside of these transient regimes, the model parameters typically change very slowly in time. Note that the Per-PDU model implicitly incorporates effects of real-time scheduling and frequency scaling within a PDU.

In addition to the efficiency-aware workload management, the Per-PDU model is used for monitoring and near-term predictions of Google datacenter power usage and electricity-based carbon footprint. These models are particularly useful when power measurements are delayed or power meters are broken.

2) Unified Models: In this section, we discuss Unified Machine and PDU models that are useful for long-term infrastructure planning. In this scenario, the models are used to predict the power consumption of future PDUs (e.g., after adding some machine capacity) that operate at a given CPU utilization. The Per-PDU model is not applicable to this use case since it implicitly assumes specific PDU hardware configuration and workload types. In order to effectively plan for power consumption of existing or future PDUs with changing server compositions, we need power models that can be generalized across PDUs and are able to incorporate knowledge about server deployment plans.

For the Unified Machine Model, a PDU is a collection of machine types, while for the Unified PDU Model, a PDU is a collection of platform families. Both models can be trained using data from one PDU to predict power usage of a different PDU (Section III).

Random Forest [26] is used to model nonlinear relationships between the features (CPU usage/utilization, machine/PDU hardware characteristics, and machine-workload sharing properties) and the power consumption of a machine (Unified Machine Model) or PDU (Unified PDU Model) across the Google fleet. Random Forest regression has been widely used to learn nonlinear relationships among variables in various research areas [27]. In particular, it can handle high dimensional data with both continuous and discrete variables. The abundance of power consumption and CPU usage data for all machines and PDUs across the Google datacenter fleet makes this class of models good candidates for predicting their relationships.

To train the Random Forest regressor, the Sklearn [28] package is used with the default values for most of the hyperparameters and the Mean Squared Error as the minimization objective. However, some hyperparameters such as number of trees, maximum tree depth and minimum number of samples in each leaf node were manually tuned so that the training and validation errors attain convergence within the selected number of samples and show no further improvement.

Unified Machine Model estimates power consumption of each machine deployed in a PDU, which are then added up to predict the PDU power. To predict machine-level power consumption, the industry has been mainly using the approach in [10], which interpolates power usage based on a straight line connecting machine's idle power (corresponding to 0 CPU usage) and maximum power (corresponding to the maximum CPU usage). However, in reality, the relationship between machine's CPU usage and its power consumption is nonlinear as shown in Figure 4. The Unified Machine Model captures the nonlinear relationship (Figure 4) for each machine type within the Google datacenter fleet.

There are three machine types: compute, storage, and machines with accelerators (TPUs or GPUs). For all three types, CPU utilization is a strong predictor of their power usage, as shown in Figure 4. Moreover, as shown in Figure 4, the relationship between CPU and power usage varies for various machine types. Thus, machines' hardware characteristics affect their power usage as well and are included as features into the model.

The Unified Machine Model is trained using the following features:

- C^m: machine m's motherboard family (i.e., configuration) comprising CPU type and size, memory quantity × size, SSD quantity × size, disk quantity × size, etc. (e.g., Xeon machine with n_{CPU} CPUs, $n_{RAM} \times 32GB$ memory, and $n_{disk} \times 1TB$ disk). Each machine configuration has a unique code in the format of a 5 digit number. Machine configuration is a discrete feature, converted into a series of binary features using the one-hot encoding approach [29]. Additional feature selection is used to remove the features that do not impact the model's predictive performance.
- P^m_-, P^m_+: idle and maximum power usage of machine m with a given configuration.
- l^m_d: binary feature indicating whether machine m is dedicated to a specific application (e.g., Search). If equal to 0, this means that machine m is shared by various applications (i.e., workloads).
- u^m_{CPU}: CPU usage of machine m.

The objective is to develop a mapping $f(.)$ between the above-described features and power consumption of machine m, expressed as $\hat{u}^m_{POW} = f(C^m, P^m_-, P^m_+, l^m_d, u^m_{CPU})$.

The model training uses a week of Google fleetwide machine usage and configuration data, which is then randomly sampled to keep 30000 5-minute data instances that capture all machine configurations and machine-workload sharing settings. Furthermore, to ensure high prediction accuracy along the full CPU/power usage operating regime of each machine family, the stratified sampling [30] is applied, i.e., machine's CPU utilization range is split into 10 equally spaced buckets, after which the data is sampled randomly from each bucket.

The total power consumed by the machines within a given PDU, $u^{M,PDU}_{POW}$, is estimated by summing over machine level power predictions, i.e.,

$$\begin{aligned}\hat{u}^{M,PDU}_{POW} &= \sum_{m \in PDU} \hat{u}^m_{POW} \\ &= \sum_{m \in PDU} f\big(C^m, P^m_-, P^m_+, l^m_d, u^m_{CPU}\big). \end{aligned} \tag{2}$$

Finally, to predict total PDU power, $\hat{u}^{PDU}_{POW}$, the power usage of its networking and cooling equipment, $u^{O,PDU}_{POW} := u^{PDU}_{POW} - u^{M,PDU}_{POW}$, is estimated and added to the total predicted machine power $\hat{u}^{M,PDU}_{POW}$. Since $u^{O,PDU}_{POW}$ exhibits small variations over time and is significantly smaller than PDU total power consumption, we estimate it using its average value from the previous day, $\bar{u}^{O,PDU}_{POW}$.

The high accuracy of the Unified Machine Model (discussed in more detail in Section III) is attributed to: 1) the fact that the model is trained using all machine configurations and operating regimes (from 0% CPU usage to almost 100% CPU usage); 2) the scale of aggregation of machine level power estimates (tens of thousands) to compute PDU power predictions; and 3) more detailed system-related features.

Unified PDU Model estimates PDU power consumption u^{PDU}_{POW}, using PDU-level hardware and CPU usage features as listed below:

- P^{PDU}_-, P^{PDU}_+: sum of the idle and maximum machine powers in PDU.
- N_{F_i}: total number of machines per platform family F_i. For example, all computing machines with Intel CPUs belong to the platform family named Intel. Another category of machines is storage which, depending on the type of storage (e.g., SSD), has a few different platform families. Overall, there are 10 platform families across Google's datacenter fleet, say $F_1, \ldots, F_{10}$, with the corresponding number of machines within a given PDU denoted as $N_{F_1}, \ldots, N_{F_{10}}$, which are used as features.
- $u^{F_i}_{CPU}$: sum of CPU usage of all machines per platform family in a PDU, that is, $u^{F_i}_{CPU} := \sum_{m \in F_i} u^m_{CPU}$, $i = 1, \ldots, 10$, are used as the CPU usage features. Note that each platform family contains several machine configurations.
- P^{PDU}_N: power drawn by the networking equipment within a PDU is not metered separately, and we use its maximum value as a proxy. A small dynamic range of network-related power consumption justifies the approximation (also previously observed in [2], [10]).
- P^{PDU}_C: Similar to P^{PDU}_N, maximum power drawn by the cooling equipment within a PDU is used as a proxy for its power.
- A^{PDU}: categorical feature used to identify the type of PDU power architecture from the three available types, and is one-hot encoded.

The objective is to develop a mapping $g(.)$ between the above described features and PDU power consumption, expressed as

$$\hat{u}^{PDU}_{POW} = g\Big(P^{PDU}_-, P^{PDU}_+, \{N_{F_i}\}^{10}_{i=1}, \{u^{F_i}_{CPU}\}^{10}_{i=1}, P^{PDU}_N, P^{PDU}_C, A^{PDU}\Big). \tag{3}$$

The model is trained using around 1 million instances of PDU-level CPU and power usage measurements at 5-minute granularity, and across all PDUs within the Google fleet.

Both Unified models (Machine and PDU) are trained using the same week of data to provide a consistent baseline for their performance comparison. The prediction accuracy of the Unified PDU Model significantly depends on the distribution of the platform family mix and CPU usage regimes captured by the training data, as discussed in more detail in Section III-B1.

E. Model Complexity

The Per-PDU model is a piecewise linear regression model and its complexity is $O(n)$, with n being the sample size of the training data [31]. The time complexity of the Random Forest regression models is $O(n \log n)$, as discussed in [31].

The Per-PDU models are decoupled and can be trained in parallel for each PDU, and thereby, completed within minutes. The Random Forest regression model also has parallelization built-in and is typically trained in 2 hours. In addition, our validation analysis has shown that the Unified models insignificantly change over time and, consequently, do not require to be retrained often.

The models in this paper typically do not take up a lot of storage space due to the small number of hyperparameters

that they use. Thus, the space complexity is not a constraining factor. The Per-PDU model is usually a few kilobytes (for one PDU) and the Unified models are a few gigabytes.

III. Performance Evaluation Through Targeted Use Cases

The previously discussed power models are rigorously validated for all PDUs across the Google datacenter fleet. Their predictive performance is analyzed while taking into consideration the use cases of interest:

1) Near-term power estimation and, consequently, carbon accounting ([15], [16]);
2) Near-term load shaping, which includes peak power shaving ([10]), load drop experiments (e.g., grid level demand response [17]), as well as carbon- and cost-aware load shifting ([15], [18], [19]);
3) Long-term planning and rightsizing of future datacenter infrastructure, including upgrades and deployments.

For the majority of the use cases listed above, the main goal is to accurately predict PDU power consumption as a function of its CPU usage (or utilization), hardware and machine-workload sharing properties. There are different performance metrics that could be used to quantify the prediction's deviation from the actual power measurements. The metrics should provide a reasonable comparison across various PDUs regardless of their maximum power capacities. To that end, Mean Absolute Percent Error (MAPE) for a day d, is chosen to evaluate performance of the power models, and is defined as

$$MAPE^{PDU}(d) = \frac{100\%}{|\mathcal{T}|} \sum_{t}^{\mathcal{T}} \frac{|\hat{u}^{PDU}_{POW}(t) - u^{PDU}_{POW}(t)|}{u^{PDU}_{POW}(t)}. \quad (4)$$

A. Use Case #1: Near-Term Power Estimation

To test 5-minute power prediction accuracy across all PDUs fleetwide, the deployed models are trained using seven days of historical data, and their performance is evaluated using the actual CPU and power usage, as well as other features collected on the following day.

To compare the performance across all discussed models, a 2-step evaluation analysis is conducted:

1) For a randomly selected week, we compute the average daily MAPE for each PDU and model using:

$$\overline{MAPE}^{PDU} := \frac{1}{7} \sum_{d \in week} MAPE^{PDU}(d). \quad (5)$$

Then, we capture the fraction of PDUs with the average daily MAPE smaller than a given percent value. As shown in Figure 5, all three models have similar performance, with MAPE smaller than 5% for more than 90% PDUs, and MAPE smaller than 10% for more than 99% of PDUs. The uniformly low MAPE across all models makes them good candidates for short-term estimation and forecasting of PDU average power, which is aggregated to obtain cluster-level and campus-level power predictions. Figure 5 also includes the average daily MAPE of the linear model in [10], referred to as the "benchmark" model. The "benchmark" model aggregates the output of the linear model between the idle and maximum power consumption for each machine type within a PDU as a function of its CPU usage. The "benchmark" model has been previously used at Google to predict datacenter power consumption and the figure shows that it is significantly outperformed by the new generation of power models discussed in this paper.

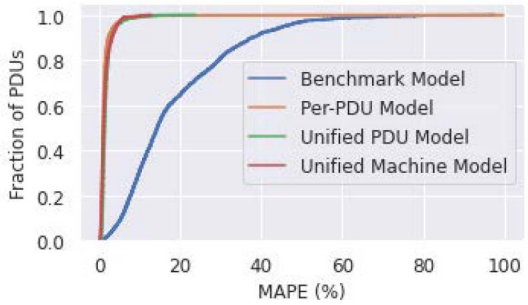

Fig. 5. Fraction of PDUs with average daily MAPE less than a given value.

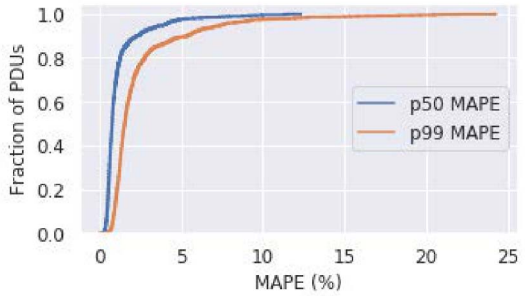

Fig. 6. Fraction of PDUs with the median and 99th percentile of daily Per-PDU's MAPEs less than a given value.

2) To test the uniformity of the evaluated performance across time, daily MAPEs are calculated for each PDU across a year-long time horizon, i.e., $\{MAPE^{PDU}(d)\}_{d \in year}$. Both the Unified PDU and Unified Machine Models are inherently time-invariant since their training data incorporates the full range of values of their features. On the other hand, to evaluate the temporal insensitivity of the Per-PDU model, the 50th and 99th percentiles of $\{MAPE^{PDU}(d)\}_{d \in year}$ are computed for each PDU. We then compute the fraction of PDUs with their daily MAPE median (50th percentile) and 99th percentile smaller than any given percentage value. As shown in Figure 6, the Per-PDU model's overall performance exhibits some performance deviations across time.

B. Use Case #2: Load Shaping

The Per-PDU model's piecewise linearity provides a suitable framework to evaluate the impact of changes in CPU usage on power usage, which can then be easily integrated into cost- and carbon-aware optimizations designed to shape compute load by shifting compute tasks in time and space ([9], [17], [18]), i.e., between datacenter clusters. However, while the discussed models have comparable MAPEs when predicting changes in power usage as a result of small variations in CPU usage, this is not the case when changes in CPU

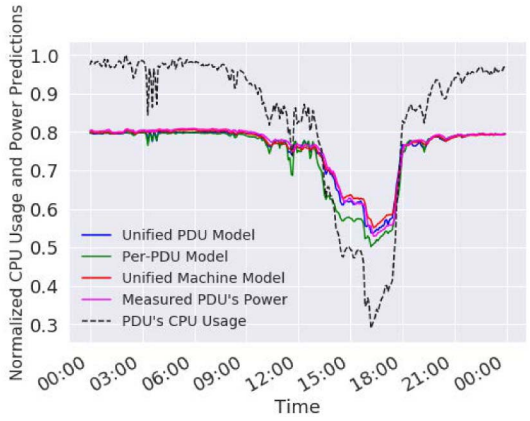

Fig. 7. Example of a substantial load drop in one of the experimental PDUs.

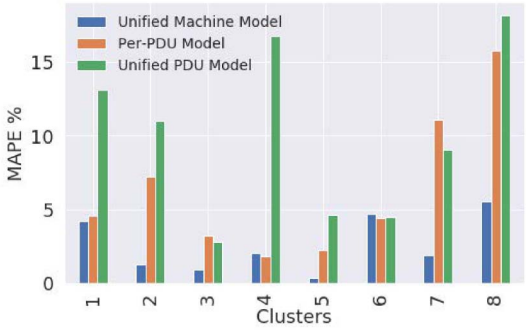

Fig. 8. Cluster level performance of the proposed models for eight experimental clusters during the planned power drop event.

usage are more substantial, which typically happens during datacenter maintenance events, or grid-level demand response events.

1) Power Drop Experiments: There are several applications in which a model's capability to accurately extrapolate power usage outside the previously seen regimes of compute load is critical. Examples of such applications are: 1) risk-aware planning of large workload migration across various clusters, and 2) substantial, planned drops in compute load to respond to grid-level demand response events.

To evaluate the applicability of the proposed approaches to such scenarios, we discuss two conducted experiments (tests).

In **Test I**, power drop experiments were conducted in 8 clusters spanning all types of power architectures across the Google datacenter fleet. The planned CPU usage drop was performed between 3:50 pm and 5:25 pm on a given day by progressively terminating non-critical tasks based on their priority (starting with the lowest priority). Performance of the proposed models was evaluated within the testing interval. Figure 7 shows an example of the actual power and CPU usage for an experimental PDU, along with the corresponding predictions of the three models.

To evaluate performance of the power models, average MAPE is computed across all PDUs within a given cluster and using the data instances within the testing interval, i.e., $\frac{\sum_{PDU \in C} MAPE^{PDU}}{|C|}$. The experimental results in Figure 8 demonstrate that the Unified Machine Model has the best predictive

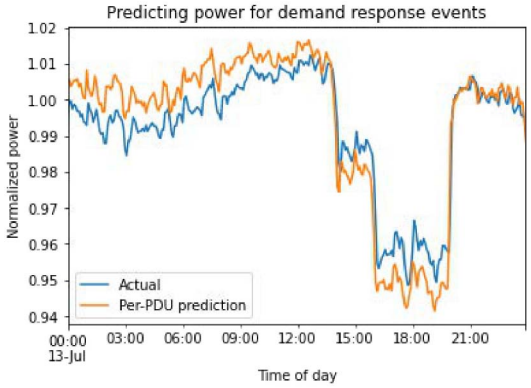

Fig. 9. Power usage profile on a test day and the corresponding Per-PDU predictions computed by applying the trained models to the actual PDU-level CPU usage profiles for all power domains within the selected datacenter campus.

performance during the power drop events, with an average MAPE of less than 6% across all the test clusters. However, there are cases where the Per-PDU and Unified PDU models perform comparably well. Clusters 3 and 6 in Figure 8 are two examples where both the Per-PDU and Unified PDU models generate accurate predictions. This is not surprising given that the dropped, experimental, CPU usage is captured in the training data. There are, however, scenarios, where the Per-PDU model's MAPE is less than 5% even when the low CPU usage regime is not present in its training data (e.g., see clusters 1, 4 and 5 in Figure 8), which implies that the extrapolated, low-usage-regime segment of the Per-PDU model is still an effective predictor of PDU power.

The reliability of the electricity grid can be jeopardized and the price for obtaining additional power capacity correspondingly rises in periods of high demand. Therefore, utilities offer incentives, through Demand Response (DR) programs, for their customers to drop loads when requested. As large power consumers on the grid, datacenters' ability to reduce demand at grid-critical hours could contribute to the electricity grid's stability and lower average cost.

During summer months of 2021, **Test II** was conducted to evaluate Google datacenters potential to drop power for 6 hours on selected days: 7/13, 7/29, 7/30, 7/31, 8/11. The power load was dropped by enforcing lower cluster-level CPU consumption using the capacity curve mechanism that leverages temporal flexibility of some of compute workloads, as recently proposed in [9]. The load shedding was accomplished in two stages lasting 2 and 4 hours to avoid negative performance implications on workloads due to the abrupt and significant drops in available CPU capacity. The outcome of the campus-level CPU usage drop on its power consumption on one of the test days is captured in Figure 9. The normalized power consumption is computed by dividing campus-level actual power usage by its average daily value over non-event hours.

The Per-PDU model was used to evaluate power usage predictability using CPU usage profiles on event days. To obtain the campus-level load shedding impact, the Per-PDU models, trained on days before the test dates were applied for every power domain within a campus, and then summed to compute

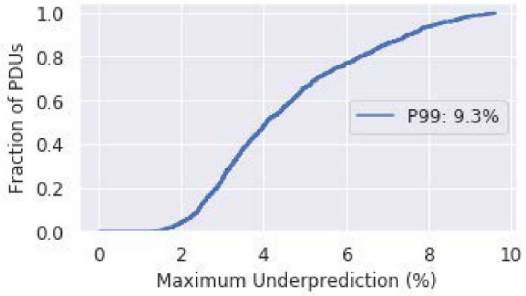

Fig. 10. CDF of $\max_{d\in 5 \text{ month horizon}} WUPE^{PDU}(d)$ across all PDUs in the datacenter fleet.

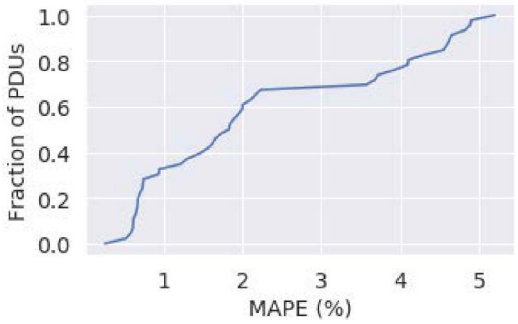

Fig. 11. Fraction of the hold-out PDUs with $MAPE^{PDU}$ smaller than a given percent error.

the campus-level power consumption. It was observed that the PDU-level predictive performance guarantees were not affected by the large perturbations in CPU usage. Moreover, as expected, the power prediction uncertainty at a campus level decreases and the evaluated, out-of-sample, MAPEs are: 0.6% (7/13), 0.5% (7/29), 0.5% (7/30), 0.3% (7/31), and 0.6% (8/11).

2) Power Capping: As discussed in Section I, another benefit of accurate power models is software-controlled power capping [10], used to limit maximum power usage of a PDU by controlling its CPU usage. To attain this goal in a risk-aware manner, characterizing model's underprediction error (i.e., when predictions are lower than the actual power measurements) is critical to ensure that the targeted PDU-level power usage limit is respected by the model-based control of its CPU usage limit. To that end, the worst (largest) 90-day underprediction (percent) error is computed daily using the data instances with high PDU-level CPU utilizations (and therefore, high power utilizations):

$$WUPE^{PDU}(d) = 100\% \max_{\substack{u^{PDU}_{CPU}(t)\in high \\ t\in[d-1,d-90]}} \frac{\left(u^{PDU}_{POW}(t) - \hat{u}^{PDU}_{POW}(t)\right)}{u^{PDU}_{POW}(t)}. \quad (6)$$

To avoid tripping PDUs' circuit breakers, the power capping system typically responds by throttling low priority computing tasks, which reduces their CPU (thereby, power) usage. To bound the frequency of the power capping events and define the related, user-perceived service level objectives (SLOs), statistical properties of a model's underprediction error is used.

Next, for each PDU, maximum underprediction error of the Per-PDU model is computed over a 5-month time horizon, (i.e., $\max_{d\in 5 \text{ month horizon}} WUPE^{PDU}(d)$), the CDF of which is shown in Figure 10. The maximum of the worst underprediction errors across all PDUs are then used to obtain the fraction of PDUs with the corresponding maxima less than a given value. The analysis shows that for 99% of PDUs, the worst underprediction error is less than 9.3%. It is observed that larger underprediction errors in some PDUs typically happen due to the unpredicted machine upgrades, to which the Per-PDU model typically adjusts within 2-3 days.

C. Use Case #3: Rightsizing Infrastructure Upgrades

The structure and accuracy of the Unified Machine Model make it suitable for cost- and performance-aware infrastructure planning, commonly referred to as rightsizing. Typical planning scenarios require: 1) provisioning power consumption after deployments/decommissioning of machines, 2) analysis of the long-term impact of platform mix in a power domain on its power consumption, and 3) studying the effect of large workload migrations across a datacenter fleet on its power consumption. Such scenarios require long-term forecasting (monthly, quarterly, etc.), where PDU power and resource usage data are either unavailable, or the predicted operating regime is different from the historically observed pattern. The Unified Machine Model predicts power consumption of a PDU by aggregating its machine-level predictions, where machine-level training instances capture the full dynamic range of CPU utilizations (see Section II-D2 for more details). This enables the model to predict power consumption at various operating regimes with higher accuracy.

To evaluate performance of the Unified Machine Model in predicting power consumption of new (future) PDUs, 47 PDUs across all power architectures were randomly selected and removed from the training dataset. The Unified Machine Model was then retrained on the reduced training data set, and used to predict PDU-level power for each of the hold-out PDUs at 5-minute time granularity. The model was tested on a week of data outside the time period used in the training data set. For each hold-out PDU, $MAPE^{PDU}$ was computed using all data instances within the test week/dataset. When averaged across all tested PDUs, the computed MAPE was 2.23%, while PDU-level MAPEs were below 5% for the majority of the tested domains (Figure 11). This was consistent with the prediction accuracy for PDUs included in the training data set, and indicates the ability of the Unified Machine Model to provision power of unseen PDU configurations.

IV. Conclusion

In this paper, design of two types of statistical power models (named the Per-PDU and Unified models) is discussed along with rigorous validations of their accuracy, simplicity, interpretability and applicability to all hardware configurations and workloads across the Google fleet of hyperscale datacenters. The two types of models are already deployed in production and trained using machine-level and PDU-level power and CPU usage measurements, and their basic hardware characteristics. The models are developed to accommodate several use cases of interest including: cost- and carbon-aware load

management, power and carbon footprint provisioning, peak power control (i.e., power capping) and infrastructure right-sizing. To the best of our knowledge, this is the largest scale datacenter power modeling effort in both the scope of diverse use cases and the variety of hardware configurations and workload types used for modeling and validation. The extensive analysis of the models' prediction accuracy across Google fleet demonstrates their state-of-the-art performance.

The models currently facilitate several decision-making scenarios that datacenter planners and compute providers face today. Our goal is to develop modeling extensions to enable Google's 24/7 carbon-free datacenters [32].

ACKNOWLEDGMENT

The authors are thankful to Vasileios Kontorinis for executing the power drop experiments and his review of their work. Also, they would like to thank Parthasarathy Ranganathan for his guidance, Carrie Grimes Bostock, Sundar Dev, Ian Schneider, Eric Mullen, Ross Koningstein and Thomas Olavson for their constructive comments. They would also like to thank the editors and reviewers for their helpful questions and comments.

REFERENCES

[1] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, "Recalibrating global data center energy-use estimates," *Science*, vol. 367, no. 6481, pp. 984–986, 2020.

[2] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 732–794, 1st Quart., 2016.

[3] J. Choi, S. Govindan, B. Urgaonkar, and A. Sivasubramaniam, "Profiling, prediction, and capping of power consumption in consolidated environments," in *Proc. IEEE Int. Symp. Model. Anal. Simulat. Comput. Telecommun. Syst.*, 2008, pp. 1–10.

[4] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam, "Managing server energy and operational costs in hosting centers," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, 2005, pp. 303–314.

[5] Y. Liu, G. Cox, Q. Deng, S. C. Draper, and R. Bianchini, "FastCap: An efficient and fair algorithm for power capping in many-core systems," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw. (ISPASS)*, 2016, pp. 57–68.

[6] S. Dev, D. Lo, L. Cheng, and P. Ranganathan, "Autonomous warehouse-scale computers," in *Proc. DAC*, 2020, pp. 1–6.

[7] L. Ismail and H. Materwala, "Computing server power modeling in a data center: Survey, taxonomy, and performance evaluation," *ACM Comput. Surveys*, vol. 53, no. 3, p. 58, 2020.

[8] A. Verma, L. Pedrosa, M. R. Korupolu, D. Oppenheimer, E. Tune, and J. Wilkes, "Large-scale cluster management at Google with Borg," in *Proc. Eur. Conf. Comput. Syst. (EuroSys)*, Bordeaux, France, 2015, p. 18.

[9] A. Radovanovic *et al.*, "Carbon-aware computing for datacenters," 2021, *arxiv:2106.11750*.

[10] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *ACM SIGARCH Comput. Archit. News*, vol. 35, no. 2, pp. 13–23, 2007.

[11] A. Gandhi, M. Harchol-Balter, and I. Adan, "Server farms with setup costs," *Perform. Eval.*, vol. 67, no. 11, pp. 1123–1138, 2010.

[12] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy, "Optimal power allocation in server farms," in *Proc. 11th Int. Joint Conf. Meas. Model. Comput. Syst.*, 2009, pp. 157–168. [Online]. Available: https://doi.org/10.1145/1555349.1555368

[13] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for Internet-scale systems," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 123–134, Aug. 2009. [Online]. Available: https://doi.org/10.1145/1594977.1592584

[14] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in *Proc. 6th Symp. Oper. Syst. Design Implement.*, 2004, pp. 137–150.

[15] "Data center demand response." Distributed Energy. 2019. [Online]. Available: https://www.distributedenergy.com/home/article/13036367/data-center-demand-response

[16] *IEEE Standard for a Method to Calculate Near Real-Time Emissions of Information and Communication Technology Infrastructure*, Green ICT Standard IEEE 1922.2-2019, 2019.

[17] Z. Liu *et al.*, "Renewable and cooling aware workload management for sustainable data centers," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 1, pp. 175–186, 2012.

[18] X. Liu and F. Kong, "Datacenter power management in smart grids," in *Foundations and Trends in Electronic Design Automation*, vol. 9. Hanover, MA, USA: Now Publ., 2015, pp. 1–98.

[19] A. Radovanovic. "Our data centers now work harder when the sun shines and wind blows." 2020. [Online]. Available: https://blog.google/inside-google/infrastructure/data-centers-work-harder-sun-shines-wind-blows

[20] S. Rivoire, P. Ranganathan, and C. Kozyrakis, "A comparison of high-level full-system power models," in *Proc. HotPower*, 2008, pp. 32–39.

[21] J. D. Davis, S. Rivoire, M. Goldszmidt, and E. K. Ardestani, "No hardware required: Building and validating composable highly accurate os-based power models," Microsoft, Albuquerque, NM, USA, Rep. MSR-TR-2011-89, 2011.

[22] L. A. Barroso, U. Hölzle, and P. Ranganathan, "The datacenter as a computer: Designing warehouse-scale machines," in *Synthesis Lectures on Computer Architecture*, vol. 13. San Rafael, CA, USA: Morgan Claypool Publ., 2018, pp. 1–189.

[23] M. Tirmazi *et al.*, "Borg: The next generation," in *Proc. EuroSys*, Heraklion, Greece, 2020, pp. 1–14.

[24] S. McVeety and R. Lippert. "Dataflow under the hood: The origin story." 2020. [Online]. Available: https://cloud.google.com/blog/products/data-analytics/how-cloud-batch-and-stream-data-processing-works

[25] J. H. Friedman, "Multivariate adaptive regression splines," *Ann. Stat.*, vol. 19, no. 1, pp. 1–67, 1991.

[26] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vols. 2–3, pp. 18–22, Dec. 2002.

[27] O. Mutanga, E. Adam, and M. A. Cho, "High density biomass estimation for wetland vegetation using WorldView-2 imagery and random forest regression algorithm," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 18, pp. 399–406, Aug. 2012.

[28] "Scikit-Learn: Machine learning in Python." [Online]. Available: https://scikit-learn.org/stable/ (Accessed: Nov. 2019).

[29] K. Potdar, T. S. Pardawala, and C. D. Pai, "A comparative study of categorical variable encoding techniques for neural network classifiers," *Int. J. Comput. Appl.*, vol. 175, no. 4, pp. 7–9, 2017.

[30] J. Neyman, "On the two different aspects of the representative method: The method of stratified sampling and the method of purposive selection," in *Breakthroughs in Statistics*. New York, NY, USA: Springer, 1992, pp. 123–150.

[31] X. Zheng *et al.*, "Full parameter time complexity (FPTC): A method to evaluate the running time of machine learning classifiers for land use/land cover classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 2222–2235, Jan. 2021, doi: 10.1109/JSTARS.2021.3050166.

[32] "24/7 by 2030: Realizing a carbon-free future." Google. 2020. [Online]. Available: https://www.gstatic.com/gumdrop/sustainability/247-carbon-free-energy.pdf

Ana Radovanovic received the Ph.D. degree in electrical engineering from Columbia University in 2005. She has been a Research Scientist with Google since early 2008. She worked for three years as a Research Staff Member with the Mathematical Sciences Department, IBM TJ Watson Research Center. Since 2013, she has been focused all her research efforts with Google on building innovative technologies and business models with two goals in mind: 1) to deliver more reliable, affordable, and clean electricity to everyone in the world and 2) to help Google become a thought leader in decarbonizing the electricity grid. Nowadays, she is widely recognized as a Technical Lead and Research Entrepreneur. She is a Senior Staff Research Scientist, serving as a Technical Lead for Energy Analytics and Carbon Aware Computing with Google.

Bokan Chen received the B.S. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2011, and the Ph.D. degree from the Department of Industrial and Manufacturing System Engineering, Iowa State University, Ames, IA, USA, in 2016. He is currently a Senior Data Scientist with Google, Sunnyvale, CA, USA. His research interests include optimization under uncertainty, machine learning, and statistical modeling.

Saurav Talukdar received the B.Tech. and M.Tech. degrees in mechanical engineering from the Indian Institute of Technology Bombay, Mumbai, India, in 2013, and the Ph.D. degree in mechanical engineering from the University of Minnesota, Minneapolis, MN, USA, in 2018. He is a Control Systems and Machine Learning Engineer with Google and focuses on energy optimization for Google Data Centers. Prior to joining Google in 2019, he was a Battery Algorithm Engineer with Apple working on system identification and thermal management of Lithium ion batteries in iPhones.

Binz Roy received the S.M. and Ph.D. degrees in mechanical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA. He has spent more than a decade in industrial Research and Development with Google, Bosch, and General Electric. In his most recent role, he's responsible for optimizing technical infrastructure to support Google's machine learning fleet. His broader areas of interest include dynamical systems and control, nonlinear programming, and statistical risk modeling.

Alexandre Duarte received the B.Sc. and M.Sc. degrees in computer science from the Federal University of Paraiba, Brazil, in 2001 and 2003, respectively, and the Ph.D. degree in computer science from the Federal University of Campina Grande, Brazil, in 2010. He is currently a Software Engineer with Google, Austin, TX, USA. His interests include distributed computing, high performance, software development, and programming languages.

Mahya Shahbazi (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Western Ontario, London, ON, Canada, in 2017. She is currently a Senior Research and a Development Engineer with Google, Sunnyvale, CA, USA. Before joining Google, she was a Postdoctoral Research Associate with Johns Hopkins University, Baltimore, MD, USA, from 2018 to 2019; and the Center for Canadian Surgical Technologies and Advanced Robotics, London, from 2017 to 2018. Her research interests include datacenter efficiency, system optimization, medical robotics, telerobotics and human–robot interaction, machine learning, and advanced control systems. She was a recipient of several awards, including the 2018 Natural Sciences and Engineering Research Council of Canada Postdoctoral Fellowship.