PCP-2LSTM: Two Stacked LSTM-based Prediction Model for Power Consumption in Data Centers

Ziyu Shen, Xusheng Zhang, Binghui Liu, Bin Xia, Zheng Liu, Yun Li
Nanjing Key Laboratory of Big Data Security & Intelligent Processing
Nanjing University of Posts and Telecommunications
Nanjing, China
1017041118, 1018041125, 1018041115, bxia, zliu, liyun@njupt.edu.cn

Saiqin Long
College of Information Engineering
Xiangtan University
Xiangtan, China
282293237@qq.com

***Abstract*—As the size of data centers and cloud computing continue to expand, power consumption in data centers is rapidly increasing. It has a great significance to predict and analyze power consumption in the data center because power consumption prediction can help data center operators perform workflow scheduling, manage energy efficiency, provide high quality-of-service (QoS), and meet the requirements of green energy use. The current methods are mainly divided into two scopes: the one is establishing a static relationship between power consumption and relevant components/applications, and the other one is treating power consumption as sequential temporal data. However, the first scope does not consider the dynamic fluctuation of power, and the other one ignores the characteristics of the power consumption data. To solve these issues, in this paper, we present a power consumption prediction framework called PCP-2LSTM based on the mean smoothing and long short-term memory (LSTM) network. We first build a power consumption system to collect data and analyze the stationary of the power series. Then we use the mean smoothing to remove the noise from the time series of power consumption. After data preprocessing, because the time for workflow and container scheduling is usually 30 seconds, we use a stacked LSTM model to predict 30s power consumption in the future. The experimental result indicates that our approach outperforms other baselines.**

***Index Terms*—Data center, time series prediction, energy efficiency, power consumption**

I. Introduction

Although the demand for computing ability in cloud computing has promoted the rapid development of data centers, it caused excessive power consumption in data centers. Moreover, applications and service requirements also cause large power consumption in data centers. The huge power consumption has brought high costs to data center operators and does harm to the environment. In data centers, the servers in the running state consume 10% - 50% [1] of the peak power on average while the idle servers can consume up to 60% of the peak power [2], which causes large waste of power. Therefore, it is imminent to improve the energy efficiency of data centers. Nowadays, three technologies, which are dynamic voltage and frequency scaling (DVFS) [3], dynamic power management (DPM) [4], and power napping [5], are commonly used in data centers for improving energy efficiency. However, these technologies are not enough to save large amounts of power consumption and unable to meet resource management requirements in data centers.

To overcome these issues, the power prediction combined with workloads scheduling helps data center operators optimize the utilization and reduce the power consumption [6]. Based on the predicted power values, workloads are allocated to keep load balancing. Therefore, it is an essential problem to estimate and predict the power consumption for data center power management. However, current power prediction methods are mainly divided into two aspects. The first method is modeling the relationship between the power consumption and relevant components/applications and management systems of data centers, while this method has three disadvantages: 1) it needs professional domain knowledge and precise measurement tools; 2) the power consumption of some components cannot be measured easily; 3) besides the hardware/software configuration, power consumption is affected by other external factors and has the dynamic pattern that is hard to be captured. The second method treats power consumption as sequential temporal data, which is the lack of consideration of the power data characteristics. A good predicted model must capture the fluctuation and consider the characteristics of the power consumption.

In this work, we propose a method to predict the power consumption of data centers, which includes data acquisition, data preprocessing, time series analysis, and error analysis. In data acquisition, we collect the power series and related features by a power consumption system and analyze the stationarity of the time series. In data preprocessing, we leverage the mean smoothing to remove the white noise from the data. With the denoised data, we train the power prediction model for predicting the power in the next 30 seconds to meet the time requirement the workflow scheduling [7] and container scheduling [8].

Our main contributions are as follows:

- We build a power consumption system and run the CPU-intensive tasks to acquire the dataset.
- We collect the power dataset and analyze the non-stationary characteristic of power time series.
- We propose a power consumption prediction (PCP-2LSTM) model to predict power consumption and compare with other approaches proposed existed.

978-1-7281-5141-0/19/$31.00 ©2019 IEEE
DOI 10.1109/CBD.2019.00013

II. Related work

For solving the problem of improving the efficiency of power consumption of data centers, it is necessary to model the power prediction. The existing methods see this problem in two different views: static power addition model and time series data prediction.

a) Static power addition models: Previous studies are based on simple physical measurement to model the relationship between power consumption and the components of data centers. Joseph and Martonosi [9] and Isci and Martonosi [10] proposed detailed analytical processor power models based on CPU performance counters. Such models have been developed for processors, single systems, and groups of systems in enterprise environments [11]. However, as cloud data centers continue to expand in size and the power consumption becomes more and more complex, this approach is unable to completely model dynamic fluctuation of power consumption and ensure the prediction of power consumption effectively. Then the non-intrusive methods are proposed to model the relationship between the power consumption and the states of the servers [12] [13] [14].

b) Time series prediction models: Time series data prediction has an important significance to discuss that it has been applied widely in many domains. Mazumdar [15] used ARIMA and exponential smoothing models to predict the stability of the data center based on real-time data of batch workload. Recently, the studies of deep learning are applied in time series. G.Peter Zhang [16] proposed a hybrid ARIMA and neural network model that combines both to take advantages for time series predicting. Li [17] proposed two deep learning-based models: a fine-grained model and a coarse-grained model for power consumption prediction in data centers. In [18], the author proposed a self-aware workload prediction in data center power consumption based on a neural network model. Salam [19] proposed a multivariate time series ELM algorithm to predict cloud data center workload based on energy consumption. Recurrent neural network based on ANN has been improved in its hidden layers, which not only receives previous layer output but also get the signal of the previously hidden layer. However, RNN has problems with gradient disappearance and explosion in dealing with long-term time series. LSTM is the improvement of RNN to add a forgotten layer for solving these problems. Cheng [20] proposed powerLSTM for power demand predicting compared to gradient boosting tree (GBT) and SVR, which perform better than both. LSTM is also used for predicting building energy load with historical consumption data [21]. However, there has been not much work using LSTM based on features for power consumption prediction in data center until now, which motivates this work we do. And in this paper, we only consider the fine-grained power consumption prediction.

III. Description of system power prediction

In this section, we present the overall architecture and workflow of the power consumption system and introduce the process of time series prediction.

A. Power consumption system

The architecture and workflow of the power consumption system are shown in Fig. 1. The system contains a server, a monitor computer, a network router, and a power contribution unit (PDU).The physical machines are built on Ubuntu, collecting data from the server using Collectd [22] in the monitor computer. The server uses Apache to handle the CPU-intensive workloads to generate power consumption. We keep the servers in different running states by adjusting CPU utilization. The PDU monitors the power consumption of the server. By this system, we record the current power and various system status (detailed in Section 4), and then use these data as input data to the prediction model. The prediction framework learns from the historical data to predict future power in a certain granularity. The prediction results help workloads scheduling to optimize power performance.

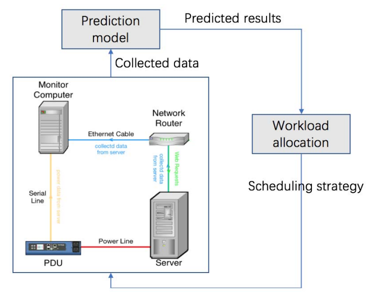

Fig. 1. The architecture and workflow of the power consumption system

B. Process of time series prediction

The prediction model is the key in the system operation. In general, time series prediction requires the algorithm to complete the prediction in time. Moreover, the transmission and acquisition of time series data are generally built on various types of sensors, causing the time series to contain the noise, which affects the accuracy of the prediction model.

To solve these issues, we propose a complete framework, including four steps: data acquisition, data preprocessing, prediction modeling, and error analysis for time series prediction, where the process is shown in Fig. 2.

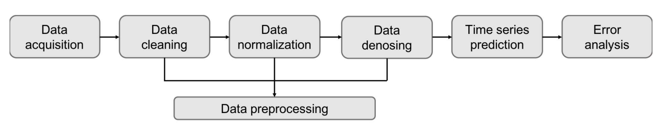

Fig. 2. Process of time series prediction

In data acquisition, we collect power consumption data, system status, and workload data (Section 4).

Data preprocessing consists of three parts: data cleaning, data normalization, and data denoising. We define the observation window size and the prediction window size in data cleaning. The observation window size is the time granularity of a sample, which is the set of the values of the features in a certain time range. The prediction window size is the predicted granularity. Data normalization is to scale the data down to a small specific interval. This paper uses normalization to scale the data between -1 and 1. In this work, we normalize the values of features except for the values of power consumption. The transforming formula is as follows:

$$x' = \frac{x - \mu}{\sigma}, \tag{1}$$

where x represents the original values of the features, μ represents the average value of the corresponding domain of features, and σ represents the standard variance of the corresponding domain of features.

The time series of power consumption contains random noise, which has a high interference with the accuracy of prediction. We use the mean smoothing to remove the noise from data.

After then, we use the denoised data to predict power consumption. Time series prediction has two types: the fine-grained prediction and the coarse-grained prediction. In this work, for meeting the time requirement of workflow scheduling, container scheduling, and task management, we study the fine-grained prediction of the data center.

IV. Data collection

In this section, we present the software level of the power consumption system, data acquisition, and data analysis.

A. Software layout of the power consumption system

In this experiment, we use CPU-intensive workloads to collect the data of power consumption and system status for training model. We use Collectd as a measurement tool, as shown in Fig. 1. Collectd measures comprehensive energy-related information such as CPU utilization, memory utilization, and network traffic. Then We use MongoDB to store data information from Collectd and PDU.

Note that the power consumption system is simplified in many aspects, compared to real data centers. We only run one type of workloads while in data centers, there are many complex applications. However, the key to power consumption prediction is that the model can capture the dynamic fluctuation and the inherent characteristics of power consumption. Moreover, our model is based on the black-box model that is not dependent on specific applications. We can predict the power of other applications by changing the relevant training dataset.

B. Power consumption dataset

In this subsection, we will describe the features of the collected data. The data we collect for power consumption prediction can be divided into three categories:

1. Parameters of system state: for predicting the power consumption, some related system counters data should be collected. We collect these data per second. The number of features is 158, and some detailed features are listed in TABLE I.

2. Temporal data of workloads: by sending requests to web servers, we can approximately acquire the workload level. These data are recorded per second, which needs to keep consistent with the global timestamp.

TABLE I
Input variables decriptions

System counters	Related	Details
CPU usage	CPU	Idle, non-idle(active)
CPUfreq		the current CPU frequency
Intel CPU performance counters		hardware CPU cache events, software events and hardware events meaurement
Memory usage	Memory	Bytes of physical memory used Percent of physical memory used
Memcached		Statistics about memory used, as well as cache utilizaiton and bandwidth used
Disk usage	Disk	The usage of physical disks and logical disks(partitions)
Disk I/O information		Bytes wtitten to/read from as disk, bytes write/read operations to a disk, opetations time
Df		Information about disk space used/free
Protocols	Network	Network protocols information (IP, TCP, UDP, etc.)
Ping		The average latency, the standard deviation and the drop rate for each host
DNS status		DNS traffic capture and analysis
Network file sysetm (NFS) status		Information about NFS used
MIC		CPU statistics, memory usage and temperatures from Intel's Many Integrated Core (MIC) system
Write mongodb		Send values to mongodb

3. Power-time data: the power consumption from PDU is recorded per second.

Actually, although statistical methods assume that the time series is stationary, the power consumption of the data center varies dramatically that the assumption is not always true [23]. We analyze the stationarity of power data series. We use the autocorrelation function (ACF) and partial autocorrelation function (PACF) to test the data. The stationary data have a short-term correlation that as the lags increases, the autocorrelation coefficient decreases rapidly to 0. Conversely, if the rate of decline is low, the data series is non-stationary. We plot the power data series diagram, autocorrelation plot, and partial autocorrelation plot, as shown in Fig. 3. It can be seen from

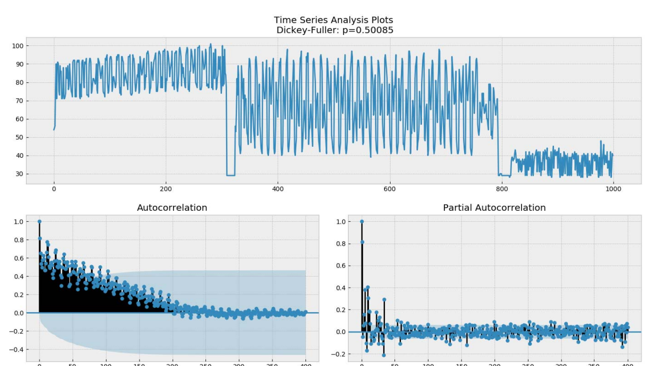

Fig. 3. ACF, PACF diagram

the timing diagram that the intervals of peaks are similar and periodic. We find that the autocorrelation coefficient decays slowly to 0. We can get a conclusion that the power data series are non-stationary.

Moreover, the power profile has a sudden drop from the peak to the bottom of the valley. The decline to the low power is set as sleep for a temporal time for simulating the servers in real data centers in idle and rarely emerging emergencies or exceptional cases which makes the time series more non-stationary and increases the difficulty in predicting accuracy.

V. Power consumption prediction model for fine-grained prediction

A. LSTM

Long short-term memory (LSTM) network [24] is a variation of recurrent neural network (RNN). In recent years, LSTM is widely applied in dealing with the problem of predicting time series, which can avoid the long-term dependency problem.

As shown in Fig. 4, a memory cell consists of three gates: an input gate, a forget gate and an output gate. These gates control the information passed whether or not by sigmoid function and a pointwise multiplication operation.

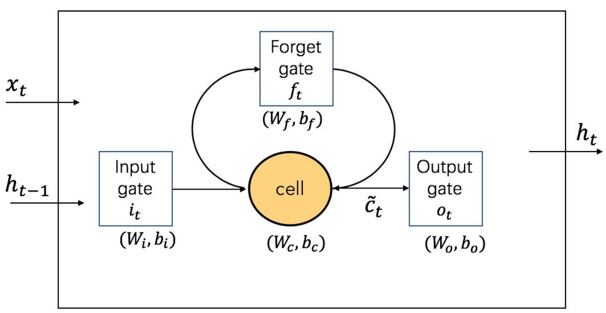

Fig. 4. LSTM cell structure

The meaning of mathematics Symbols in Fig. 5 are as follows:

1. x_t represents the input vector at time t;
2. h_t is the state value of the memory cell at time t, used as input at the next time $t+1$;
3. W_i, W_f, W_o and W_c are the corresponding weight matrics of three gates and cell respectively;
4. b_i, b_f, b_o and b_c are the corresponding bias vectors of three gates and cell respectively;
5. i_t, f_t and o_t are the values of three gates.

In the first step, LSTM decides which information to go through the cell by using the forget gate. The gate generates a value between 0 and 1 according to h_{t-1} and x_t. 1 means "complete acceptance" while 0 means "complete rejection". The formulation is as followed:

$$f_t = \sigma(W_f \cdot [x_t, h_{t-1}] + b_f), \tag{2}$$

In the second step, the input gate uses sigmoid to decide which information need to update to generate a new matric, $\widetilde{c_t}$, a candidate state of the memory cell at time t. It is formulated as below,

$$i_t = \sigma(W_i \cdot [x_t, h_{t-1}] + b_i), \tag{3}$$

$$\widetilde{c_t} = tanh(W_c \cdot [x_t, h_{t-1}] + b_c), \tag{4}$$

In the third step, we use f_t and $\widetilde{c_t}$ to update the state of the memory cell, c_t.

$$c_t = f_t * c_{t-1} + i_t * \widetilde{c_t}, \tag{5}$$

In the last step, we first use the output gate to calculate a coefficient of output, then we use $tanh$ function to scale c_t into [-1,1] as the final output h_t.

$$o_t = \sigma(W_o \cdot [x_t, h_{t-1}] + b_o), \tag{6}$$

$$h_t = o_t * tanh(c_t), \tag{7}$$

B. PCP-2LSTM model

We use two stacked LSTM to model the relationship between power consumption and relevant features to predict the power.

In Fig. 4, x_t represents the input and $\hat{y}_{t+1}$ is the output value at time t which represents the power value in the future time.

Fig. 5 shows the flow chart of the algorithm.

We predict the power consumption by modeling the relationship between power consumption and related features. First, we use the mean smoothing to remove the noise from original power data. Then we leverage the denoised data to train the power consumption prediction model. We define the size of the prediction window to predict power consumption. We predict the next 30 seconds by using the previous 30 seconds of the samples. Therefore, our observation window size is 30, and the prediction size is set to 30. In the last step, the model outputs the predicted value of power consumption and is compared with the other five baselines.

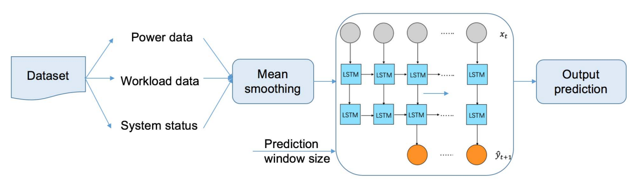

Fig. 5. Flow chart of the algorithm

VI. Experiment

A. Data preprocessing

In this section, we process the power consumption data to make it more suitable for prediction. Fig. 6 shows the power profile when the server is in the idle state for two different periods. As observed from Fig. 6 that the power consumption curve fluctuates frequently and dramatically when there is no load on the server, which indicates that there is a certain noise in the idle state of the server power consumption.

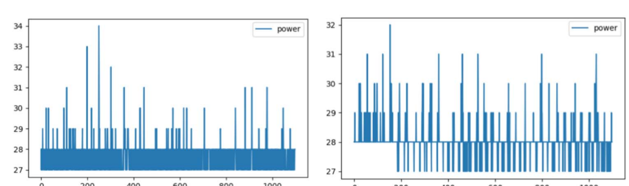

Fig. 6. Power profile when the server is in idle state

To alleviate the effect of noise, we leverage the mean smoothing to remove the white noise. We substitute the average values of the power consumption for the original data at the length of l and downsample the value in the window. When the length of the window is large enough, the impact of the noise can be eliminated. It is important that the length is not chosen at random. The key to this method is to ensure a suitable size of the window to smooth. The window size l decides to eliminate how much noise and discard how much information.

We experiment with the size of the different windows and choose the appropriate window size by comparing their variance. We use the variance ratio as the choice of window size, as shown in the following formula.

$$variance\ ratio = \frac{std(P_i(t))}{std(P_{idle}(t))}, \tag{8}$$

where $std(P_i(t))$represents the standard variance in running power data after smoothing of different window sizes and $std(P_{idle}(t))$ represents the standard variance in data when the server is idle.

For the data series, as the smoothed window size increases, the variance ratio of the data decreases, as shown in Fig. 7. After the window size is greater than 30, the variance ratio polyline tends to be horizontal. The variance ratio is smaller; the effect of denoising is better. However, a larger window size will cause the time series to lose too much information. We collect the data per second, which means the original time unit of power consumption is 1 second. The new power series now has $1 \cdot l$ time unit. Therefore, we set the observation window size to 30 seconds and deal with all the data with this smoothing method.

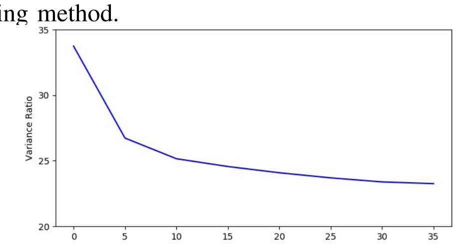

Fig. 7. The relationship between window size and variance ratio

B. Settings and results

We normalize the data of features selected except values of power to unify the values of individual features for eliminating the effect of different dimensions. The observation window size and prediction window size are set to 30. We use the PCP-2LSTM model, where the number of hidden layers is 3, and the number of nodes is 32. We conduct the experiments compared our model with the stacked LSTM without the mean smoothing (2LSTM), GBR, recursive autoencoder (RAE) [25], LSTM, and ARIMA. Except for the 2LSTM model, GBR, RAE, LSTM, and ARIMA use the mean smoothing for data preprocessing. In the dataset, the first 70 percent is the training set; the last 30 percent is the testing set. We set dropout = 0.1 and epoch = 20000. Mean squared error (MSE) measures the average value of the squared errors as the following formula (9).

$$MSE = \frac{1}{M}\sum_{t=1}^{M}(Y_t - \hat{Y}_t), \tag{9}$$

where M is the number of predicted values, Y_t represents the actual power values that the PDU measures at time t and $\hat{Y}_t$ is the predicted values at time t.

We use the normalized Root Mean Squared Error (nRMSE) to evaluate the accuracy of the prediction model. nRMSE is calculated by Eq (10).

$$nRMSE = \frac{\sqrt{MSE}}{\sigma'_Y}, \tag{10}$$

where σ'_Y represents the standard error of the testing set of power consumption.

TABLE II
PRECTION ACCUARCY COMPARISON OF METHODS

Error	PCP-2LSTM	2LSTM	LSTM	GBR	RAE	ARIMA
nRMSE	0.417	1.002	0.474	0.5	32.8	0.47

Fig. 8 shows the results of different prediction models. TABLE II presents the predicting accuracy of six models by nRMSE. We observe that our model can fit the actual power consumption and outperforms the other five models in predicting the power consumption. The experimental results show that the mean smoothing can improve prediction accuracy. The statistic model ARIMA outperforms other methods except for our proposed model because ARIMA can smooth the non-stationary time series data and build the corresponding model according to the identified characteristics. The LSTM has a larger prediction error than PCP-2LSTM since the stacked LSTM has more hidden layers and can do better representations. However, surprisingly, RAE has rarely fluctuation in our dataset, which has an enormous error and has the worst prediction performance of all. The cause of this problem may be the linear model RAE cannot capture the fast fluctuation of the power consumption in our dataset.

VII. Conclusion

In this paper, we propose a time series predicting framework for power consumption of data centers, which contains four steps: data acquisition, data preprocessing, time series prediction, and error analysis. We build a power consumption system to collect the dataset and analyze the stationarity of our dataset. In the data preprocessing stage, the mean smoothing is applied to remove the noise. For the prediction model, we use a stacked LSTM to learn characteristic from the data and test the predictive accuracy compared with the other five models. The result indicates that our model has a higher efficiency by measured nRMSE metric and tracks the fast changes of the power consumption with high randomness well. However, the system to implement the experiment on is simplified compared to the real data center, and we only simulate CPU-intensive tasks while there are many types of requests from users in the data center. In future work, we will simulate the requests from

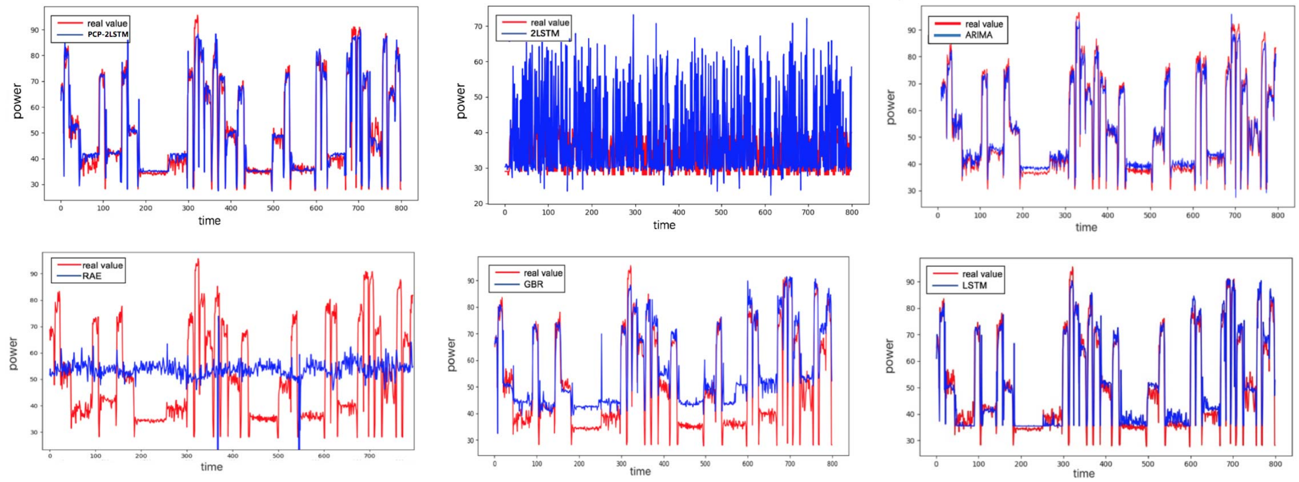

Fig. 8. Prediction results of PCP-2LSTM and other methods

the users based on a larger scale power consumption system to study the power consumption and study the influence of different tasks on power consumption. Moreover, we will study the power consumption prediction of containers.

VIII. ACKNOWLEDGEMENT

This work was supported by National Key Research and Development Program of China (2018YFB1003702).

REFERENCES

[1] D. Filani, J. He, S. Gao, M. Rajappa, A. Kumar, P. Shah, and R. Nagappan, "Dynamic data center power management: Trends, issues, and solutions." *Intel Technology Journal*, vol. 12, no. 1, 2008.

[2] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *ACM SIGCOMM computer communication review*, vol. 39, no. 1, pp. 68–73, 2008.

[3] G. Von Laszewski, L. Wang, A. J. Younge, and X. He, "Power-aware scheduling of virtual machines in dvfs-enabled clusters," in *2009 IEEE International Conference on Cluster Computing and Workshops*. IEEE, 2009, pp. 1–10.

[4] L. Benini, A. Bogliolo, and G. De Micheli, "A survey of design techniques for system-level dynamic power management," *IEEE transactions on very large scale integration (VLSI) systems*, vol. 8, no. 3, pp. 299–316, 2000.

[5] D. Meisner, B. T. Gold, and T. F. Wenisch, "Powernap: eliminating server idle power," in *ACM sigplan notices*, vol. 44, no. 3. ACM, 2009, pp. 205–216.

[6] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser, "Renewable and cooling aware workload management for sustainable data centers," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 1. ACM, 2012, pp. 175–186.

[7] M. Malawski, G. Juve, E. Deelman, and J. Nabrzyski, "Algorithms for cost- and deadline-constrained provisioning for scientific workflow ensembles in iaas clouds," *Future Generation Computer Systems*, vol. 48, no. 1, pp. 1–18, 2015.

[8] L. Yin, L. Juan, and L. Haibo, "Tasks scheduling and resource allocation in fog computing based on containers for smart manufacture," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4712–4721, 2018.

[9] C. Isci and M. Martonosi, "Runtime power monitoring in high-end processors: Methodology and empirical data," in *Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*. IEEE Computer Society, 2003, p. 93.

[10] R. Joseph and M. Martonosi, "Run-time power estimation in high performance microprocessors," in *ISLPED'01: Proceedings of the 2001 International Symposium on Low Power Electronics and Design (IEEE Cat. No. 01TH8581)*. IEEE, 2001, pp. 135–140.

[11] S. Rivoire, P. Ranganathan, and C. Kozyrakis, "A comparison of high-level full-system power models." *HotPower*, vol. 8, no. 2, pp. 32–39, 2008.

[12] G. Tang, W. Jiang, Z. Xu, F. Liu, and K. Wu, "Nipd: Non-intrusive power disaggregation in legacy datacenters," *IEEE Transactions on Computers*, vol. 66, no. 2, pp. 312–325, 2017.

[13] W. Jiang, S. Ren, F. Liu, and J. Hai, "Non-it energy accounting in virtualized datacenter," in *IEEE International Conference on Distributed Computing Systems*, 2018.

[14] W. Jiang, F. Liu, G. Tang, K. Wu, and J. Hai, "Virtual machine power accounting with shapley value," in *IEEE International Conference on Distributed Computing Systems*, 2017.

[15] S. Mazumdar and A. S. Kumar, "Forecasting data center resource usage: An experimental comparison with time-series methods," in *International Conference on Soft Computing and Pattern Recognition*. Springer, 2016, pp. 151–165.

[16] G. P. Zhang, "Time series forecasting using a hybrid arima and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[17] Y. Li, H. Hu, Y. Wen, and J. Zhang, "Learning-based power prediction for data centre operations via deep neural networks," in *Proceedings of the 5th International Workshop on Energy Efficient Data Centres*. ACM, 2016, p. 6.

[18] Y.-F. Hsu, K. Matsuda, and M. Matsuoka, "[ieee 2018 18th ieee/acm international symposium on cluster, cloud and grid computing (ccgrid) - washington, dc, usa (2018.5.1-2018.5.4)] 2018 18th ieee/acm international symposium on cluster, cloud and grid computing (ccgrid) - self-aware workload forecas," pp. 321–330, 2018.

[19] S. Ismaeel and A. Miri, "Multivariate time series elm for cloud data centre workload prediction," in *International Conference on Human-Computer Interaction*. Springer, 2016, pp. 565–576.

[20] Y. Cheng, C. Xu, D. Mashima, V. L. Thing, and Y. Wu, "Powerlstm: Power demand forecasting using long short-term memory neural network," in *International Conference on Advanced Data Mining and Applications*. Springer, 2017, pp. 727–740.

[21] D. L. Marino, K. Amarasinghe, and M. Manic, "Building energy load forecasting using deep neural networks," in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2016, pp. 7046–7051.

[22] K. Jackson, *OpenStack cloud computing cookbook*. Packt Publishing Ltd, 2012.

[23] M. Rossi and D. Brunelli, "Forecasting data centers power consumption with the holt-winters method," in *2015 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS) Proceedings*. IEEE, 2015, pp. 210–214.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.