

# Computing Server Power Modeling in a Data Center: Survey, Taxonomy and Performance Evaluation

LEILA ISMAIL\* and HUNED MATERWALA, Department of Computer Science and Software Engineering, College of Information Technology, United Arab Emirates University, United Arab Emirates

Data centers are large scale, energy-hungry infrastructure serving the increasing computational demands as the world is becoming more connected in smart cities. The emergence of advanced technologies such as cloud-based services, internet of things (IoT) and big data analytics has augmented the growth of global data centers, leading to high energy consumption. This upsurge in energy consumption of the data centers not only incurs the issue of surging high cost (operational and maintenance) but also has an adverse effect on the environment. Dynamic power management in a data center environment requires the cognizance of the correlation between the system and hardware level performance counters and the power consumption. Power consumption modeling exhibits this correlation and is crucial in designing energy-efficient optimization strategies based on resource utilization. Several works in power modeling are proposed and used in the literature. However, these power models have been evaluated using different benchmarking applications, power measurement techniques and error calculation formula on different machines. In this work, we present a taxonomy and evaluation of 24 software-based power models using a unified environment, benchmarking applications, power measurement technique and error formula, with the aim of achieving an objective comparison. We use different servers architectures to assess the impact of heterogeneity on the models' comparison. The performance analysis of these models is elaborated in the paper.

**Additional Key Words and Phrases:** Data center, server power consumption modeling, machine learning, energy-efficiency, resource utilization, green computing

## ACM Reference Format:

Leila Ismail and Huned Materwala. 2020. Computing Server Power Modeling in a Data Center: Survey, Taxonomy and Performance Evaluation. *ACM Comput. Surv.*, 41 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 Introduction

Data centers are substantial computing facilities serving as a back-end infrastructure for enabling globally competitive innovations and contributing to the socio-economic development [22, 37]. There is rapid growth to data centers comprising of thousands of computing nodes due to the emergence of smart cities and consequently the need of paradigms such as Cloud Computing [84], IoT [23] and Big Data Analytics [60]. This continuous storage and computing needs lead technical firms like Microsoft and Google to expand their data center infrastructures as large as a football field able to host thousands of nodes[24]. The data center services market is projected to grow at a

\*Corresponding Author

Authors' address: Leila Ismail; Huned Materwala, emails:{leila,huned.m}@uaeu.ac.ae, Department of Computer Science and Software Engineering, College of Information Technology, United Arab Emirates University, P.O. Box 15551, Al Ain, United Arab Emirates.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2020 Association for Computing Machinery.

0360-0300/2020/5-ART \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

compound annual growth rate (CAGR) of 13.69% over the forecast period of 2018-2023 [118]. A data center, however, has a massive energy consumption engendering various economic problems and environmental hazards.

Energy consumption of data centers is becoming an important issue in an enterprise environment and has gained significant importance in recent years. A typical data center may consume energy equivalent to that of 25,000 households [24]. According to the report by National Resources Defense Council (NRDC) in the USA, the data centers in 2013 consumed 91 billion kWh of energy, comparable to 34 large power plants (coal fired) [32]. This energy consumption of the data center is anticipated to reach around 140 billion kWh by 2020, equivalent to the annual output of 50 power plants, incurring the cost of \$13 billion in electricity to the American business. Furthermore, a typical data center's energy cost increases by 100% every five years [24]. The carbon emissions caused by data centers in 2005 in the USA was as much as that of a mid-sized nation like Argentina [81]. It is expected that by 2020 the annual carbon emission of the data centers will reach 100 million metric tons [32].

The data center's power consumption comprises of [49, 91, 124]: 1) the power consumed by the data center's IT equipment such as computing servers and storage (56%), 2) the power consumed by the infrastructure facilities such as the cooling systems (30%), the power distribution/conditioning systems (8%), and the lighting (1%), and 3) the power consumed by the network (5%) [91, 124]. To reduce the data center energy consumption, different methods have been introduced in the literature, such as deploying energy-efficient algorithms, modifying the hardware components architecture [83], designing measures for efficient air handling [50] and cooling [52], and using efficient options for the power supply. These methods require modeling of the relationship between systems' power consumption (considered as dependent variable) and performance counters (considered as independent variable) [114]. Consequently, data center operators need an accurate power model for designing an energy efficient system [98, 113], managing a center power consumption [66] and using energy-aware scheduling for optimization [121]. These power models proposed in the literature are classified as 1) hardware-based models that use fan speed, voltage, current, capacitance, resistance, and motherboard components as the independent variables, and 2) software-based models that target either individual subsystems of a server, such as CPU, memory, disk and network, or a virtual machine, or a full-system (a computing server) [86]. We use the terminology computing server for a full-system in the remainder of the paper. The hardware-based power models require sensors to measure different variables on a server's hardware. This adds in extra hardware and energy consumption costs incurred by these sensors attached to thousands of servers in a data center. However, the software-based models do not require external sensors to get the values of the model variables adding no extra cost. Therefore, in this paper, we focus on the software-based power models. The software models use performance metrics provided by the operating system that we call *System\_Performance\_Metrics* (S\_PM) based models, or performance monitoring counters provided by the hardware subsystems of a server that we call *System\_Performance\_Counters* (S\_PC) based, or a combination of system performance metrics and counters that we call *System\_Performance\_Metrics\_Counters* (S\_PMC) based. The metrics provided by the operating system indicate the utilization level of a system (CPU utilization, memory utilization, disk I/O rate and network I/O rate), whereas the counters provided by the hardware indicate the performance of the different server's subsystems, such as number of cache misses [2], number of branch instructions [1], and number of interrupts [3]. In this work, we evaluate the performance of software-based computing server's power consumption models that have been proposed in the literature. However, we are unaware of any objective comparison of these models using a unified experimental setup for a diverse set of applications. This paper focuses to address this void.

In this study, we present a taxonomy and comparative evaluation of software-based power models. We evaluate their performance in terms of standard error of estimation. This is in a unified environment and experimental setup. In this evaluation, we make use of four different tools for models formulation and validation and five different applications for models testing. The key research contributions of this work are as follows.

- We classify the work on power modeling into software and hardware-based models, and present a taxonomy of different software-based power models in the data center's power consumption modeling literature. We discuss the temporal evolution of the various models and capture the assumptions conducting to the development of a given model in a certain period of time.
- We evaluate the performance of 24 different software-based power models in terms of standard error of estimation using a diverse set of benchmarking applications in a unified experimental setup. The experiments show that the support vector machine (SVM) power model has the least standard error of estimation.
- The portability of the relationship between a server's power consumption and the user and system performance counters is also verified on different server architectures in our experimental testbed.

To our knowledge, this is the first work to classify software-based power models in the literature and evaluate their performance in a unified environment and setup.

The rest of the paper is organized as follows. Section II synthesizes a taxonomy of the works on software-based power models. The experimental setup, experiments and the performance evaluation in terms of standard error of estimation for the studied power models are described in Section III. Section IV overviews the related works. The paper is concluded with the lessons learned and possible future research directions in Section V.

## 2 Software-based Power Models

We first present the taxonomy (Figure 1) and limitations (Table 1) of state-of-the-art software-based power models (Sub-section 2.1). This allows to capture which were the assumptions conducting to the development of a given model in a certain period of time and unravel on one hand the technological development of servers and, on the other hand, the corresponding improvements in the precision of the models. In sub-section 2.2, we recall the experimental setups used for the evaluation of these models in the literature along with their precision (Table 2) and describe the workflow of power model development system that is used to build the studied power models.

### 2.1 Taxonomy of Software-based Power Models

We present a classification and temporal evolution of the software-based power models in the literature. We classify these models into two primary categories: 1) linear and 2) non-linear. We further classify the models in these categories into 1) mathematical formula based on fixed slope and intercept and 2) machine learning based on variable slope and intercept. The former is based on a server's idle and full load power consumption values. It does not consider the implication of the spatial distribution of power consumption for the  $S_{PM}$  and  $S_{PC}$  values which lie between the idle and the full load states of the server. Whereas, the machine learning models are developed based on the distribution of the power consumption of the  $S_{PM}$  and  $S_{PC}$  utilization values. Figure 1 shows our taxonomy of power models in the literature.

#### 2.1.1 Linear Power Models

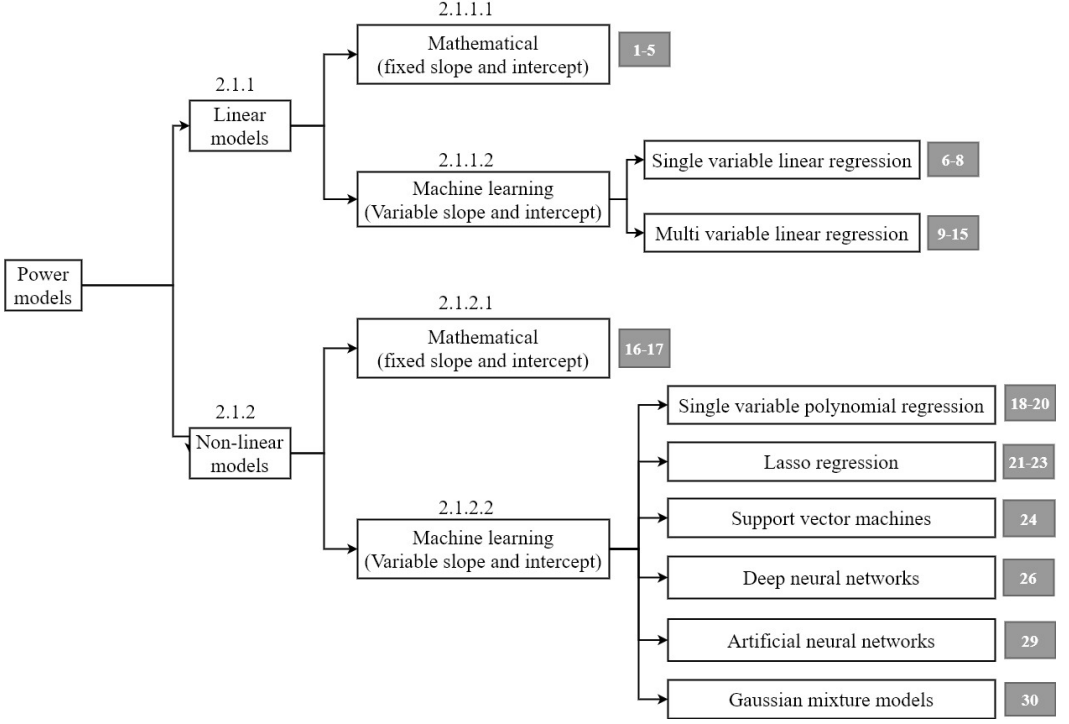


Fig. 1. Taxonomy of the power models. The numbers above the classified categories denote the sub-subsection and sub-sub-subsection, and the gray box besides each category represents the corresponding equation numbers of the power models.

### 2.1.1.1 Mathematical Formula: Single Variable Linear With Fixed Slope and Intercept (SVLF)

In the 2000s, with the research attention drawn towards the energy efficiency of the data centers due to the onset of large-scale web services and the underlying massive parallel computing infrastructure, studies were conducted to analyze the computing servers' hardware power consumption breakdown [21, 41]. [21] studied the power consumption of web servers using real workloads derived from the logs of three production websites (1998 Winter Olympics, a financial services company, and Information Resource Caching project affiliated with the National Laboratory for Applied Network Research). Results from this study stated that the server's power consumption is highly dominated by its CPU utilization in a linear manner, influencing many works, that come after, on server power consumption modeling. In 2006, [57] formulated a linear model as stated in Equation 1 to calculate power consumption ( $P$ ) of a server with CPU utilization ( $u_{cpu}$ ) as the independent variable (S<sub>PM</sub>-based). The slope of the linear model is the difference between the power consumption when the server is at full load ( $P_{MAX}$ ) and the power consumption when the server is idle ( $P_{MIN}$ ), and the intercept of the model being  $P_{MIN}$ . However, the model was not evaluated for its accuracy by the authors. Later in 2007, [41] studied the power usage of thousands of servers for workloads taken from different classes of web services such as Websearch, Webmail, and Mapreduce for over a period of approximately six months and confirmed the results obtained by [21].

$$P = (P_{MAX} - P_{MIN}) \times u_{cpu} + P_{MIN} \quad (1)$$

The model in Equation 1 was later evaluated by [41] and [27] for its accuracy. The model has been used by various works in the literature for electricity consumption cost prediction in a heterogeneous server environment [92], for energy-aware resource management technique [47], and for energy-efficient cloud computing [35, 72, 95, 101, 122]. The model in Equation 1 was then presented by [22] using the ratio  $\frac{P_{MIN}}{P_{MAX}}$  denoted by  $k$  as stated in Equation 2. The slope  $P_{MAX} - P_{MIN}$  and the intercept  $P_{MIN}$  of Equation 1 are then represented as  $[1 - k] \times P_{MAX}$  and  $k \times P_{MAX}$  respectively in Equation 2. Substituting the value of  $k$  in Equation 2 yields to Equation 1, making them identical. The model has been used by various works on energy efficient cloud resource management in the literature [12, 58, 89, 109].

$$P = ([1 - k] \times P_{MAX} \times u_{cpu}) + (k \times P_{MAX}) \quad (2)$$

Later in the year 2007, [41] showed that an idle server almost consumes 70% of its peak power consumption (i.e.  $k = 0.7$ ). Based on this study, [14] stated a S\_PM-based power model as in Equation 3 derived from Equation 2 by substituting the value of  $k$  as 0.7. The model has been used in the literature for data center energy efficiency [56, 103].

$$P = P_{MAX} \times (0.7 + 0.3 \times u_{cpu}) \quad (3)$$

In 2011, with the advancement of cloud computing technology wherein computing infrastructure and applications are provided as services to end users under pay-per-use model, [25] developed CloudSim software for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. The power model used in the simulation software to predict the power consumption for executing the cloud applications is based on the method of linear interpolation [70] using CPU utilization as the independent variable (S\_PM-based) as stated in Equation 4. Linear interpolation assumes a piece-wise linear relation between each interval of CPU utilization values and corresponding power consumption values instead of assuming a single linear function between idle and full load utilization. The model was later adopted by various works using CloudSim to evaluate the energy efficiency of the cloud resource allocation algorithms [11, 15, 28, 42, 65, 69, 94, 117].

$$P = P_1 + (\Delta \times (\frac{u_{cpu} - u_{1cpu}}{10}) \times 100) \quad (4)$$

where  $P_1$  and  $P_2$  are the power values corresponding to the CPU utilization  $u_1$  and  $u_2$  respectively ( $u_1 \leq u \leq u_2$ ) and  $\Delta$  is the slope of the line between points  $(u_1, P_1)$  and  $(u_2, P_2)$ .

In 2013, with the accelerating adoption of cloud computing serving applications of type I/O and web-based interaction, a significant amount of network is then used due to the virtualization of computing servers. Therefore, [63] studied the energy consumption of servers in a cloud computing environment for energy efficiency for network-intensive benchmarks and developed the relationship between the power consumed by an application running on a server and the application's throughput as stated in Equation 5. The power model is S\_PM-based, using the application's throughput as the independent variable.

$$P = ([P_{MAX} - P_{MIN}] \times \frac{Throughput}{Throughput_{Max}}) + P_{MIN} \quad (5)$$

where the throughput is application specific and is defined as the amount of load executed per unit of time. For instance, the throughput of a network-intensive HTTP application is known as the request rate defined by the number of requests processed per second.

### 2.1.1.2 Machine Learning Linear With Variable Slope and Intercept (MLLV)

- *Single Variable Linear Regression (SVLR)*: In 2008, [93] proposed a power management solution for data center based on a predicted power consumption value using a fitted-line regression model as stated in Equation 6. The authors experimentally collected the power consumption values of a server while running workloads at different CPU utilization values to develop this S\_PM-based model. The linear regression [13] model develops a linear relationship between the power consumption and the CPU utilization by calculating an intercept and a slope for the linear line. The model was later used by the works on energy-aware scheduling in a data center [16, 17, 48] and the works on power consumption modeling [90, 123].

$$P = \alpha + \beta u_{cpu} \quad (6)$$

where  $\alpha$  and  $\beta$  are the intercept and slope of the regression line whose values are calibrated for each server type experimentally such that the squared error is the minimum.

In 2010, [115] used the same linear regression model stated in Equation 6 for power and cooling management in the data centers. The authors fixed the value of intercept at server's idle power consumption as stated in Equation 7, instead of calibrating the value of intercept experimentally (Equation 6).

$$P = P_{MIN} + \beta u_{cpu} \quad (7)$$

In 2010, [67] conducted experiments on servers using web-transactions, HPC, and I/O-intensive workloads and found that the linear regression power model based on the CPU utilization only (Equation 6) predicts the power consumption with a large error. Consequently, the authors proposed a S\_PM-based power model to establish a linear relationship between the power consumption of an application and the application's throughput to predict the power of heterogeneous applications as stated in Equation 8.

$$P = \alpha + \beta(Throughput) \quad (8)$$

where, throughput is application specific and defined as the number of requests executed per second.

- *Multi Variable Linear Regression (MVLR)*: In 2006, [39] conducted experiments on a blade server to study its metric-level power consumption (CPU, memory, disk, and network) using different benchmarks such as SPECcpu2000 integer, SPECcpu2000 floating point, SPECjbb2000, SPECweb2005, the streams, and the matrix multiplication. Based on these results, the authors stated that the memory power consumption is likely to be equally important, if not more, as that of the CPU. Moreover, the power consumed by the disk and the network I/O peripherals can not be neglected. Consequently, the authors proposed a power consumption model based on multi linear regression as stated in Equation 9, with CPU utilization ( $u_{cpu}$ ), memory utilization ( $u_{mem}$ ), disk I/O rate ( $u_{disk}$ ) and network I/O rate ( $u_{net}$ ) as the independent variables (S\_PM-based). The model was later used by [104] while profiling power usage in cloud computing environment. In 2014, [9] proposed multi regression model as stated in Equation 10 using the performance metrics similar to that in Equation 9. However, the model used the server's idle power consumption as the intercept of the model instead of calculating it based on the regression fitted-hyperplane.

$$P = \alpha + \beta_1 u_{cpu} + \beta_2 u_{mem} + \beta_3 u_{disk} + \beta_4 u_{net} \quad (9)$$

$$P = P_{MIN} + \beta_1 u_{cpu} + \beta_2 u_{mem} + \beta_3 u_{disk} + \beta_4 u_{net} \quad (10)$$

where  $\alpha$ ,  $\beta_1$ ,  $\beta_2$ ,  $\beta_3$ , and  $\beta_4$  are the model parameters calibrated for each server type experimentally such that the squared error of estimation is the minimum.

In 2010, [64] examined the power consumption of a server and found that the most power consuming components of a server are the CPU, the memory and the disk. Based on these results, the authors proposed a S\_PM-based multi regression power model as stated in Equation 11. The model was later used by [75].

$$P = \alpha + \beta_1 u_{cpu} + \beta_2 u_{mem} + \beta_3 u_{disk} \quad (11)$$

Later in 2010, [20] conducted experiments and found that the power consumption of a server is a linear function of the CPU load, memory utilization, disk operations per second and instruction cache. Based on these results, the authors proposed a model based on S\_PMC as in Equation 12. In the same year, [34] conducted experiments to study the energy consumed by different applications. The authors used synthetic workload to stress the server and simultaneously measured the power consumption and 165 server performance indexes. Nine significant S\_PMC, such as the square of CPU utilization, context switches, cache references, square of cache misses, disk read/write per second, number of TLB interrupts, RES interrupts, NMI interrupts and LOC interrupts corresponding to power consumption are then used to develop a fine-grained power model as stated in Equation 13.

$$P = \alpha + \beta_1 u_{cpu} + \beta_2 u_{mem} + \beta_3 u_{disk} + \beta_4 cache \quad (12)$$

$$P = P_{MIN} + \sum_{n=1}^9 \beta_n C_n \quad (13)$$

where  $C_n$  are the performance counters.

In 2012, [119] studied different hardware and software power measurement solutions for HPC applications. Based on the analysis presented, the hardware solutions are expensive compared to the software ones. The authors proposed S\_PMC based model for power prediction as stated in Equation 14 using LLC load misses, LLC loads, LLC store misses, LLC stores, branch misses, branches, cache misses, cache references, context switches, cycle, dTLB load misses, dTLB loads, dTLB store misses, dTLB stores, iTLB load misses, iTLB loads, instructions, major and minor faults, page faults, CPU utilization, number of bus transactions, and DRAM access as the independent variables. In addition to these variables, the model also used CPU temperature and frequency. The authors developed different models using only the highest CPU frequency and using all the CPU frequencies. The authors studied the correlation of S\_PMC with the power consumption and used all the variables that increased the prediction accuracy, starting from the one which has the highest correlation.

$$P = \beta_0 + \sum_{n=1}^{33} \beta_n C_n \quad (14)$$

Later in 2013, [62] studied the power consumption of HPC servers using a set of real-life applications. The authors proposed the use of clustering approach to group the applications having similar power characteristics. Each group of applications have a different power model. The selection of power model is done automatically using decision tree. The authors proposed S\_PC-based power model as stated in Equation 15 using 24 performance counters namely: LLC load misses, LLC loads, LLC stores, LLC store misses, branches, cache misses, cache references, context switches, cycles, dTLB load misses, dTLB loads, dTLB stores, dTLB store misses, iTLB loads, iTLB load misses, instructions, major faults, minor faults, page faults, DRAM access 1 and 2, and number of bus transactions. In addition to these counters, the model also uses CPU temperature.

$$P = \alpha + \sum_{n=1}^{24} \beta_n C_n \quad (15)$$

### 2.1.2 Non-Linear Models

#### 2.1.2.1 Mathematical Formula: Single Variable Non-Linear With Fixed Slope and Intercept (SVNLF)

In 2007, while studying the power usage characteristics of servers, [41] confirming that the power consumption of a server is highly dominated by the server's CPU utilization, the authors presented, in addition to the linear power model stated previously in Equation 1, an empirical non-linear power model. The authors performed experiments on thousands of heterogeneous servers and found that S\_PM-based non-linear power model, stated in Equation 16, fits the power consumption curve of the server better than the linear model (Equation 1). This non-linear model was later used by various works on power consumption modeling [92, 107].

$$P = ([P_{MAX} - P_{MIN}] \times [2u_{cpu} - u_{cpu}^r]) + P_{MIN} \quad (16)$$

where  $r$  is the calibration parameter whose value is obtained experimentally for each server type such that it minimizes the squared error of estimation.

Later in 2007, considering the increasing concern of power consumption in streaming-media servers, [76] studied the power behavior of the media servers. The authors performed experiments and observed the power consumed by different media servers using streaming media workloads and found that the power consumption of the servers is based on the idle power and the full load power, and the power consumption is related to the CPU utilization of the server. The results of the experiments showed that the power consumption of the media servers increases non-linearly when the CPU utilization of the server increases from 0% to 100%. Based on these observations, [76] considered that the power consumption of a streaming-media server as an exponential function of the server's CPU utilization and proposed a S\_PM-based power model as stated in Equation 17. The model was later used by [108] for energy efficient resource management in cloud service data centers.

$$P = ([P_{MAX} - P_{MIN}] \times [\alpha u_{cpu}^\beta]) + P_{MIN} \quad (17)$$

where  $\alpha$  and  $\beta$  are the model parameters calibrated for each server type experimentally such that the squared error of estimation is the minimum.

#### 2.1.2.2 Machine Learning Non-Linear With Variable Slope and Intercept (MLNLV)

- *Single Variable Polynomial Regression (SVPR)*: In 2012, [61] extended the linear regression power model based on CPU utilization only (Equation 6) to quadratic model as stated in Equation 18 based on the fact that the power consumption of a server is proportional to the square of the CPU frequency. This relationship between a server's power consumption and the CPU frequency is due to Dynamic Voltage and Frequency Scaling (DVFS) capability exhibited by a server with its advancement. DVFS is a technique that dynamically adjusts the voltage and frequency of a server's CPU to optimize resource allocation and maximize power savings when the resources are not required [71]. In the same year, [53] presented a power model having a polynomial of degree 'r' as stated in Equation 19 instead of using the quadratic polynomial as in Equation 18 to avoid over-fitting of the model. In 2013, [123] performed experiments on seven heterogeneous servers using the SPECpower benchmark



[31] to analyze the accuracy of the linear regression model based on the CPU utilization (Equation 6). The results showed that not all the servers hold the linear relationship between power consumption and CPU utilization. Based on this finding, the authors extended the linear model to higher degree polynomial models of degree two and three as stated in Equations 18 and 20 respectively and found that the polynomial models are more accurate than the linear model. The models in Equations 18-20 are S\_PM-based using CPU utilization as the independent variable. In 2016, [26] used polynomial regression to capture the non-linear behavior between S\_PM independent variables and server power consumption.

$$P = \alpha + \beta_1 u_{cpu} + \beta_2 u_{cpu}^2 \quad (18)$$

$$P = \alpha + \beta_1 u_{cpu} + \beta_2 u_{cpu}^r \quad (19)$$

$$P = \alpha + \beta_1 u_{cpu} + \beta_2 u_{cpu}^2 + \beta_3 u_{cpu}^3 \quad (20)$$

where  $\alpha$ ,  $\beta_1$ ,  $\beta_2$ ,  $\beta_3$ , and  $r$  are the model parameters calibrated for each server type experimentally such that the squared error of estimation is the minimum.

- *Lasso Regression (LR)*: In 2011, [82] studied the power consumption of servers corresponding to the CPU and memory utilization values. The authors found that the CPU and memory variables are independent of each other and consequently using a linear regression model based on these variables might not produce accurate results. The authors proposed the use of polynomial regression of order 3 with CPU and memory utilization values as the independent variables. Consequently, the S\_PM-based model has linear, quadratic, and cubic functions of the CPU and memory utilization values (CPU,  $CPU^2$ ,  $CPU^3$ , mem,  $mem^2$ ,  $mem^3$ ). To reduce the number of estimators, the authors proposed the use of lasso regression along with the polynomial, which they call as polynomial with lasso. Lasso regression is a linear model which performs L1 regularization that reduces the number of regression variables and obtains a subset of variables that minimizes the prediction error [106]. Equation 21 shows the basic function of the polynomial with lasso power model. In addition, the authors also proposed the use of exponential function along with the polynomial leading to the exponents of linear, quadratic, and cubic functions of the CPU and memory utilization values ( $e^{CPU}$ ,  $e^{CPU^2}$ ,  $e^{CPU^3}$ ,  $e^{mem}$ ,  $e^{mem^2}$ ,  $e^{mem^3}$ ). Lasso regression is performed in a similar way to reduce the number of estimators. The basic function of the polynomial + exponential with lasso is stated in Equation 22. The polynomial with lasso and polynomial + exponential with lasso models were later compared and used by [79] for energy-efficient scheduling in cloud computing.

$$\phi(.) = \{x_i^a; 1 \leq a \leq 3\} \quad (21)$$

$$\phi(.) = \{e^{x_i^a}; 1 \leq a \leq 3\} \quad (22)$$

where  $x_i$  is the CPU and the memory utilization.

In 2017, [80] experimentally found that the power consumption of a server is a function of various S\_PC that corresponds to a server's hardware resources such as the processor, the random access memory, the network interface controller. The authors used 30 different S\_PC representing the hardware resources to develop the relationship between the server's power consumption and resource utilization. The exposed low-level system performance counters (S\_PC) are branch-instructions, instructions, cache-misses, L1-icache-load-misses, branch-loads, branch-load-misses, LLC-loads, LLC-store-misses, LLC-load-misses, LLC-stores, dTLB-store-misses, dTLB-load-misses, dTLB-loads, dTLB-stores, bus-cycles, L1-dcache-stores,

L1-dcache-load-misses, L1-dcache-loads, CPU cycles, branch-misses, cache-references, iTLB-loads, iTLB-load-misses, node-load, node-stores, node-load-misses, node-stores-misses, ref-cycles, number of if octets out, and number of if octets in. In order to avoid over-fitting by only selecting the significant counters for a server, the author proposes the use of Lasso regression. Equation 23 shows the minimization function for the lasso regression model.

$$\underset{w}{\text{minimize}} \frac{1}{2n_{\text{samples}}} \|X_w - y\|_2^2 + \alpha \|w\|_1 \quad (23)$$

- *Support Vector Machines (SVM)*: In 2011, [82] found that the CPU and memory utilization values of a server are interdependent and cannot be used in a linear model to accurately predict the power consumption of the server. The authors proposed the use of SVM-based regression model to predict the power consumption using the function stated in Equation 24. SVM regression aims at finding a linear hyperplane, that fits the non-linearly correlated multidimensional regression parameters to the output variable [100]. The model is S\_PM-based using CPU and memory utilization as the independent variables. The model was later evaluated and used by [79] for energy-efficient scheduling in cloud computing.

$$f(x) = W\phi(x) + b \quad (24)$$

where  $W$  and  $b$  are the regression parameters calculated using the optimization problem to minimize the function stated in Equation 25 and  $\phi(x)$  is a function of CPU and memory utilization.

$$\underset{w, b, \zeta}{\text{minimize}} \frac{1}{2} \|W\|^2 + C \sum_{i=1}^N (\zeta_i + \zeta_i^*) \quad (25)$$

where  $C$  is the error penalty constraint, and  $\zeta_i$  and  $\zeta_i^*$  are the slack variables bounding the allowable regression errors.

- *Deep Neural Networks (DNN)*: In 2016, [74] stated that the static power models such as SVLF and SVNLF can not predict the power consumption accurately due to the heterogeneous and dynamic nature of workloads in a data center. The authors proposed the use of deep neural networks to analyze the trend in the past data center power consumption for prediction. The proposed deep learning prediction model is based on recursive auto-encoder and uses the power consumption data of a server corresponding to its CPU utilization, CPU load averaged over 5, 10 and 15 minutes, memory utilization, number of disk read/write, packets/s in and out, and the file system used/available for training the model (S\_PM-based). The recursive auto-encoder are neural networks that encodes the input into a latent space and tries to reconstruct the input as the output [105]. The auto-encoder output is then used to predict the value of power consumption such that it minimizes the objective function stated in Equation 26.

$$\epsilon_{RAE} = \epsilon_{PRD} \times 0.95 + \epsilon_{AE} \times 0.05 \quad (26)$$

where  $\epsilon_{PRD}$  and  $\epsilon_{AE}$  are the prediction error and the reconstruction error respectively. The value of  $\epsilon_{PRD}$  and  $\epsilon_{AE}$  are calculated using Equation 27 and 28.

$$\epsilon_{PRD} = \frac{\sum_{i=1}^{N_{train}} (||y(t) - y'(t)||)^2}{N_{train}} + 0.0001 \times (||W||)^2 \quad (27)$$

$$\epsilon_{AE} = \frac{\sum_{i=1}^{N_{train}} Err_{REC}(t)}{N_{train}} \quad (28)$$

where  $N_{train}$  is the size of training data set,  $\epsilon_{PRD}$  is the mean square error of the predicted values using the  $L_2$ -norm regularization parameter, and  $Err_{rec}$  is the reconstruction error.

- *Artificial Neural Networks (ANN)*: In 2015, [33] conducted an empirical study showing that the CPU-based linear power models do not provide accurate power prediction, especially for servers having multicore processor. This is mainly due to two reasons: 1) the power consumption has a non-linear relationship with the number of cores utilized, and 2) the power consumption of a server is application dependent for a given CPU utilization. The authors proposed the use of artificial neural networks for the prediction of power consumption. They used multilayer perceptron (MLP), which is a feedforward ANN composed of one or more hidden layers. The output of each hidden layer is computed using Equation 29.

$$a = \phi(Wi + b) \quad (29)$$

where  $W$  is the weight matrix,  $i$  is in input vector that consists of the independent variables,  $b$  is the bias vector,  $\phi(\cdot)$  is the activation function and  $a$  is the output vector, i.e., the predicted power consumption.

The authors used a MLP model with two hidden layers and a sigmoid activation function. The power model is based on different  $S\_PMC$  variables such as number of instructions, cycles, cache references, cache misses, branch instructions, branch misses, bus cycles, idle cycles frontend, task clock, page faults, context switches, CPU migrations, major and minor faults, L1d loads, L1d load misses, L1d stores, L1d store misses, L1d prefetch misses, L1i load misses, LLC loads, LLC load misses, LLC stores, LLC store misses, L1d prefetches, LLC prefetch misses, dTLB loads, dTLB load misses, dTLB stores, dTLB store misses, iTLB loads, iTLB load misses, branch loads, branch load misses, node loads, node load misses, node stores, node store misses, node prefetches, node prefetch misses, CPU usage, received and sent bytes, and CPU time. In addition to these variables the model also uses CPU temperature and frequency.

- *Gaussian Mixture Models (GMM)*: In 2010, [38] performed experiments to study the power consumption of heterogeneous applications at different CPU utilization levels. The results show that the relationship between the power consumption and the CPU utilization is not always linear but it is application specific. The authors found that the power consumption increases linearly with the CPU utilization for an application having high instructions per cycle (IPC), while for an application having high memory access with an increase in the CPU utilization after a certain value, there is no further increase in power consumption. Moreover, for an application having high cache conflicts, the power consumption decreases after certain CPU utilization value. Consequently, the authors proposed a power model based on different  $S\_PMC$  variables such as CPU utilization, instructions per cycles (IPC), memory access and cache transactions as the independent variables as stated in Equation 30. The prediction is done using Gaussian mixture model (GMM) to dynamically map different clusters of power consumption values with the corresponding clusters of performance metrics. GMM is a probabilistic model that assumes all the data points of distribution are generated from a mixture of a finite number of Gaussian distributions with unknown parameters [96].

$$P = f(CPU, IPC, memaccess, cachetransactions) \quad (30)$$

Table 1. Limitations of Software-based Power Models.

Power Model Equation	Work	Limitations
Mathematical Formula: SVLF [ $S\_PM$ - based]		

1 and 2	[12, 22, 27, 35, 41, 47, 57, 58, 72, 89, 92, 95, 101, 109, 122]	The model is based only on the minimum and the maximum server power consumption values and does not take into consideration the power consumption values of a server's CPU utilization between its idle and full load state.
3	[14, 56, 103]	The accuracy of the model depends on the ratio of $P_{MIN}$ to $P_{MAX}$ . If the ratio is not close to 0.7, the model gives high value of error. Moreover, the model does not consider the power consumption values for the CPU utilization values between 0% and 100%.
4	[11, 15, 25, 28, 42, 65, 69, 94, 117]	To predict a power consumption value $p$ for a CPU utilization $u$ , the model requires the power consumption values $p_1$ and $p_2$ corresponding to the CPU utilization values $u_1$ and $u_2$ respectively, such that $u_1 < u < u_2$ .
5	[63]	The model requires the power consumption values corresponding to the minimum and the maximum throughput values for each application type. Moreover, the model does not consider the applications' power consumption behavior between the minimum and the maximum throughput values.
<b>Machine Learning: MLLV - SVLR [S_PM - based]</b>		
6	[16, 17, 48, 90, 93, 123]	The model's accuracy depends on the deviation of the training data set values from the fitted regression line. Moreover, the model requires calibration for the values of $\alpha$ and $\beta$ for each server architecture type.
7	[115]	The model's accuracy depends on the deviation of the training data set values from the fitted regression line and on the increment in power consumption for idle server and server with minimum load. The higher the increment, the more will be the error. Moreover, the model requires calibration for the values of $\beta$ for each application type on each server architecture type.
8	[67]	The model's accuracy depends on the deviation of the training data set values from the fitted regression line. Moreover, the model requires calibration for the values of $\alpha$ and $\beta$ for each server architecture type.
<b>Machine Learning: MLLV - MVLR [S_PM - based]</b>		
9	[39, 104]	The model's accuracy depends on the deviation of the training data set values from the fitted regression Euclidean hyperplane. Moreover, the model requires calibration for the values of the regression parameter for each server architecture type.
10	[9]	The model's accuracy depends on the deviation of the training data set values from the fitted regression Euclidean hyperplane. The accuracy also depends on the increment in power consumption for idle server and server with minimum load. The higher the increment, the more will be the error. Moreover, the model requires calibration for the values of the regression parameter for each server architecture type.

11	[64, 75]	The model's accuracy depends on the deviation of the training data set values from the fitted regression Euclidean hyperplane. Moreover, the model requires calibration for the values of the regression parameter for each server architecture type. The model gives a high value of error for network-intensive applications.
<b>Machine Learning: MLLV - MVLR [S_PMC - based]</b>		
12	[20]	The model's accuracy depends on the deviation of the training data set values from the fitted regression Euclidean hyperplane. Moreover, the model requires calibration for the values of the regression parameter for each server architecture type.
13	[34]	The accuracy of the model depends on how close the data are to the fitted regression model. The model has high probability of over-fitting the data. Moreover, the model requires calibration for the values of the regression parameter for each server architecture type.
14	[119]	The accuracy of the model depends on how close the data are to the fitted regression model. The model has high probability of over-fitting the data. Moreover, the model requires calibration for the values of the regression parameter for each server architecture type. It requires CPU temperature hardware indicator.
<b>Machine Learning: MLLV - MVLR [S_PC - based]</b>		
15	[62]	The model's accuracy depends on the deviation of the training data set values from the fitted regression Euclidean hyperplane. Moreover, the model requires calibration for the values of the regression parameter for each server architecture type and needs to perform different transformations of the performance counters having non-linear relationship with power consumption. It requires CPU temperature hardware indicator.
<b>Mathematical Formula: SVNLF [S_PM - based]</b>		
16	[41, 92, 107]	The model is based only on the minimum and the maximum server power consumption values without considering the power for a server's CPU utilization values between its idle and full load state. Moreover, the model requires calibration for the value of $r$ for each server architecture type.
17	[76, 108]	The model is based only on the minimum and the maximum server power consumption values and requires calibration for the values of $\alpha$ and $\beta$ for each server architecture type.
<b>Machine Learning: MLNLV - SVPR [S_PM - based]</b>		
18	[61, 123]	The model's accuracy depends on the deviation of the training data set values from the fitted regression polynomial curve. Moreover, the model requires calibration for the values of the regression parameter for each server architecture type.
19	[53]	The model's accuracy depends on the deviation of the training data set values from the fitted regression polynomial curve. Moreover, the model requires calibration for the values of the regression parameter for each server architecture type.

20	[123]	The model's accuracy depends on the deviation of the training data set values from the fitted regression polynomial curve. The model generally suffers from the issue of over-fitting. Moreover, the model requires calibration for the values of the regression parameter for each server architecture type.
-	[26]	The model's accuracy depends on the deviation of the training data set values from the fitted regression polynomial curve. The model generally suffers from the issue of over-fitting. Moreover, the model requires calibration for the values of the regression parameter for each server architecture type.
<b>Machine Learning: MLNLV - LR [S_PM - based]</b>		
21	[79, 82]	The model's accuracy depends on the shrinking of the non-significant variables. Moreover, the selection of high order polynomial variables may over-fit the model making the prediction less accurate.
22	[79, 82]	The model's accuracy depends on the shrinking of the non-significant variables. Moreover, the model has high probability of over-fitting due to the use of exponential along with the polynomial function.
<b>Machine Learning: MLNLV - LR [S_PC - based]</b>		
23	[80]	The model while shrinking the non-significant variables to zero, does not consider the integrated correlation between those variables and their combined association on the power consumption.
<b>Machine Learning: MLNLV - SVM [S_PM - based]</b>		
24	[79, 82]	The model's accuracy depends on the selection of the kernel and can be computationally complex.
<b>Machine Learning: MLNLV - DNN [S_PM - based]</b>		
26	[75]	The model's accuracy depends on the size of the training data set. Moreover, the model training process is computationally complex compared to regression based models.
<b>Machine Learning: MLNLV - ANN [S_PMC - based]</b>		
29	[33]	The model's accuracy depends on the size of the training data set, number of hidden layers and the activation function used. Moreover, the model training process is computationally complex compared to regression based models. It requires CPU temperature hardware indicator.
<b>Machine Learning: MLNLV - GMM [S_PMC - based]</b>		
30	[38]	The accuracy of the model decreases and the computational complexity increases with an increasing number of variables.

Single Variable Linear with Fixed Slope and Intercept (SVLF),  
 Machine Learning Linear with Variable Slope and Intercept (MLLV),  
 Single Variable Linear Regression (SVLR), Multi Variable Linear Regression (MVLR),  
 Single Variable Non-Linear with Fixed Slope and Intercept (SVNLF),  
 \* Machine Learning Non-Linear with Variable Slope and Intercept (MLNLV),  
 Single Variable Polynomial Regression (SVPR), Lasso Regression (LR), Support Vector Machine (SVM),  
 Deep Neural Network (DNN), Artificial Neural Network (ANN), Gaussian Mixture Model (GMM),  
 System\_Performance\_Metrics (S\_PM), System\_performance\_Counters (S\_PC),  
 and System\_Performance\_Metrics\_Counters (S\_PMC)

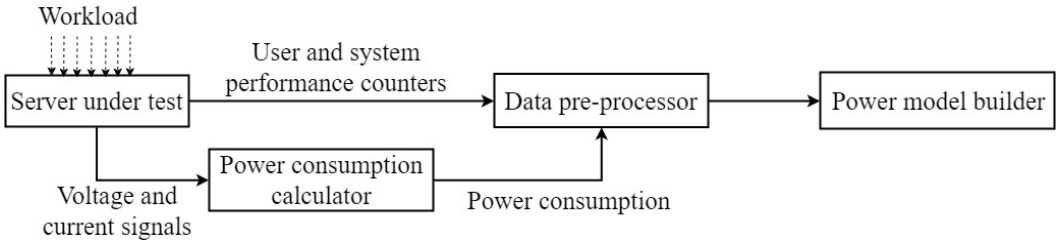


Fig. 2. Workflow of power model development.

## 2.2 Evaluated Works and Software-based Power Model Development Workflow

The methodology for power model evaluation comprises of two stages; model development and prediction. The model development stage generally known as training stage involves building the model based on power consumption values and corresponding performance counters values for some workload or representative benchmark. The power model development is specific to server architecture requiring a particular model to be trained for each different type of architecture. Once the model is developed, it is used to predict the power consumption of a server. This is known as the prediction stage. Fig. 2 shows the workflow we use to develop the software-based power models under study. The server under test is the server for which the power models are to be developed. The workload stressing different user and system performance counters runs on the server. While the workload is running, the values of the user and low-level system metrics are recorded and written in a file. Simultaneously, the voltage and the current signal values of the server are measured and sent to the power consumption calculator module. The values of the counters and the power consumption are then sent to the data pre-processor module, where they are synchronized and averaged to develop the training data set. This data set is then sent to the power model builder module which builds the power model to be used.

A similar workflow is used by the works in the literature evaluating different power models. However, the servers for model development, workload to stress the servers, and the power measurement technique used by these works are different. For instance, [9, 20, 33, 34, 38, 61, 64, 75, 79, 80, 90, 104] used a single server for power model development, while [26, 27, 39, 41, 62, 67, 74, 76, 119, 123] used multiple heterogeneous servers. Moreover, [34, 79, 80, 90] used synthetic workload to stress the server, while [9, 20, 26, 27, 33, 38, 39, 41, 62, 64, 67, 74–76, 104, 119, 123] used different benchmarking applications and real-world workload traces. In addition, [76] used DW-6090 power meter, [90] used a power analyzer, [123] used Chroma 66202 power meter, [67] used IBM active energy manager, [61] used Voltcraft Energy Logger 4000, [26, 39, 75] used a AC power meter, [9] used Yokogawa WT210 power meter, [20, 64] used WattsUp PRO ES power meter, [34] used a watt meter, [79] used a smart power meter, [62, 119] used home-brew power meter, and [33] used a plugg power meter to measure the power consumption of the server.

Table 2 summarizes different software-based power models evaluated in the literature. It shows that these power models are evaluated under different experimental environment, using different error formula, applications and power measurement techniques, which makes it difficult to compare them. As shown in the table, the errors reported by different works evaluating a similar power model are different. These discrepancies in the result are due to the use of different experimental environment, setup, and evaluation formula. To date, there is no work comparing the performance of the models examined in this study. Thus, in this work, we evaluate these models under a unified experimental environment, power measurement technique and error formula. We compare the performance on three different server architectures using a diverse set of applications.

Table 2. Evaluated Works on Software-based Power Models.

Work	Power Model Equation	Experimental Setup	Workload used to Stress the Server	Power Measurement Technique	Error Formula	Error
[41]	1	Thousand of heterogeneous servers	Webmail, Web-search, and Mapreduce	-	-	-
[27]	1	Servers from SPECpower 2008 database	SPECpower 2008 database	-	$\frac{1}{N} \times \sum_{i=1}^N \frac{ Predicted-Actual }{Actual}$	25.7%
[41]	16	Thousand of heterogeneous servers	Webmail, Web-search, and Mapreduce	-	-	1%
[76]	17	Thirteen heterogeneous servers	Streaming media workload using Windows media load simulator	DW-6090 power meter	$\frac{Actual-Predicted}{N}$	6%
[90]	6	Single server	Synthetic workload using workload generator	Power analyzer	-	9%
[123]	6	Seven heterogeneous servers	SPECpower benchmarking application	Chroma 66202 power meter	$\sqrt{\frac{(Actual-Predicted)^2}{N}}$	12.95%



[67]	8	Three heterogeneous servers	TPC-W, SPECpower, Domino, daxpy, fma, and HPL applications	IBM active manager	$\frac{ Predicted-Actual }{Actual} \times 100\%$	5%
[61]	18	Single server	-	Voltcraft Energy Logger 4000	-	9%
[123]	18	Seven heterogeneous servers	SPECpower benchmarking application	Chroma 66202 power meter	$\sqrt{\frac{(Actual-Predicted)^2}{N}}$	7.984%
[123]	20	Seven heterogeneous servers	SPECpower benchmarking application	Chroma 66202 power meter	$\sqrt{\frac{(Actual-Predicted)^2}{N}}$	3.319%
[26]	-	Six heterogeneous servers	Ibench, Stress, Sysbench, Prime 95, Linpack-neon, Pmbw, STREAM, fio, iperf3, Cloud-Suite, and NAS benchmarks	AC power meter	$\frac{100}{n} \sum_{i=1}^n \frac{(Actual-Predicted)}{Actual}$	2.6-5.7%

[39]	9	Two heterogeneous servers	SPECcpu 2000 integer, SPEC-cpu2000 floating point, SPECjbb 2000, SPECweb 2005, Streams application, and matrix multiplication	AC power meter	$\frac{1}{N} \times \sum_{i=1}^N \frac{Predicted-Actual}{Actual}$	4%
[104]	9	Four homogeneous servers	Video sharing web application	-	-	3.91%
[9]	10	Single server	scp, rsync, ftp, bbcp, and gridftp data transfer tools	yokogawa WT210 power meter	$\frac{1}{N} (\sum_{i=1}^N \frac{Predicted-Actual}{Actual}) 100\%$	6%
[64]	11	Single server	SPECcpu 2006, and Iometer	Power meter WattsUp PRO ES power meter	$\frac{ Predicted-Actual }{N}$	5%
[75]	11	Six homogeneous servers	pi, sudoku, sort, random writer, and word count Hadoop programs	AC power meter	$\frac{ Predicted-Actual }{Actual}$	4%

[20]	12	Single server	NAS-NPB, Izone, Bonnie++, BYTEmark, Cachebench, Dense matrix multiplication, and Gcc benchmark programs	WattsUp PRO power meter	-	94% accuracy
[34]	13	Single desktop	Synthetic workload	Watt meter	$\sqrt{\frac{(Actual - Predicted)^2}{N}}$	2.7%
[119]	14	Three heterogeneous servers	Abinit, NAMD, HMMER, MEncoder and CPU Burn applications, and Intel LINPACK, C-ray and Cavity benchmarks	Home-brew watt meter based around the chipset ADE7763	$\sqrt{\frac{\sum_{i=1}^N (Actual - Predicted)^2}{N}}$	1-4%
[62]	15	Four heterogeneous HPC servers	Abinit, CPU Burn, HMMER, Namd, MEncoder, FFTE, Make, Mprime, Open-FOAM and Tar applications, and Cavity and C-ray benchmarks	Home-brew device based around the chipset ADE7763	$\frac{1}{N} \sum_{i=1}^N (Actual - Predicted)^2$	1-4%

[82]	21	Single server	*	Data acquisition system and WattsUp meter	$\frac{1}{N}(\sum_{i=1}^N \frac{Predicted-Actual}{Actual})$	-
[79]	21	Single server	Synthetic workload	Smart power meter	-	10%
[82]	22	Single server	*	Data acquisition system and WattsUp meter	$\frac{1}{N}(\sum_{i=1}^N \frac{Predicted-Actual}{Actual})$	-
[79]	22	Single server	Synthetic workload	Smart power meter	-	10%
[80]	23	Single server	Synthetic workload	-	$\sqrt{\frac{(Actual-Predicted)^2}{N}}$	10%
[82]	24	Single server	*	Data acquisition system and WattsUp meter	$\frac{1}{N}(\sum_{i=1}^N \frac{Predicted-Actual}{Actual})$	-
[79]	24	Single server	Synthetic workload	Smart power meter	-	-
[74]	26	Two heterogeneous servers	WC98, and clark web application traces	-	$\frac{\sqrt{\sum (Actual-Predicted)^2}}{Standard deviation}$	1.03±0.13
[33]	29	RECS compute box having eighteen homogeneous computer modules	C0, CU, ALU, FPU, RAND, L1, L2, L3 and RAM micro benchmarks	Plogg power meter	$\frac{100}{N} \times \sum_{i=1}^N  \frac{Actual-Predicted}{Actual} $	1.83%

[38]	30	Single server	amm, app, art, cra, eon, equ, fac, fma, gap, gcc, gzi, mcf, mes, per, swi, two, vpr, and wup bench-marks	-	-	10%
------	----	---------------	--	---	---	-----

\* sleep, streamcluster, canneal, memcache, bodytrack, freqmine, x264, blackscholes, stressApp, LinuxBuild, namd, dedup, zeusmp, Bonnie, mcf, sphinx3, povray, soplex, cpuload(S\_PMC)

### 3 PERFORMANCE ANALYSIS

In this section, we analyze and compare the performance of the studied software-based power models on three different classes of servers, using various tools and benchmarks. We evaluate their performance using the standard error of estimation.

#### 3.1 Experimental Environment

We use three heterogeneous servers from our Lab located at the College of Information Technology of the United Arab Emirates University, to evaluate the performance of the studied models. The servers' specifications are listed in Table 3. We perform various experiments to generate the training, validation and testing data sets. The power models are developed using the training data set and are then validated using the validation data set. The models are then evaluated in terms of standard error of estimation using the testing data set. Table 4 shows the list of tools used to generate the training and validation data set. Table 5 shows the list of benchmarks and applications used to generate the testing data set. We run these tools and applications on each server and measure the values of different resource metrics and corresponding power consumption. To measure the value of the metrics we use Linux perf utility [51] and collectd tool [6], and to measure the corresponding power consumption values, Tektronix's TDS2012B [110] 100 MHz with 1GS/s of sampling (2-channel digital oscilloscope) was used. We connect the oscilloscope to a current probe [116] and a high differential voltage probe [116] to measure the current and the voltage signals respectively. The power consumption is then the product of the measured current and voltage signals. We also use the servers from SPEC power [31] to evaluate the performance of the power models in order to verify our evaluation on modern server architectures. We only evaluate single variable power models considering CPU utilization as the independent variable for the SPEC power servers because only the data for power consumption corresponding to CPU utilization is available on the SPEC power website. We use two servers listed in the SPEC power results for quarter 1 of 2019, whose specifications are listed in Table 6. The servers from the SPEC Power website belong to the same family of servers as the ones we use in our experimental testbed, but with different architectures and capabilities.

#### 3.2 Experiments

The set of experiments performed on the servers to obtain the training, validation, and testing data sets for the power model development and performance evaluation are discussed in this

Table 3. List of Servers Used in the Experiments.

Server 1	CELSIUS R940power 2 x Intel Xeon E5-2680v4 CPU (2.40 GHz, 14 cores), 8 x 32GB DDR4, 2 x HDD SAS 600GB, OS version Redhat Enterprise Linux Server RHEL 7.4 64-bit.
Server 2	Sun Fire Intel_Xeon CPU core of 2.80 GHz, Dual core, with 512 KB of cache and 4 GB of memory for each core, OS version CentOS 6.8(i686).
Server 3	Sun Fire X4100 with AMD_Operaton252 CPU of 2.59 GHz, dual CPU, single core, with 1MB of cache and 2GB of memory for each core, OS version Red Hat Enterprise Linux Server release 7.3 (Mapio).

Table 4. Tools Used to Generate the Training and Validation Data sets.

Tool	Resource Stressed	Description
CPU Load Generator [45]	CPU	CPU Load Generator is a script written in Python that allows generating a fixed CPU load for a finite user defined time duration. The script takes in as input the desired CPU load, the duration of the experiment and the CPU core on which the load must be generated.
Stress [5]	Memory	Stress is used to generate a configurable measure of CPU, memory and disk load. We use Stress-1.0.4 to generate configurable stress on memory. The inputs to the stress command line are the number of vm workers, memory allocation size per vm worker and the duration of the experiment.
Vdbench [112]	Disk I/O rate	Vdbench is used to generate configurable amount of disk I/O workloads on a system. We set the desired I/O rate using the curve parameter of the run definition file.
iperf3 [7, 87]	Network I/O rate	Iperf3 is a tool for active measurement of the maximum available bandwidth on IP networks. We use iperf3 between to generate a configurable network I/O rate between the test server and a remote host server.

section. The performance of the different user and system counters used by the studied power models was measured in real-time using the Linux tools. We also measure the corresponding power consumption values using a LabVIEW program. The values of the counters and power consumption are written to a file every one second and are then averaged. We repeat all the experiments 5 times and averaged the averages.

For all the models under study, except for throughput-based S\_PM model, the experiments for generating the training and validation data sets are performed by stressing the CPU, memory, disk operations, and network transfers individually on each of the three servers. We stress the CPU by generating a CPU load between 0% - 100% for 5 minutes each at random intervals, using the CPU Load Generator tool. For multi-core servers, we generate the CPU load on all the cores simultaneously. We use the Stress tool to populate the memory using random memory sizes for a virtual machine (vm) worker, for 5 minutes each. To stress the disk I/O at configurable I/O rate, we then use vdbench tool. We first find the maximum disk I/O rate of the server and then generate I/O rates between 0% - 100% of the maximum I/O rate for 5 minutes each. We stress the network I/O

Table 5. Applications and Benchmarks Used to Generate the Testing Data set.

Application/Benchmark	Resource Stressed	Description
Sysbench benchmark [10, 68]	CPU	Sysbench benchmark is used to evaluate the OS parameters like CPU utilization, memory utilization, and Disk I/O. We use the Sysbench to stress the CPU, using the CPU workload [46]. The CPU workload calculates prime numbers between zero and a specified number.
MEncoder application [8]	CPU and Memory	We used MEncoder 4.45, a video compressor application included in the Mplayer project, to stress the CPU and the memory. We use the MPEG-4 video format [4], with 1920 x 1080 resolution, 18,356.7 kbps, 23 fps, and 24 bpp.
PARSEC benchmark -Black Scholes Model (Portfolio management) [18, 111]	CPU, Memory and Disk	The Black Scholes by Intel RMS benchmark calculates the prices of European options' portfolio analytically using the partial differential equation (PDE).
Data Mining - Ensemble Clustering application [54]	CPU, Memory and Disk	We use Weka 3.8 [54] to perform k-means clustering of the forest cover [43] data set consisting of Geospatial descriptions of various forest's types. The data contains 581,000 instances, 7 classes, and 54 attributes.
PARSEC benchmark -Streamcluster [18, 111]	CPU, Memory, Disk and Network	Streamcluster is a part of the PARSEC 3.0 benchmark suite to solve the online clustering problem. Stream clustering is memory intensive for low-dimensional data and becomes CPU intensive as the dimension increases.

Table 6. List of Servers from SPEC Power Used in the Experiments.

SPEC_Server 1	Dell PowerEdge R7425, AMD EPYC 7601 2.20 GHz, 32 core, 64 MB L3 cache, 16x8 GB of memory, 240 GB SATA SSD, and OS Microsoft Windows Server [29].
SPEC_Server 2	Lenovo Global Technology ThinkSystem SR150, Intel Xeon E-2176G, 6 core, 3.7 GHz, 12MB L3 cache, 2x16 GB of memory, 128 GB M.2 SSD, and OS Microsoft Windows Server [30].

rate by specifying the desired network bandwidth between the test server and a remote desktop. We measure the maximum available bandwidth for the server using `iperf3` and then ping the remote desktop with random bandwidth between 0% - 100% of the maximum bandwidth for 5 minutes each.

For the throughput-based  $S_{PM}$  power model, we generate the training and validation data sets as follows. We use different tools to mimic different resource-intensive applications. We measure the maximum throughput that can be achieved by an application and run it with random throughput varying between the minimum and maximum. We use the CPU load of Sysbench benchmark to mimic a CPU- intensive application with throughput represented as the floating operations per

second. For disk-intensive applications, we use vdbench tool with the throughput represented as the number of disk reads/writes per second. For the network-intensive applications, we use iperf3 tool with the throughput represented as the number of data transferred/received per second. For the evaluation of the power models based only on CPU utilization for the servers from SPEC power, we used the SPEC power results of power consumption corresponding to different level to CPU utilization for each of the SPEC power servers.

For all the models under study, the training and validating data sets set are selected randomly using 70% and 30% of the generated experimental data set respectively.

To generate the testing data set, we run Sysbench for the CPU workload to calculate the prime numbers up to 20000000 with the number of threads increasing randomly from 0 up to the total number of threads of the server under test. In addition, we run the MEncoder application to compress a video file of size 100MB. We repeat the process for video files of sizes 200MB-2GB, with an increment of 100MB. Furthermore, we use the Black Scholes application to calculate the prices of a 65,536 European options portfolio. We also use the ensemble clustering application to perform k-means clustering of data sets with a different number of instances. We use 4 different sizes of 7.38MB, 74.2MB, 746MB, and 941MB having 27900, 279000, 2790000, and 5580000 instances respectively, form the UCI Forest data repository. Moreover, we run the Streamcluster application from the PARSEC benchmark suite to perform online stream clustering for native input options having 1,000,000 input points and 128 dimensions. The power models performance for the servers from SPEC power is not evaluated as the testing data set using different benchmarking applications for those servers can not be obtained as they are not part of our experimental testbed.

We use the R programming language [44] to develop the studied power models using the generated training data set and to evaluate their performance using the validation and testing data sets. The performance of the models is analyzed using standard error of estimation calculated using Equation 31.

$$e_{est} = \sqrt{\frac{\sum_{i=1}^n (P_i - P'_i)^2}{n}} \quad (31)$$

Where  $P$  and  $P'$  are the actual and predicted values of power consumption respectively and  $n$  is the length of the testing data set.

### 3.3 Experimental Results Analysis

In this section, we discuss the results obtained by the works on software-based power models when evaluated by those works under the same experimental environment and setup and compare them with our results. We also analyze our experimental results and give insights and conclusions of these evaluations. In particular, we reveal the reasons behind the performance of these models.

#### 3.3.1 Analysis of the Evaluated Works on Software-based Power Models in the Literature

Table 7 shows the results for prediction errors of different power models obtained by the works in the literature using a unified setup. [123] evaluated the SVLR model in Equation 6 and SVPR models in Equation 18 and Equation 20 and reported an error of 12.95%, 7.98% and 3.32% respectively. These results indicate that 3rd order SVPR model (Equation 20) outperforms the 2nd order SVPR (Equation 18) and SVLR (Equation 6) models. [79] evaluated and compared the Equations 21, 22, and 24 and showed that the SVM model in Equation 24 has the least error, while the model in Equations 21 and 22 has almost the same error. The GMM model in Equation 30 was evaluated and compared to the SVLR and MVLR models by [38]. The results showed that GMM has an error of 10%, compared to SVLR model in Equation 6 having an error of 50%. To our knowledge, there is



Table 7. Results of Power Prediction Errors Obtained by the Works Comparing Models in the Literature

Work	Power Model Equation	Error
[123]	6	12.9%
	18	7.9%
	20	3.3%
[79]	21	10.0%
	22	10.0%
	24	<10.0%
[38]	6	50.0%
	30	10.0%

no work which compares the software-based power models in the literature. In the sections that follow, we compare and analyze these models using a unified experimental setup, workload, and error calculation formula.

### 3.3.2 Analysis of our Experimental Results

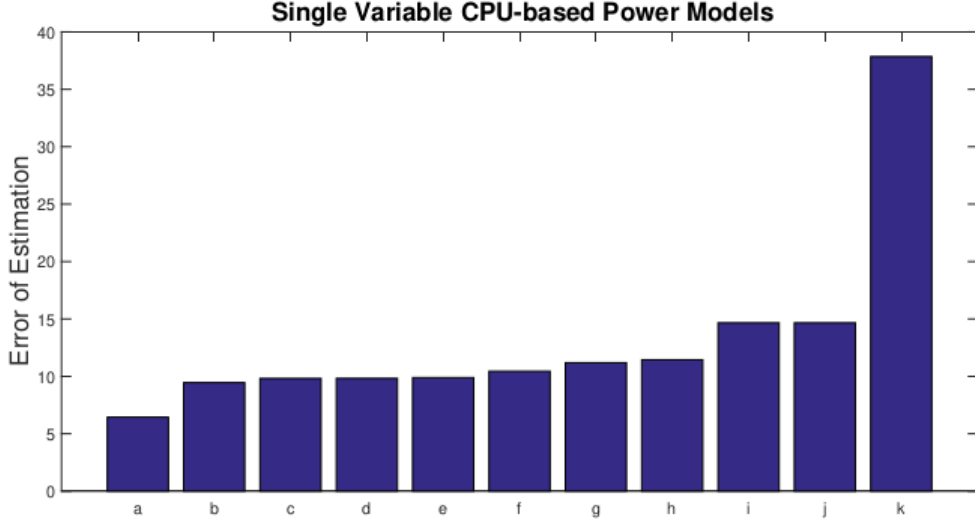
In this section, we evaluate the performance of the power models under study for the generated validating data sets and the testing data set. In particular, we compare the performance among the CPU-based models, among the throughput-based models, and among the multi variable models for the validating data sets. In addition, we compare the performance among all the models for the testing data set.

#### 3.3.2.1 Single Variable CPU-based Power Models for Validating Data Set

Figure 3 shows the standard error of estimation of the single variable CPU-based power models for the validating data set for Server 1. It shows that the interpolation model has the least error of estimation, followed by the models based on SVPR, SVLR, SVNLF, and SVLF. This is thanks to the piece-wise linearization between every two data points of the training data set resulting in a better prediction.

Comparing the performance of SVPR with SVLR models, the error of estimation with the SVPR models is less, which is also confirmed by the evaluation results in the literature (Table 7). This is because, for a server, the power consumption profile corresponding to the CPU utilization values fits well to a curve rather than a linear line. Among the SVPR models, the 3rd order has the least error compared to the 2nd and  $r$ th order. This is because the power consumption behavior of the server is an increasing function at the endpoints which can be more accurately represented using a 3rd-degree polynomial curve with the end-points moving in the same direction. Whereas, in a 2nd-degree polynomial curve, the endpoints move in the opposite direction resulting in a higher error.

The error of SVLF and SVNLF power models is more compared to that of SVLR models (Figure 3). This is because the SVLF and SVNLF are based only on the endpoint power consumption values,  $P_{MAX}$  and  $P_{MIN}$ , to construct a line where all the possible predicted values will lie. Therefore, they do not consider the implications of other power consumption data between the endpoints for predictions. However, the SVLR models compute a linear regression line to best fit the data distribution while minimizing the sum of the squares of the vertical regression deviations. For the SVLR, the model with the fixed intercept ( $P_{MIN}$ ) has a higher error compared to the model with dynamic intercept. This is because there is a sudden change of slope in the power consumption



a	$P_1 + (\Delta \times \frac{(u_{CPU} - u_{1CPU})}{\times} 100)$	g	$P_{MIN} + \beta_1 u_{CPU}$
b	$\alpha + \beta_1 u_{CPU} + \beta_2 u_{CPU}^2 + \beta_3 u_{CPU}^3$	h	$((P_{MAX} - P_{MIN}) \times (2u_{CPU} - u_{CPU}^r)) + P_{MIN}$
c	$\alpha + \beta_1 u_{CPU} + \beta_2 u_{CPU}^r$	i	$(k \times P_{MAX}) + ((1 - k) \times P_{MAX} \times u_{CPU})$
d	$\alpha + \beta_1 u_{CPU} + \beta_2 u_{CPU}^2$	j	$(P_{MAX} - P_{MIN}) \times u_{CPU} + P_{MIN}$
e	$\alpha + \beta_1 u_{CPU}$	k	$P_{MAX} \times (0.7 + 0.3u_{CPU})$
f	$((P_{MAX} - P_{MIN}) \times (\alpha u_{CPU}^\beta)) + P_{MIN}$		

Fig. 3. Error of estimation of single variable CPU-based power models for validation data set.

trend for the CPU utilization value at 0% and at values greater than 1%. This change is not captured by the regression model having fixed intercept, consequently having more prediction error.

Comparing the performance of SVLF with SVNLF power models, the SVLF has more error. This is because SVLF models do not capture slight non-linearity of the power data distribution over the range of CPU utilization values. Consequently, the straight line computed by the SVLF models for predictions has a high value of offset compared to the SVNLF models. The SVLF model assuming that  $P_{MIN}$  is 70% of  $P_{MAX}$  has the maximum error among all the single variable power models. The rationale behind that is this assumption, which does not hold true for each class of the server. The higher the deviation from the assumption, the higher will be the error.

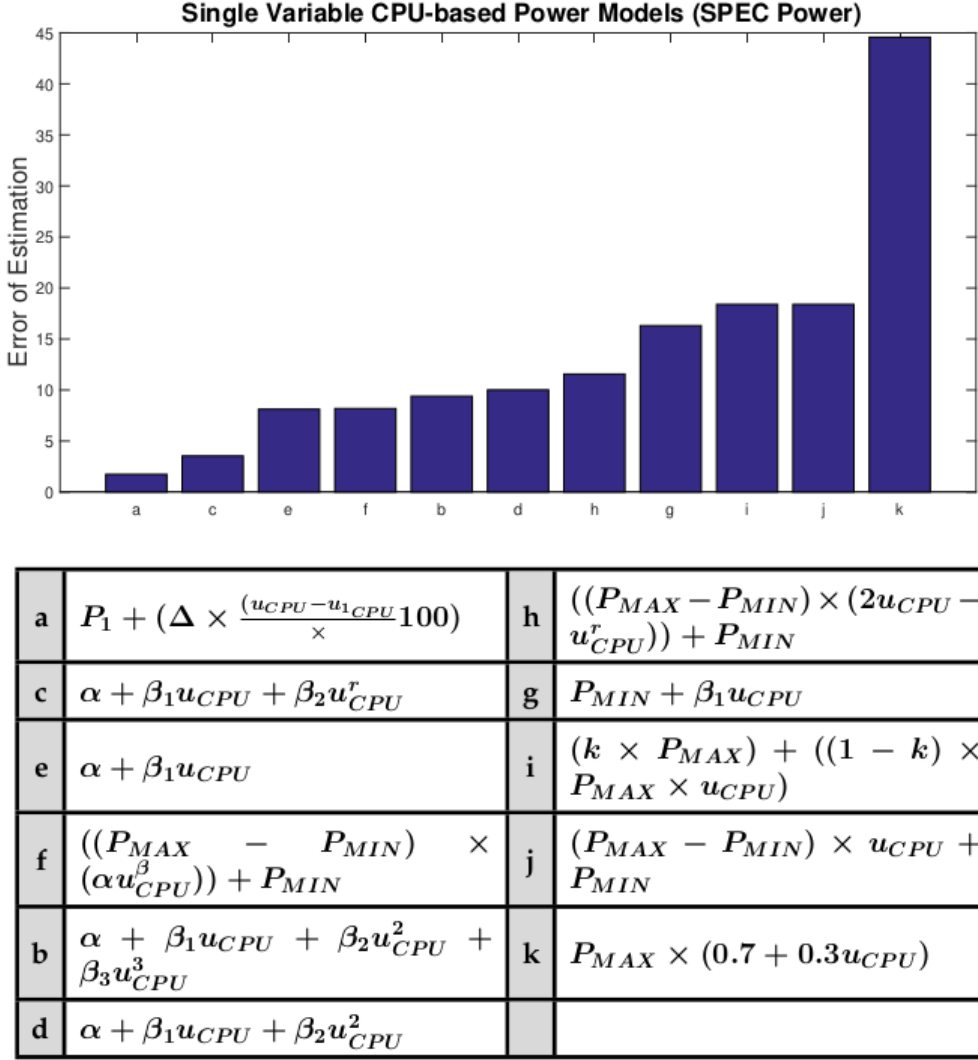


Fig. 4. Error of estimation of single variable CPU-based power models for SPEC power data set.

Figure 4 shows the error of estimation of the single variable CPU-based power models for the SPEC power servers. The performance of the models remains the same as that of the servers used in our experimental testbed. However, the performance of the SVLR is better than that of SVPR models using the SPEC power servers. This is because the power consumption profile is linear with the CPU utilization for SPEC power servers while for our servers the power profile fits better to polynomial curve than a linear line. This indicates that the performance relative performances of SVLR and SVPR depend on the server power profile. The performance of the remaining models which are CPU-based is the same on SPEC Power and our experimental testbed (Figures 3 and 4).

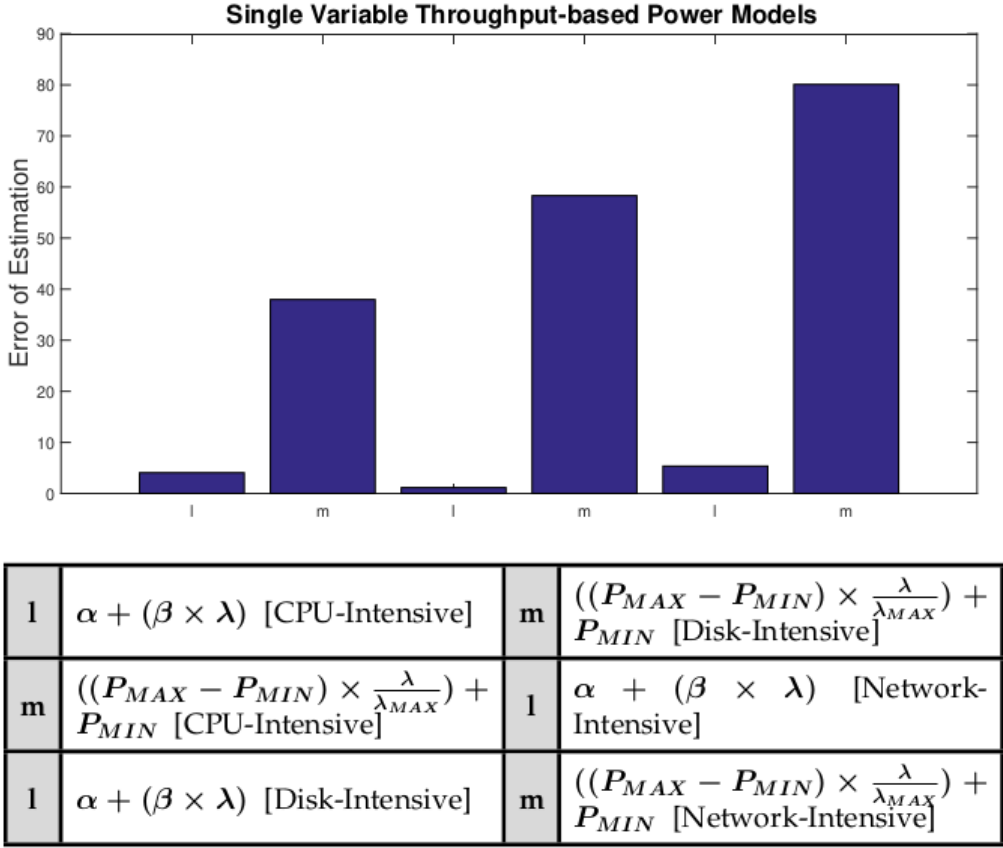


Fig. 5. Error of estimation of single variable throughput-based power models for validation data set.

### 3.3.2.2 Single Variable Throughput-based Power Models for Validating Data Set

Figure 5 shows the error of estimation for the throughput-based power models when evaluated using the validation data set generated by running CPU, memory and disk-intensive applications with varying throughput. It shows that for all application types the error of estimation of SVLF is greater than that of the SVLR model. This is because SVLF models only consider the power consumption values of the endpoints CPU utilization and do not take into consideration the spatial distribution and behavior of the power data. The SVLR model based on throughput also outperforms the models based only on CPU utilization, except the interpolation model. This is because the throughput based model considers the impact of each underlying resource of the server environment that contributes to the power consumption.

### 3.3.2.3 Multi Variable Power Models for Validating Data Set

Our results (Figure 6) for the standard error of estimation of multi variable power models for the validating data set for Server 1 shows that the SVM model based on CPU and memory utilization has the least error of estimation compared to other evaluated multi variable power models. This is

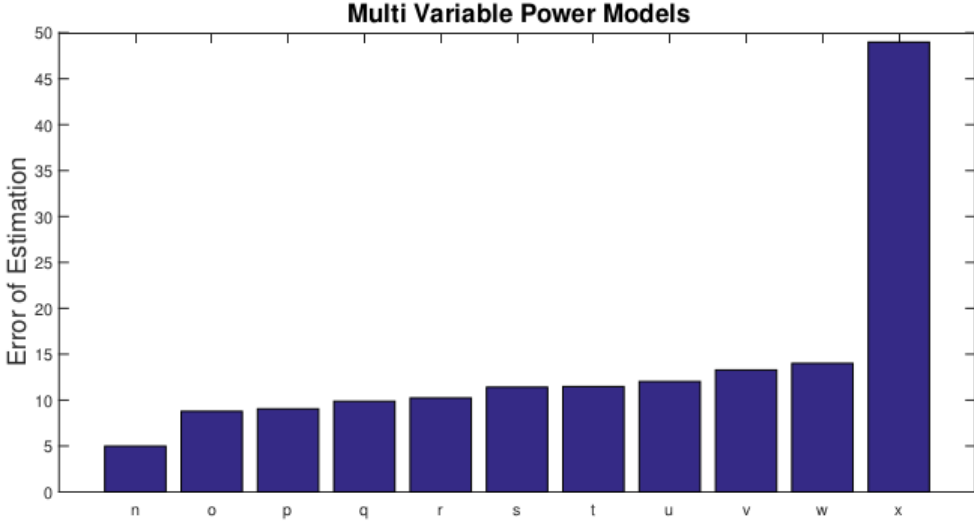
because of the server's non-linear power profile and cross-dependency between the variables. SVM considers the variable dependency and transforms the non-linear data into a high dimensional feature space acknowledging the presence of non-linearity and gives precise predictions compared to other multi variable linear models where the variables are assumed to be independent. The non-linear polynomial and polynomial+exponential power models with lasso have almost similar performance with the second least error after SVM. These models consider the quadratic and cubic functions of the CPU and memory utilization values resulting in a regression hyperplane that fits close to the actual values. Consequently, these models have less error compared to the models considering only the linear functions of resource utilization.

The error of estimation of DNN model is less compared to that of the power models based on GMM and MVLR models. This is thanks to the use of recursive autoencoder in the neural network power model. The recursive autoencoder model generates an encoder output as a function of the current data point and previous encoder output. Consequently, the recursive autoencoder will generate a dynamically varying prediction line as a time series minimizing the prediction offset. The better performance of GMM compared to the MVLR models (also confirmed by the evaluation in the literature as stated in Table 7) is because that GMM considers the interaction of different variables resulting in various levels of power consumption instead of having a single linear hyperplane representing the power.

Comparing the performance of different MVLR models, the power models based on CPU, memory and disk has less error compared to the models including network and cache in addition (Figure 6). This is because the inclusion of variables that are not significant for power consumption will over-fit the regression model causing a high offset between the fitted and the actual values. The model including the cache in addition to CPU, disk and memory has less error compared to the network inclusive model. This is because the cache transaction is reflected by the utilization of memory contributing to the power consumption and thus yielding accurate predictions than the model including network instead of cache. The error of MVLR model with fixed intercept is more compared to the MVLR models with dynamic intercept having at most 4 independent variables. The rationale behind this is the sudden change of slope in the power consumption trend for an idle and utilized server, which is not modeled when using a fixed intercept. The performance of the throughput-based models (Figure 5) is better than the MVLR models because the MVLR models do not capture all the underlying performance counters that contribute to the power which is however captured by the application throughput.

Figure 6 shows that the MVLR model with 9 independent variables has the second-highest error and the LR model with 30 variables has the highest error among all the evaluated models. This is because the model with 9 variables includes context switch and interrupt requests, which do not contribute to the power consumption majorly. The worst performance of the LR model with 30 variables is because instead of considering the CPU utilization, the model takes 28 different low-level performance counters contributing to CPU utilization, each of them contributes to the power consumption. The lasso regression only selects some significant power contributors while shrinking the remaining. Consequently, the model leads to a high prediction error as the relationship between the rejected metrics and the power consumption is not modeled.

In summary, interpolation model has the least error of estimation for the single variable power models when evaluated using the validation data set, while the model assuming the idle power to be 70% of the server's peak power has the maximum error of estimation. For the multi variable power models, SVM has the best performance with least error, while the lasso regression model with 30 variables has the maximum error of estimation. The errors of estimation for interpolation, a model assuming idle power as 70% of peak power, SVM and lasso regression with 30 variables are



n	$f(x) = w\phi(x) + b$ , (CPU and memory)	t	$\alpha + \beta_1 u_{CPU} + \beta_2 u_{mem} + \beta_3 u_{disk} + \beta_4 u_{cache}$
o	$\phi(.) = x_i^a, 1 \leq x \leq 3$ , (CPU and memory)	u	$\alpha + \beta_1 u_{CPU} + \beta_2 u_{mem} + \beta_3 u_{disk} + \beta_4 u_{net}$
p	$\phi(.) = e^{x_i^a}, 1 \leq x \leq 3$ , (CPU and memory)	v	$P_{MIN} + \beta_1 u_{CPU} + \beta_2 u_{mem} + \beta_3 u_{disk} + \beta_4 u_{net}$
q	$\epsilon_{RAE} = (\epsilon_{PRD} \times 0.95) + (\epsilon_{AE} \times 0.05)$ , 7 variables contributing to CPU, memory, disk and network	w	$P_{MIN} + \sum_{n=1}^9 \beta_n C_n$ , 9 variables contributing to CPU, memory, disk and network
r	f(CPU, IPC, memaccess, cachetransaction)	x	Lasso regression, 30 variables contributing to CPU, memory, disk and network
s	$\alpha + \beta_1 u_{CPU} + \beta_2 u_{mem} + \beta_3 u_{disk}$		

Fig. 6. Error of estimation of multi Variable power models for validation data set.

6.44, 37.85, 4.98 and 48.98 respectively. Our experiments show that the relative performance of the models remains the same for servers 2 and 3 used in the experimental setup.

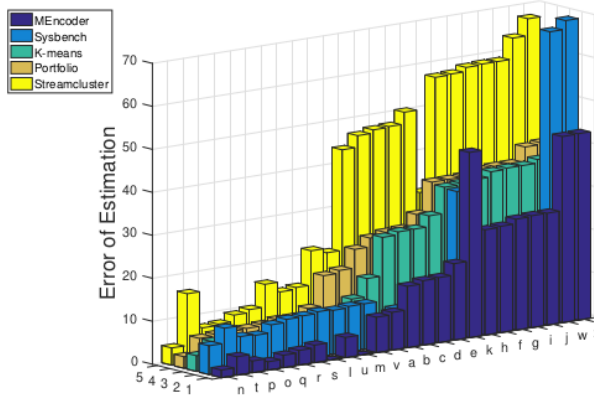
### 3.3.2.4 Software-based Power Models for Testing Data Set

Figure 7 shows the performance of all the software-based power models when evaluated for the testing data set, i.e., real applications. For the CPU-intensive Sysbench application, it is expected that the single variable power models considering CPU utilization should perform better than the multi variable models. Our results (Figure 7) show that the interpolation model has the least error of estimation compared to the other models because of its piece-wise linearization approach as discussed previously. The SVM, polynomial with lasso and polynomial+exponential with lasso models still outperforms other models in terms of error of estimation. This is because of the acknowledgment of non-linearity by the SVM model and the inclusion of quadratic and cubic functions by the polynomial and polynomial+exponential models. Comparing the performance of MVLR models having at most 4 independent variables with SVLF and SVNLF models, the non-regression models considering only the power consumption values at the end points have more error of estimation. The LR model with 30 variables has the worst performance with the maximum error of estimation compared to other evaluated models.

Regarding the CPU and memory-intensive application MEncoder, the performance of multi variable models is better than that of the single variable models. This is thanks to the inclusion of memory utilization while modeling the power consumption by the multi variable models. For the MEncoder application, SVM model has the least error while the lasso regression model with 30 variables has the maximum error of estimation. Figure 7 shows that for CPU, memory and disk-intensive K-means application, the overall performance of all the models remains the same except the relative performance of MVLR model with CPU, memory and disk, and model with CPU, memory, disk and cache. It shows that the model including the cache has less error because K-means application generates cache references contributing to power consumption considered by the power model, thus leading to more accurate predicted values. For the portfolio application with more cache transactions, the multi variable regression model with cache outperforms the polynomial with lasso and polynomial+exponential with lasso models, with SVM having the least error while the lasso regression model with 30 variables having the maximum error.

Comparing the performance of the MVLR models with at most 4 independent variables, the model that includes network utilization has the least error of estimation for the Streamcluster application. This is because the application performs online clustering utilizing the network contributing to a small amount of power consumption (Figure 8). Consequently, the model with network utilization in addition to CPU, memory and disk models the relationship between resource utilization and power consumption more precisely, leading to less error. The overall performance of other models still remains the same for the Streamcluster application.

The error of estimation for the power models based on CPU utilization is higher than the multi variable models and application's throughput-based models. This is because the correlation between the power and the performance counters other than CPU is not considered in the models based only on CPU utilization. Figure 8 shows the power consumed by server 1 with increasing values of CPU, memory, disk and network utilization. It shows that the power consumption of the server is dominated by CPU utilization. It shows that at 100% CPU load the server consumes 302W of power. However, the memory consumes 200W independent of the memory load. This consumption is higher the server power consumption at idle state. The increase in power consumption with disk and network utilization is not significant. The maximum power consumption for 100% disk utilization is 173W and with 100% network utilization is 178W. Consequently, the models not considering memory utilization have a high error of estimation compared to the ones considering the memory.



n	$f(x) = w\phi(x) + b$ , (CPU and memory)	b	$\alpha + \beta_1 u_{CPU} + \beta_2 u_{CPU}^2 + \beta_3 u_{CPU}^3$
t	$\alpha + \beta_1 u_{CPU} + \beta_2 u_{mem} + \beta_3 u_{disk} + \beta_4 u_{cache}$	c	$\alpha + \beta_1 u_{CPU} + \beta_2 u_{CPU}^r$
p	$\phi(\cdot) = e^{x_i^a}$ , $1 \leq x \leq 3$ , (CPU and memory)	d	$\alpha + \beta_1 u_{CPU} + \beta_2 u_{CPU}^2$
o	$\phi(\cdot) = x_i^a$ , $1 \leq x \leq 3$ , (CPU and memory)	e	$\alpha + \beta_1 u_{CPU}$
q	$\epsilon_{RAE} = (\epsilon_{PRD} \times 0.95) + (\epsilon_{AE} \times 0.05)$ , 7 variables contributing to CPU, memory, disk and network	k	$P_{MAX} \times (0.7 + 0.3 u_{CPU})$
r	f(CPU, IPC, memaccess, cachetransaction)	h	$((P_{MAX} - P_{MIN}) \times (2u_{CPU} - u_{CPU}^r)) + P_{MIN}$
s	$\alpha + \beta_1 u_{CPU} + \beta_2 u_{mem} + \beta_3 u_{disk}$	f	$((P_{MAX} - P_{MIN}) \times (\alpha u_{CPU}^\beta)) + P_{MIN}$
l	$\alpha + (\beta \times \lambda)$	g	$P_{MIN} + \beta_1 u_{CPU}$
u	$\alpha + \beta_1 u_{CPU} + \beta_2 u_{mem} + \beta_3 u_{disk} + \beta_4 u_{net}$	i	$(k \times P_{MAX}) + ((1 - k) \times P_{MAX} \times u_{CPU})$
m	$((P_{MAX} - P_{MIN}) \times \frac{\lambda}{\lambda_{MAX}}) + P_{MIN}$	j	$(P_{MAX} - P_{MIN}) \times u_{CPU} + P_{MIN}$
v	$P_{MIN} + \beta_1 u_{CPU} + \beta_2 u_{mem} + \beta_3 u_{disk} + \beta_4 u_{net}$	w	$P_{MIN} + \sum_{n=1}^9 \beta_n C_n$ , 9 variables contributing to CPU, memory, disk and network
a	$P_1 + (\Delta \times \frac{(u_{CPU} - u_{1CPU})}{x} 100)$	x	Lasso regression, 30 variables contributing to CPU, memory, disk and network

Fig. 7. Error of estimation of software-based power models using testing data set for different applications.

The results obtained in our experiments can not be compared with that obtained in the past due to discrepancies in the experimental setup, environment, and workloads. However, we compare the relative performance of the models that are evaluated under the same setup in the past with the results from our experiments. Similar, to evaluation result by [123] the relative performance of the models in Equations 6, 18, and 20 remains the same in our results. Equation 20 has the least error of estimation compared to Equations 6 and 18. The relative performance of Equations 21, 22 and 24 reported by [79] is also confirmed in our experimental results with SVM model (Equation



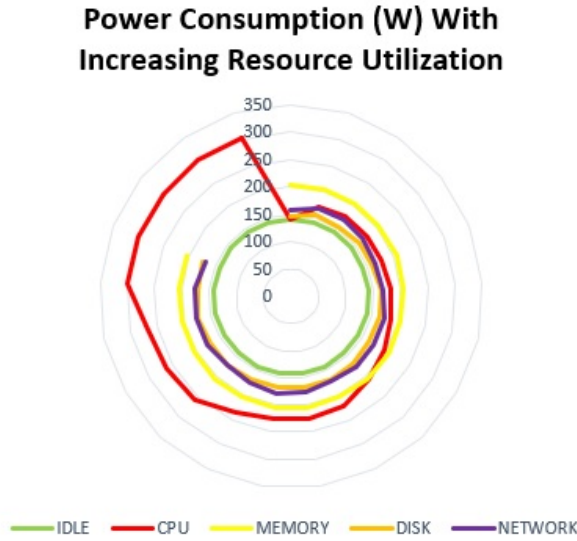


Fig. 8. Power consumption (W) of server 1 for increasing utilization of CPU, memory, disk and network.

24) having the least error of estimation. According to the results obtained by [38], GMM model (Equation 30) has least error compared to the SVLR and MVLR models. This is also confirmed in our results. The results of the works evaluating single power models in the past can not be compared with the results in the literature and with our results. This is due to the use of different experimental setup, environment, power measuring technique, error calculation formula, and workloads. Thus in this paper, we evaluated the software-based power models in a unified setup to have a qualitative comparison between them.

In summary for the Sysbench application, interpolation has the least error of 2.60, while the lasso regression model with 30 variables has the maximum error of 68.98. For the MEncoder, K-means, portfolio and Streamcluster applications, the SVM has the least error of 1.78, 3.66, 2.70, and 4.02 respectively, while the lasso regression model with 30 variables has the maximum error of 43.48, 36.60, 39.11, and 67.23 respectively. Our experiments reveal that the relative performance of the models remains the same for servers 2 and 3.

#### 4 RELATED WORKS

In the last decade, there have been many research efforts both by the academic and the industrial researchers aimed at reducing the computing infrastructure's energy consumption from the circuit level to the data center level. Power consumption modeling at different levels in a data center has then been proposed in the past for energy efficient designing and optimization, to curb the increasing energy consumption. Several works have proposed power models to be used either for simulation as a tool in designing energy-efficient data centers [36, 39, 41, 57, 61, 73, 88, 90, 93], or for server-level optimization [9, 12, 14, 22, 35, 36, 39, 41, 58, 72, 73, 78, 88, 90, 92, 95, 101, 122]. The power models can be classified as: hardware-based, using variables such as server fan speed, voltage, current, capacitor, motherboard components, and resistance for modeling [40, 55, 59, 85, 99, 102, 120] and software-based, using variables such as utilization of CPU, memory, disk, network, throughput,

interrupts, cache transactions and disk file system for modeling [9, 11, 12, 14–17, 19, 20, 22, 25–28, 33–35, 38, 39, 41, 42, 47, 48, 53, 56–58, 61–65, 67, 69, 72, 74–76, 80, 90, 92–95, 101, 103, 104, 107–109, 115, 117, 119, 122, 123]. In this paper, we focused on software-based computing server’s power models. These models include modeling based on different linear [9, 11, 12, 14–17, 20, 22, 25, 27, 28, 34, 35, 39, 41, 42, 47, 48, 56–58, 62–65, 67, 69, 72, 75, 90, 92–95, 101, 103, 104, 109, 115, 117, 119, 122, 123] and non-linear [26, 33, 38, 41, 53, 61, 74, 76, 79, 80, 82, 92, 107, 108, 123].

Despite the increasing interest in the energy consumption issue of the data centers, little work has been done to systematically analyze and compare the performance of different software-based power consumption models. These models in the literature are evaluated under different environment, experimental setup and analyzed using variants of formula to calculate the error. To date, there have been relatively very few surveys conducted for server level software-based power consumption modeling. Rivoire et al. proposed a power consumption model for servers and compared its performance with four other power models in a unified setup using a diverse set of applications [39, 97]. However, a key limitation of this work is that it fails to be comprehensive and only compares power models proposed before 2008. Möbius et al. provided a comprehensive survey of different power models for predicting power or single-core or multi-core processors, virtual machines, and entire server [86]. In addition, the work extracted the factors affecting the estimation error of the power models based on the literature review. Dayarathna et al. in the year 2016, conducted a survey of different energy consumption modeling techniques covering more than 200 models [37]. Though providing a detail literature review of different power models, the surveys [37, 86] lacks a comparative performance evaluation of the studied models. Lin et al. in the year 2018, reported on the performance of different power models for disk, CPU and memory individually in a cloud system [77]. However, the comparison does not involve the models for the overall server power consumption. In this work, we conducted the evaluation of software-based power models in a unified setup.

## 5 Conclusion

The surging data center energy consumption with the rapid popularity of cloud services, big data analysis and IoT has led to crucial economic and environmental issues. An increasing amount of research on power optimization for energy efficient designing and resource management has thus gained major attention in recent years. Power modeling and prediction at different levels of data center plays a vital role in this context. Many works on power modeling have been proposed in the literature aiming towards energy efficient computing. Those models were evaluated using different experimental setup, benchmarking applications, power measurement technique and error calculation formula, which makes it difficult to compare their relative performance. To our knowledge, this is the first work presenting a survey and comparative analysis of these models in a unified setup.

In this study, we present taxonomy and comparative evaluation of state-of-the-art software-based server power consumption models under a unified experimental setup. For that purpose, we perform a series of experiments on three different server architectures. The evaluation uses nine different tools and benchmarking applications having diverse resource utilization, for model development and evaluation.

Our experimental results show that among the single variable power models, interpolation has the least error while among the multi variable ones, SVM power model has the least error of estimation. Comparing the overall performance for the different applications, the interpolation model gives the least error for CPU-intensive application, while SVM model gives the least error for CPU+memory, CPU+memory+disk, and CPU+memory+disk+network-intensive applications. The lasso regression with 30 variables performs the worst with a maximum error of estimation for

all the studied application types. Our experiments reveal that the relative performance of these models remains on different server architectures.

When developing/using power consumption models in a computing environment, the following requirements should be considered.

- (1) *Linear versus non-linear models*: The accuracy of the linear (regression) models mainly depends on the selection of features significantly related to power consumption which requires domain knowledge. Moreover, linear models assume that the selected features have no correlation, which might not be the case. On the other hand, the non-linear models such as SVM can capture the correlation between the features which results in less error of estimation.
- (2) *CPU utilization dominance*: Most often the server's power is represented as a function of its CPU utilization, considering CPU to be dominant power consumer. For the applications that are not CPU-intensive, this assumption breaks down. It is advisable to consider at least memory utilization as it is the second most power consuming resource after CPU. Memory utilization refers to accessing the Dynamic Random-Access Memory (DRAM) for requests which are not served by the three levels of cache (L1, L2 and L3). The power consumption is directly related to the DRAM access through its controller. However, when the memory used by an application is distributed across multiple memory controllers for better throughput, the DRAM accesses through controllers will increase. This may lead to more power consumption as the number of accessed memory controllers increases. Consequently, the impact of memory controllers should be considered in power models.
- (3) *Server's idle power*: The idle power varies with the server architecture and assuming it to 70% of its peak power, by one of the power models under study, may lead to drastically misleading predictions. The more the server is energy efficient, the less is its idle power compared to the peak power. But, an energy-aware scheduler might avoid placing the task on the energy efficient server predicting its energy consumption based on the assumption of idle power to be 70% of its peak.
- (4) *Kernel Function*: The selection of the kernel for the SVM model should be done efficiently to yield most accurate results with least complexity. The kernel function selection is dependent on the behavior of the training data set.
- (5) *Quadratic and cubic utilization functions*: Combinations of linear, quadratic and cubic functions of different performance counters, selected using different variable selection models should be first used to select the variables representing the power consumption with high correlation. The selected variables should be then used for model development.
- (6) *Throughput versus performance counters*: The throughput-based power model has a better performance than the MVLR models. However, the throughput-based requires calibrations of the regression coefficients for every application with a different throughput unit, for each server architecture type. The MVLR models can be trained periodically for each server type.

For Future research work, we propose investigations in the following directions. First, we would like to investigate a multi-objective scheduling algorithm in conjunction with an energy model to optimize the energy consumption, performance and the quality of services in the data centers. Second, it would be valuable to compare the performance of software-based and hardware-based power models used in the literature of power modeling.

## Acknowledgments

This work is supported by the Emirates Center for Energy and Environment Research of the United Arab Emirates University under Grant 31R101. The authors would like to thank the anonymous reviewers for their valuable comments which helped us improve the content, quality, and presentation of this paper.

## References

- [1] [n.d.]. Branch (computer science) - Wikipedia. [https://en.wikipedia.org/wiki/Branch\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Branch_(computer_science)). (Accessed on 11/17/2019).
- [2] [n.d.]. CPU cache - Wikipedia. [https://en.wikipedia.org/wiki/CPU\\_cache](https://en.wikipedia.org/wiki/CPU_cache). (Accessed on 11/17/2019).
- [3] [n.d.]. Interrupt - Wikipedia. <https://en.wikipedia.org/wiki/Interrupt>. (Accessed on 11/17/2019).
- [4] 2009. What is H.264 | H264info.com. Retrieved April 25, 2019 from <http://www.h264info.com/h264.html>
- [5] 2014. *Stress*. Retrieved April 25, 2019 from <https://people.seas.harvard.edu/~apw/stress/>
- [6] 2019. *Collectd - The system statistics collection daemon*. Retrieved April 25, 2019 from <https://collectd.org/>
- [7] 2019. *iPerf*. Retrieved April 25, 2019 from <https://iperf.fr/iperf-download.php>
- [8] 2019. *MPlayer - The Movie Player*. Retrieved April 25, 2019 from <http://www.mplayerhq.hu/design7/news.html>
- [9] Ismail Alan, Engin Arslan, and Tefvik Kosar. 2014. Energy-aware data transfer tuning. *Proceedings - 14th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2014* (2014), 626–634. <https://doi.org/10.1109/CCGrid.2014.117>
- [10] Kopytov Alexey. 2019. *GitHub - akopytov/sysbench: Scriptable database and system performance benchmark*. Retrieved April 25, 2019 from <https://github.com/akopytov/sysbench>
- [11] Ehsan Arianyan, Hassan Taheri, and Saeed Sharifian. 2015. Novel energy and SLA efficient resource management heuristics for consolidation of virtual machines in cloud data centers. *Computers and Electrical Engineering* 47 (2015), 222–240. <https://doi.org/10.1016/j.compeleceng.2015.05.006>
- [12] Zahra Bagheri and Kamran Zamanifar. 2014. Enhancing energy efficiency in resource allocation for real-time cloud services. *2014 7th International Symposium on Telecommunications, IST 2014* (2014), 701–706. <https://doi.org/10.1109/ISTEL.2014.7000793>
- [13] Jenny A Baglivo. 2005. *Mathematica laboratories for mathematical statistics: Emphasizing simulation and computer intensive methods*. Vol. 14. Siam.
- [14] Anton Beloglazov, Jemal Abawajy, and Rajkumar Buyya. 2012. Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. *Future Generation Computer Systems* 28, 5 (2012), 755–768. <https://doi.org/10.1016/j.future.2011.04.017>
- [15] Anton Beloglazov and Rajkumar Buyya. 2012. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. *Concurrency Computation Practice and Experience* 24, 13 (2012), 1397–1420. <https://doi.org/10.1002/cpe.1867> arXiv:1006.0308
- [16] J.Ll. Berral, Í. Goiri, R. Nou, F. Julià, J. Guitart, R. Gavalda, and J. Torres. 2010. Towards energy-aware scheduling in data centers using machine learning. In *1st International Conference on Energy-Efficient Computing and Networking*, Vol. 2. 215–224. <https://doi.org/10.1145/1791314.1791349>
- [17] J L Berral, R Gavalda, and J Torres. 2011. Adaptive Scheduling on Power-Aware Managed Data-Centers Using Machine Learning. In *2011 IEEE/ACM 12th International Conference on Grid Computing*. 66–73. <https://doi.org/10.1109/Grid.2011.18>
- [18] Christian Bienia. 2011. *Benchmarking Modern Multiprocessors*. Ph.D. Dissertation.
- [19] William Lloyd Bircher and Lizy K John. 2011. Complete system power estimation using processor performance events. *IEEE Trans. Comput.* 61, 4 (2011), 563–577.
- [20] Ata E Husain Bohra and Vipin Chaudhary. 2010. VMeter Power modelling for virtualized clouds.pdf. (2010), 1–8.
- [21] Pat Bohrer, Elmootazbellah N Elnozahy, Tom Keller, Michael Kistler, Charles Lefurgy, Chandler McDowell, and Ram Rajamony. 2002. The Case for Power Management in Web Servers. *Power Aware Computing* (2002), 261–289.
- [22] Rajkumar Buyya, Anton Beloglazov, and Jemal Abawajy. 2010. Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges. *2010 International Conference on Parallel and Distributed Processing Techniques and Applications Vm* (2010), 1–12. <https://doi.org/10.1002/cpe.1867> arXiv:1006.0308
- [23] Rajkumar Buyya and Amir Vahid Dastjerdi. 2016. *Internet of Things: Principles and Paradigms* (1st ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [24] Rajkumar Buyya, Christian Vecchiola, and S Thamarai Selvi. 2013. *Mastering Cloud Computing: Foundations and Applications Programming, 1st edition*. 469 pages. <https://doi.org/10.1016/B978-0-12-411454-8.00001-2>

- [25] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya. 2011. CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. *Softw. Pract. Exper.* 41, 1 (Jan. 2011), 23–50. <https://doi.org/10.1002/spe.995>
- [26] Mauro Canuto, Raimon Bosch, Mario Macias, and Jordi Guitart. 2016. A methodology for full-system power modeling in heterogeneous data centers. In *Proceedings of the 9th International Conference on Utility and Cloud Computing*. ACM, 20–29.
- [27] Howard Cheung, Shengwei Wang, Chaoqun Zhuang, and Jiefan Gu. 2018. A simplified power consumption model of information technology (IT) equipment in data centers for energy system real-time dynamic simulation. *Applied Energy* 222 (2018), 329 – 342. <https://doi.org/10.1016/j.apenergy.2018.03.138>
- [28] Mohammed Rashid Chowdhury, Mohammad Raihan Mahmud, and Rashedur M. Rahman. 2015. Implementation and performance analysis of various VM placement strategies in CloudSim. *Journal of Cloud Computing* 4, 1 (2015), 1–21. <https://doi.org/10.1186/s13677-015-0045-5>
- [29] Standard Performance Evaluation Corporation. 2019. *Dell Inc. PowerEdge R7425 (AMD EPYC 7601 2.20 GHz)*. Retrieved April 25, 2019 from [https://www.spec.org/power\\_ssj2008/results/res2019q1/power\\_ssj2008-20190212-00876.html](https://www.spec.org/power_ssj2008/results/res2019q1/power_ssj2008-20190212-00876.html)
- [30] Standard Performance Evaluation Corporation. 2019. *Lenovo Global Technology ThinkSystem SR150*. Retrieved April 25, 2019 from [https://www.spec.org/power\\_ssj2008/results/res2019q1/power\\_ssj2008-20181225-00874.html](https://www.spec.org/power_ssj2008/results/res2019q1/power_ssj2008-20181225-00874.html)
- [31] Standard Performance Evaluation Corporation. 2019. *SPECpower*. Retrieved April 25, 2019 from [https://www.spec.org/power\\_ssj2008/results/](https://www.spec.org/power_ssj2008/results/)
- [32] Natural Resources Defense Council. 2015. *America's Data Centers Consuming and Wasting Growing Amounts of Energy*. Retrieved April 25, 2019 from <https://www.nrdc.org/resources/americas-data-centers-consuming-and-wasting-growing-amounts-energy>
- [33] Leandro Fontoura Cupertino, Georges Da Costa, and Jean-Marc Pierson. 2015. Towards a generic power estimator. *Computer Science-Research and Development* 30, 2 (2015), 145–153.
- [34] Georges Da Costa and Helmut Hlavacs. 2010. Methodology of measurement for energy consumption of applications. *Proceedings - IEEE/ACM International Workshop on Grid Computing* (2010), 290–297. <https://doi.org/10.1109/GRID.2010.5697987>
- [35] Xiangming Dai, Jason Min Wang, and Brahim Bensaou. 2016. Energy-Efficient Virtual Machines Scheduling in Multi-Tenant Data Centers. *IEEE Transactions on Cloud Computing* 4, 2 (2016), 210–221. <https://doi.org/10.1109/TCC.2015.2481401>
- [36] John D. Davis, Suzanne Rivoire, Moises Goldszmidt, and Ehsan K. Ardestani. 2012. CHAOS: Composable Highly Accurate OS-based power models. *Proceedings - 2012 IEEE International Symposium on Workload Characterization, IISWC 2012* (2012), 153–163. <https://doi.org/10.1109/IISWC.2012.6402920>
- [37] M Dayarathna, Y Wen, and R Fan. 2016. Data Center Energy Consumption Modeling: A Survey. *IEEE Communications Surveys & Tutorials* 18, 1 (2016), 732–794. <https://doi.org/10.1109/COMST.2015.2481183>
- [38] Gaurav Dhiman, Kresimir Mihic, and Tajana Rosing. 2010. A system for online power prediction in virtualized environments using gaussian mixture models. 3 (2010), 807–812. <http://files.125/dhiman2010.pdf>
- [39] Dimitris Economou, Suzanne Rivoire, Christos Kozyrakis, and Partha Ranganathan. 2006. Full-System Power Analysis and Modeling for Server Environments. *Workshop on Modeling, Benchmarking and Simulation (MoBS)* 3 (2006), 807–812.
- [40] Elmootazbellah N Elnozahy, Michael Kistler, and Ramakrishnan Rajamony. 2003. Energy-efficient server clusters. *Power Aware Computing Systems: Second International Workshop* (2003), 179–197. <https://doi.org/10.1017/CBO9781107415324.004> arXiv:arXiv:1011.1669v3
- [41] Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. 2007. Power provisioning for a warehouse-sized computer. *ACM SIGARCH Computer Architecture News* 35, 2 (2007), 13. <https://doi.org/10.1145/1273440.1250665> arXiv:arXiv:1006.1401v2
- [42] Fahimeh Farahnakian, Pasi Liljeberg, and Juha Plosila. 2014. Energy-Efficient Virtual Machines Consolidation in Cloud Data Centers Using Reinforcement Learning. *2014 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing* (2014), 500–507. <https://doi.org/10.1109/PDP.2014.109>
- [43] Center for Machine Learning and Intelligent Systems. 2019. *UCI Machine Learning Repository*. <https://archive.ics.uci.edu/ml/index.php> (Accessed on 10/16/2018).
- [44] The R foundation. 2019. *R: The R Project for Statistical Computing*. Retrieved April 25, 2019 from <https://www.r-project.org/>
- [45] Carlucci Gaetano. 2018. *CPUloadGenerator*. Retrieved April 25, 2019 from <https://github.com/GaetanoCarlucci/CPUloadGenerator>
- [46] Inc. Gentoo Foundation. 2018. *Sysbench*. Retrieved April 25, 2019 from <https://wiki.gentoo.org/wiki/Sysbench>
- [47] Daniel Gmach, Jerry Rolia, Ludmila Cherkasova, and Alfons Kemper. 2009. Resource pool management: Reactive versus proactive or let's be friends. *Computer Networks* 53, 17 (2009), 2905–2922. <https://doi.org/10.1016/j.comnet.2009.08.011>

- [48] Chen Gong, He Wenbo, Liu Jie, Nath Suman, Rigas Leonidas, Xiao Lin, and Zhao Feng. 2008. Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services. *USENIX Symposium on Networked Systems Design and Implementation (NSDI)* (2008), 337–350. <https://doi.org/10.1109/INFCOM.2012.6195719>
- [49] Albert Greenberg, James Hamilton, David A Maltz, and Parveen Patel. 2008. The cost of a cloud: research problems in data center networks. *ACM SIGCOMM computer communication review* 39, 1 (2008), 68–73.
- [50] Steve Greenberg, Evan Mills, Bill Tschudi, and Lawrence Berkeley. 2006. Best Practices for Data Centers : Lessons Learned from Benchmarking 22 Data Centers T. *Aceee SUMMER, Lbnl* (2006), 76–87. <https://doi.org/10.1016/j.energy.2012.04.037>
- [51] Brendan Gregg. 2008. *Linux perf Examples*. Retrieved April 25, 2019 from <http://www.brendangregg.com/perf.html>
- [52] The Green Grid. 2011. *The ROI of Cooling System Energy Efficiency Upgrades - Case Study*. Technical Report. 1–42 pages.
- [53] Marco Guazzzone, Cosimo Anglano, and Massimo Canonico. 2012. Exploiting VM Migration for the Automated Power and Performance Management of Green Cloud Computing Systems. In *E2DC 2012: Energy Efficient Data Centers*. 81–92.
- [54] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter* 11, 1 (2009), 10–18.
- [55] Sang Woo Ham, Min Hwi Kim, Byung Nam Choi, and Jae Weon Jeong. 2015. Simplified server model to simulate data center cooling energy consumption. *Energy and Buildings* 86 (2015), 328–339. <https://doi.org/10.1016/j.enbuild.2014.10.058>
- [56] Guangjie Han, Wenhui Que, Gangyong Jia, and Lei Shu. 2016. An efficient virtual machine consolidation scheme for multimedia cloud computing. *Sensors (Switzerland)* 16, 2 (2016), 1–18. <https://doi.org/10.3390/s16020246>
- [57] Taliver Heath, Ana Paula Centeno, Pradeep George, Luiz Ramos, Yogesh Jaluria, and Ricardo Bianchini. 2006. Mercury and freon: temperature emulation and management for server systems. *Proceedings of the 12th international conference on Architectural support for programming languages and operating systems* (2006), 106–116. <https://doi.org/10.1145/1168857.1168872>
- [58] Li Hongyou, Wang Jiangyong, Peng Jian, Wang Junfeng, and Liu Tang. 2013. Energy-aware scheduling scheme using workload-aware consolidation technique in cloud data centres. *China Communications* 10, 12 (2013), 114–124. <https://doi.org/10.1109/CC.2013.6723884>
- [59] T. Horvath and K. Skadron. 2008. Multi-mode energy management for multi-tier server clusters. In *2008 International Conference on Parallel Architectures and Compilation Techniques (PACT)*. 270–279.
- [60] Intel IT Center. 2012. *Big Data Analytics*. Technical Report. 27 pages. <https://doi.org/10.1007/978-3-319-10665-6>
- [61] Stefan Janacek, Kiril Schröder, Gunnar Schomaker, Wolfgang Nebel, Marco Rüschén, and Günter Pistor. 2012. Modeling and approaching a cost transparent, specific data center power consumption. *2012 International Conference on Energy Aware Computing, ICEAC 2012* (2012). <https://doi.org/10.1109/ICEAC.2012.6471012>
- [62] Mateusz Jarus, Ariel Oleksiak, Tomasz Piontek, and J Węglarz. 2013. Runtime power usage estimation of HPC servers for various classes of real-life applications. *Future Generation Computer Systems* 36 (2013), 299–310.
- [63] Yichao Jin, Yonggang Wen, Qinghua Chen, and Zuqing Zhu. 2013. An Empirical Investigation of the Impact of Server Virtualization on Energy Efficiency for Green Data Center. *Comput. J.* 56, 8 (2013), 977–990. <https://doi.org/10.1093/comjnl/bxt017>
- [64] Aman Kansal, Feng Zhao, Jie Liu, Nupur Kothari, and Arka A. Bhattacharya. 2010. Virtual machine power metering and provisioning. *Proceedings of the 1st ACM symposium on Cloud computing - SoCC '10* (2010), 39. <https://doi.org/10.1145/1807128.1807136>
- [65] He Kejing, Li Zhibo, Deng Dongyan, and Chen Yanhua. 2017. Energy-Efficient Framework for Virtual Machine Consolidation in Cloud Data Centers. 3536, c (2017), 1–13. <https://doi.org/10.1109/ACCESS.2017.2711043>
- [66] Daniel C. Kilper, Gary Atkinson, Steven K. Korotky, Suresh Goyal, Peter Vetter, Dusan Suvakovic, and Oliver Blume. 2011. Power trends in communication networks. *IEEE Journal on Selected Topics in Quantum Electronics* 17, 2 (2011), 275–284. <https://doi.org/10.1109/JSTQE.2010.2074187>
- [67] Ricardo Koller, Akshat Verma, and Anidya Neogi. 2010. WattApp : An Application Aware Power Meter for Shared Data Centers. *International Conference on Autonomic Computing* (2010), 10. <https://doi.org/10.1145/1809049.1809055>
- [68] Alexey Kopytov. 2006. SysBench manual. *Test* (2006). <http://imysql.com/wp-content/uploads/2014/10/sysbench-manual.pdf>
- [69] N Kord and H Haghighi. 2013. An energy-efficient approach for virtual machine placement in cloud based data centers. *Information and Knowledge Technology (IKT), 2013 5th Conference on* (2013), 44–49. <https://doi.org/10.1109/IKT.2013.6620036>
- [70] Rainer Kress. 1998. *Numerical analysis*. Springer, New York, NY, USA.
- [71] Etienne Le Sueur and Gernot Heiser. 2010. Dynamic voltage and frequency scaling: The laws of diminishing returns. In *Proceedings of the 2010 international conference on Power aware computing and systems*. 1–8.

- [72] Young Choon Lee and Albert Y. Zomaya. 2012. Energy efficient utilization of resources in cloud computing systems. *The Journal of Supercomputing* 60, 2 (2012), 268–280. <https://doi.org/10.1007/s11227-010-0421-3>
- [73] Tao Li and Lizy Kurian John. 2003. Run-time modeling and estimation of operating system power consumption. *ACM SIGMETRICS Performance Evaluation Review* 31 (2003), 160. <https://doi.org/10.1145/885651.781048>
- [74] Yuanlong Li, Han Hu, Yonggang Wen, and Jun Zhang. 2016. Learning-based power prediction for data centre operations via deep neural networks. *Proceedings of the 5th International Workshop on Energy Efficient Data Centres - E2DC '16* (2016), 1–10. <https://doi.org/10.1145/2940679.2940685>
- [75] Yanfei Li, Ying Wang, Bo Yin, and Lu Guan. 2012. An online power metering model for cloud environment. *Proceedings - IEEE 11th International Symposium on Network Computing and Applications, NCA 2012* (2012), 175–180. <https://doi.org/10.1109/NCA.2012.10>
- [76] Chia Hung Lien, Ying Wen Bai, and Ming Bo Lin. 2007. Estimation by software for the power consumption of streaming-media servers. *IEEE Transactions on Instrumentation and Measurement* 56, 5 (2007), 1859–1870. <https://doi.org/10.1109/TIM.2007.904554>
- [77] Weiwei Lin, Wentai Wu, Haoyu Wang, James Z. Wang, and Ching-Hsien Hsu. 2018. Experimental and quantitative analysis of server power model for cloud data centers. *Future Generation Comp. Syst.* 86 (2018), 940–950. <https://doi.org/10.1016/j.future.2016.11.034>
- [78] Haikun Liu, Cheng-Zhong Xu, Hai Jin, Jiayu Gong, and Xiaofei Liao. 2011. Performance and energy modeling for live migration of virtual machines. *Proceedings of the 20th international symposium on High performance distributed computing - HPDC '11* May 2014 (2011), 171. <https://doi.org/10.1145/1996130.1996154>
- [79] Liang Luo, Wenjun Wu, W.T. Tsai, Dichen Di, and Fei Zhang. 2013. Simulation of power consumption of cloud data centers. *Simulation Modelling Practice and Theory* 39 (2013), 152 – 171. <https://doi.org/10.1016/j.simpat.2013.08.004>
- [80] Theodosios Makris. 2017. *Measuring and Analyzing Energy Consumption of the Data Center*. Ph.D. Dissertation.
- [81] Vimal Mathew, Ramesh K. Sitaraman, and Prashant Shenoy. 2012. Energy-aware load balancing in content delivery networks. *2012 Proceedings IEEE INFOCOM* (2012), 954–962. <https://doi.org/10.1109/INFOCOM.2012.6195846>
- [82] John C McCullough, Yuvraj Agarwal, Jaideep Chandrashekar, Sathyanarayan Kuppaswamy, Alex C Snoeren, and Rajesh K Gupta. 2011. Evaluating the effectiveness of model-based power characterization. In *USENIX Annual Technical Conf.* Vol. 20.
- [83] David Meisner and Thomas F Wenisch. 2010. Peak power modeling for data center servers with switched-mode power supplies. *Proceedings of the 16th ACM/IEEE international symposium on Low power electronics and design* (2010), 319–324. <https://doi.org/10.1145/1840845.1840911>
- [84] Peter Mell and Timothy Grance. 2011. The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. *Nist Special Publication* 145 (2011), 7. <https://doi.org/10.1136/emj.2010.096966> arXiv:2305-0543
- [85] Bryan Mills, Taieb Znati, Rami Melhem, Kurt B. Ferreira, and Ryan E. Grant. 2014. Energy consumption of resilience mechanisms in large scale systems. *Proceedings - 2014 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2014* (2014), 528–535. <https://doi.org/10.1109/PDP.2014.111>
- [86] Christoph Möbius, Waltenegus Dargie, and Alexander Schill. 2013. Power consumption estimation models for processors, virtual machines, and servers. *IEEE Transactions on Parallel and Distributed Systems* 25, 6 (2013), 1600–1614.
- [87] Mathijs Mortimer. 2018. iperf3 Documentation. (2018).
- [88] Hitoshi Nagasaka and Naoya Maruyama. 2010. PPT: Statistical Power Modeling of GPU Kernels Using Performance Counters GPGPU in Scientific Computing. *Computing* (2010).
- [89] Riddhi Patel, Hitul Patel, and Sanjay Patel. 2015. Quality of Service Based Efficient Resource. *International Journal For Technological Research In Engineering* 2, 9 (2015), 2008–2013.
- [90] Massoud Pedram and Inkwon Hwang. 2010. Power and performance modeling in a virtualized server system. *Proceedings of the International Conference on Parallel Processing Workshops* (2010), 520–526. <https://doi.org/10.1109/ICPPW.2010.76>
- [91] Steven Pelley, David Meisner, Thomas F Wenisch, and James W VanGilder. 2009. Understanding and abstracting total data center power. In *Workshop on Energy-Efficient Design*, Vol. 11.
- [92] Asfandiyar Qureshi, Rick Weber, Hari Balakrishnan, John Guttag, and Bruce Maggs. 2009. Cutting the electric bill for internet-scale systems. *ACM SIGCOMM Computer Communication Review* 39, 4 (2009), 123. <https://doi.org/10.1145/1594977.1592584>
- [93] Ramya Raghavendra, Parthasarathy Ranganathan, Vanish Talwar, Zhikui Wang, and Xiaoyun Zhu. 2008. No ÅIJ Power ÅÄI Struggles : Coordinated Multi-level Power Management for the Data Center. , 48–59 pages. <https://doi.org/10.1145/1346281.1346289>

- [94] AA Rahmanian, GH Dastghaibiyar, and H Tahayori. 2017. Penalty-aware and cost-efficient resource management in cloud data centers. *International Journal of Communication Systems* 30, 8 (2017), e3179. <https://doi.org/10.1002/dac.3179>
- [95] Patrick Raycroft, Ryan Jansen, Mateusz Jarus, and Paul R. Brenner. 2014. Performance bounded energy efficient virtual machine allocation in the global cloud. *Sustainable Computing: Informatics and Systems* 4, 1 (2014), 1–9. <https://doi.org/10.1016/j.suscom.2013.07.001>
- [96] Douglas Reynolds. 2015. Gaussian mixture models. *Encyclopedia of biometrics* (2015), 827–832.
- [97] Suzanne Rivoire, Parthasarathy Ranganathan, and Christos Kozyrakis. 2008. A comparison of high-level full-system power models. *Conference on Power aware computing and systems (HotPower 2008)* (2008), 1–5.
- [98] Suzanne Marion Rivoire. 2008. *MODELS AND METRICS FOR ENERGY-EFFICIENT COMPUTER SYSTEMS*. Ph.D. Dissertation. Stanford University.
- [99] Osman Sarood, Akhil Langer, Abhishek Gupta, and Laxmikant Kale. 2014. Maximizing Throughput of Overprovisioned HPC Data Centers under a Strict Power Budget. *International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2015-Janua*, January (2014), 807–818. <https://doi.org/10.1109/SC.2014.71>
- [100] Bernhard Scholkopf and Alexander J Smola. 2001. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.
- [101] Neeraj Sharma and Ram Mohana Guddeti. 2016. Multi-Objective Energy Efficient Virtual Machines Allocation at the Cloud Data Center. *IEEE Transactions on Services Computing* 1374, c (2016), 1–1. <https://doi.org/10.1109/TSC.2016.2596289>
- [102] Donghwa Shin, Jihun Kim, Naehyuck Chang, Jinhang Choi, Sung Woo Chung, and Eui-Young Chung. 2009. Energy-optimal dynamic thermal management for green computing. *Proceedings of the 2009 International Conference on Computer-Aided Design - ICCAD '09* (2009), 652. <https://doi.org/10.1145/1687399.1687520>
- [103] Richa Sinha, Nidhi Purohit, and Hiteshi Diwanji. 2011. Power aware live migration for data centers in Cloud using dynamic threshold. *International Journal of Computer Technology and Applications* 2, 6 (2011), 2041–2046. <https://doi.org/10.1.1.658.4169>
- [104] James William Smith, Ali Khajeh-Hosseini, Jonathan Stuart Ward, and Ian Sommerville. 2012. CloudMonitor : Profiling Power Usage. In *CLOUD '12 Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing*. 3–4.
- [105] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 151–161.
- [106] Stanford. 2007. *Regularization : Ridge Regression and the LASSO The Bias-Variance Tradeoff*. Technical Report. <http://statweb.stanford.edu/~owen/courses/305/Rudyregularization.pdf>
- [107] Gang Sun, Vishal Anand, Dan Liao, Chuan Lu, Xiaoning Zhang, and Ning-Hai Bao. 2015. Power-Efficient Provisioning for Online Virtual Network Requests in Cloud-Based Data Centers. *IEEE Systems Journal* 9 (2015), 427–441.
- [108] Cheng Jen Tang and Miao Ru Dai. 2011. Dynamic computing resource adjustment for enhancing energy efficiency of cloud service data centers. *2011 IEEE/SICE International Symposium on System Integration, SII 2011* (2011), 1159–1164. <https://doi.org/10.1109/SII.2011.6147613>
- [109] M. Tang and S. Pan. 2015. A Hybrid Genetic Algorithm for the Energy-Efficient Virtual Machine Placement Problem in Data Centers. *Neural Processing Letters* 41, 2 (2015), 211–221. <https://doi.org/10.1007/s11063-014-9339-8>
- [110] Tektronix. 2003. *Digital Storage Oscilloscope*. Technical Report 7. 2004 pages. <https://doi.org/10.1002/ejoc.201200111> arXiv:arXiv:1011.1669v3
- [111] Princeton University. 2018. *The PARSEC Benchmark Suite*. Retrieved April 25, 2019 from <http://parsec.cs.princeton.edu/index.htm>
- [112] Henk Vandenbergh. 2012. Vdbench Users Guide. October (2012), 1–114.
- [113] Micha vor dem Berge, Georges Da Costa, Andreas Kopecki, Ariel Oleksiak, Jean-Marc Pierson, Tomasz Piontek, Eugen Volk, and Stefan Wesner. 2012. Modeling and simulation of data center energy-efficiency in coolemall. In *International Workshop on Energy Efficient Data Centers*. Springer, 25–36.
- [114] Di Wang, Chuangan Ren, Sriram Govindan, Anand Sivasubramaniam, Bhuvan Ugaonkar, Aman Kansal, and Kushagra Vaid. 2013. ACE: abstracting, characterizing and exploiting peaks and valleys in datacenter power consumption. In *ACM SIGMETRICS Performance Evaluation Review*, Vol. 41. ACM, 333–334.
- [115] Zhikui Wang, Niraj Tolia, and Cullen Bash. 2010. Opportunities and challenges to unify workload, power, and cooling management in data centers. *ACM SIGOPS Operating Systems Review* 44, 3 (2010), 41. <https://doi.org/10.1145/1842733.1842741>
- [116] B.D. Wedlock and J.K. Roberge. 1969. *Electronic components and measurements*. Prentice-Hall.
- [117] Yingyou Wen, Zhi Li, Shuyuan Jin, Chuan Lin, and Zheng Liu. 2017. Energy-Efficient Virtual Resource Dynamic Integration Method in Cloud Computing. *IEEE Access* 5 (2017), 12214–12223. <https://doi.org/10.1109/ACCESS.2017.2721548>



- [118] BUSINESS WIRE. 2018. *Global Data Center Services Market Growth, Trends, and Forecasts 2018-2023: Tier 4 Data Center Type to Have the Highest Share*. Retrieved April 25, 2019 from <https://www.businesswire.com/news/home/20180517005800/en/Global-Data-Center-Services-Market-Growth-Trends>
- [119] Michal Witkowski, Ariel Oleksiak, Tomasz Piontek, and J Węglarz. 2012. Practical power consumption estimation for real life HPC applications. *Future Generation Computer Systems* 29, 1 (2012), 208–217.
- [120] Wei Wu, Lingling Jin, Jun Yang, Pu Liu, and Sheldon X.-D. Tan. 2007. Efficient power modeling and software thermal sensing for runtime temperature monitoring. *ACM Transactions on Design Automation of Electronic Systems* 12, 3 (2007), 26–es. <https://doi.org/10.1145/1255456.1255462>
- [121] Hong Xu and Baochun Li. 2013. Reducing Electricity Demand Charge for Data Centers with Partial Execution. (2013), 51–61. <https://doi.org/10.1145/2602044.2602048> arXiv:1307.5442
- [122] X. Ye, Y. Yin, and L. Lan. 2017. Energy-Efficient Many-Objective Virtual Machine Placement Optimization in a Cloud Computing Environment. *IEEE Access* 5 (2017), 16006–16020. <https://doi.org/10.1109/ACCESS.2017.2733723>
- [123] Xiao Zhang, Jian Jun Lu, Xiao Qin, and Xiao Nan Zhao. 2013. A high-level energy consumption model for heterogeneous data centers. *Simulation Modelling Practice and Theory* 39 (2013), 41–55. <https://doi.org/10.1016/j.simpat.2013.05.006>
- [124] Kuangyu Zheng, Xiaodong Wang, Li Li, and Xiaorui Wang. 2014. Joint power optimization of data center network and servers with correlation analysis. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2598–2606.