An Artificial Neural Network Approach to Power Consumption Model Construction for Servers in Cloud Data Centers

Weiwei Lin, Guangxin Wu, Xinyang Wang, and Keqin Li, *Fellow, IEEE*

Abstract—The power consumption estimation or prediction of cloud servers is the basis of energy-aware scheduling to realize energy saving in cloud datacenters. The existing works are mainly based on the static mathematical formulas which establish the relationship between the server power consumption and the system performance. However, these models are weak in adaptability and generalization ability, not adaptable to the changes and fluctuation of different workload, and demanding on the clear and profound understanding of the inner relationship among related power consumption parameters. Therefore, we propose the ANN (Artificial Neural Network) method to model the power consumption of the servers in datacenters, a kind of end-to-end black box model. We performed a fine-grained and in-depth analysis about the system performance and power consumption characteristics of the CPU, memory, and disk of the server running different types of task loads, and selected a set of performance counters that can fully reflect the status of system power consumption as the input of the model. Then, we establish power consumption models based on BP neural network, Elman neural network, and LSTM neural network, respectively. In order to get a better result, we use data collected from four different types of task loads (i.e., CPU-intensive, memory-intensive, I/O-intensive, and mixed load) to train, validate, and test our target models. The experimental results show that, compared with multiple linear regression and support vector regression, the proposed three power models have better performance in predicting the server's real-time power consumption.

Index Terms—Power consumption, cloud datacenters, artificial neural network, power modelling

1 Introduction

With the rapid development of the cloud computing industry, as an important carrier of information, the datacenter has ushered in a wave of new construction. Studies [1] have shown that it is expected that there will be more than 500 ultra-large datacenters worldwide by 2020. At the same time, problems, such as operating costs, energy consumption and environmental protection, brought about by the rapid expansion of the datacenter, have gradually attracted people's attention. According to the statistics published in 2013, the power consumption of datacenters in the United States alone have reached 91 billion kWh; and by 2020, energy consumption is expected to increase to nearly 1400 kWh [2]. A research report [3] for the European datacenter shows that the global information and communication technology (ICT) departments (including datacenters) accounted for 2 percent of the total carbon emissions, with the fastest growth rate in datacenter.

Under the trend of energy saving, establishing a complete energy consumption monitoring mechanism in the datacenter is a prerequisite for energy planning and management. Generally, energy consumption of datacenter can be generally divided into two parts, i.e., energy consumption of IT equipment (such as servers, network equipment, and storage equipment) and energy consumption of infrastructures (such as cooling facilities, power supply facilities). For the key component in datacenters—servers, there are two primary ways to monitor its power consumption. One is the traditional hardware-based method while the other is software-based monitoring mechanism. The former generally refers to directly measuring the power consumption of servers through external power measurement devices or embedding collectors in the specific servers. This approach is feasible in small-scale datacenters, but fails to meet the low-cost, easy-to-expand monitoring requirements [4]. In contrast, the latter one can realizes multi-granular and highly scalable monitoring systems in a cost-effective manner, making them applicable for complex, heterogeneous, and frequently expanding device environments in cloud datacenters. Software-based energy consumption monitoring typically relies on pre-established energy consumption models. The energy consumption model refers to a functional model that maps system state related variables to system energy consumption or power consumption [5], and generally includes one or more function expressions with state indicators (such as CPU utilization and instruction cycle) at a certain granularity of the system as independent variable. The output of the model is the system energy consumption for a period of time or the

- *W. Lin, G. Wu, and X. Wang are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China. E-mail: linww@scut.edu.cn, cswgx1nfinite@mail.scut.edu.cn, wxyyuppie@139.com.*
- *K. Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561. E-mail: lik@newpaltz.edu.*

Manuscript received 12 Oct. 2018; revised 23 Jan. 2019; accepted 9 Mar. 2019. Date of publication 11 Apr. 2019; date of current version 8 Sept. 2020.
(Corresponding author: Weiwei Lin.)
Recommended for acceptance by Y. Wu, Y. Pan, N. Georgalas, and G. Min.
Digital Object Identifier no. 10.1109/TSUSC.2019.2910129

power consumption value at a certain moment. Our work mainly tends to predict the real-time power consumption of the server so that the proposed models are a kind of power consumption model.

At present, most of the energy consumption models used in research and engineering are based on regression analysis methods, with linear regression models as the most representative methods, whose principal advantages are the good interpretability and small training cost [6]. For example, Hsu and Poole [7] investigated several energy consumption models based on CPU utilization. Lin et al. [8] summarized and evaluated several sub-component power consumption models based on the static mathematical formulas. However, models of this type have some limitations, for example, it's difficult to establish energy consumption prediction models that are applicable to different load environments through well-defined mathematical formulas due to the complexity and variability of the running workload in servers. In addition, most of the energy consumption models only consider building the function to map the system state to corresponding power consumption, without considering the time continuity of system state change, which may impact on the system energy consumption. For instance, when the CPU temperature reaches the threshold because of high utilization, it will trigger a frequency reduction or other heat dissipation measures, and finally reflects the power consumption performance.

In the field of cloud computing, many attempts based on machine learning methods, especially for artificial neural networks (ANN). For example, in the research of server load forecasting, literatures like [9], [10], [11] adopted different types of ANNs to model and predict the load changes of servers. ANN is a computation model that simulates the working principle of human brain, and consists of many nodes (neurons) connecting each other. Zuo et al. [12] adapt a sequence-to-sequence model, a learning-based network, for dynamic path planning in traffic engineering. In order to cope with weakened trustworthiness of cloud services, Huang et al. [13] improve and propose Linear programming SVDD based on Support Vector Data Description (SVDD) to apply to detect anomalous performance metrics of cloud services. According to the topology of networks, ANN can be divided into different types, such as feed-forward network, feedback network and random network. Three primary advantages of choosing different ANNs to model can be concluded as:

1) Certain adaptability. During ANN learning and training process, the weights in the network will change accordingly with input data and training methods to adapt to different environments and to obtain different target models;
2) Strong generalization ability. For some untrained samples, especially the samples with noise, the models have better predictive ability;
3) Strong nonlinear mapping ability. In the process of establishing prediction models by mathematical methods such as linear regression, it is usually necessary for the designers to have comprehensive understanding of the objectives of modeling, and it is particularly difficult to establish an accurate prediction model when the objectives are very complicated. The ANN-based prediction model requires not a thorough understanding of the modeling objectives, and can establish an accurate mapping function between input and output in an easier way.

Based on the above research background, this paper is devoted to the research on ANN-based server power consumption modeling, different from the traditional regression method. While running with different types of workload (i.e., CPU-intensive workloads, memory-intensive workloads, I/O-intensive workloads, and mixed workloads), we monitor resource utilizations of the three main components (CPU, memory, and disk) of the server, and perform a quantitative analysis for the performance status of the server and their corresponding energy consumption characteristics. In order to explore the contribution of system performance state in a continuous time range to the server power consumption at the current prediction time, this paper proposes three corresponding server power consumption models respectively based on BP neural network (BPNN), Elman neural network (ENN) and Long Short-Term Memory neural network (LSTM), and we name the corresponding power consumption model as TW_BP_PM, ENN_PM, and MLSTM_PM, respectively. Among them, the TW_BP_PM can be a high-precision server power consumption prediction model owing to its good nonlinear fitting, and takes into consideration the impact of the system performance state accumulation over a period of time on the server power consumption at the current moment. Compared to TW_BP_PM, the ENN_PM takes the state layer output of the previous step as part of the next input and the output of each state layer can be treated as the result of a cumulative change in the global system performance state. The MLSTM_PM (i.e., a multi-layer LSTM neural network) improves the long-term dependence problem, which is a common problem in RNN- based model, including ENN_PM, owing to the specific structure in LSTM unit. The gate control structure in LSTM unit can choose to memorize or forget the input and generated state information during the running process, and obtain useful information from the accumulation of global system performance states for further prediction. However, the complex control logic in the network brings about a huge increase in operating overhead and computing resources. This paper collects the relevant performance counters data and corresponding power consumption data, running with different workloads, and conducts experiments based on three proposed ANN- based models mentioned above to validate and compare. The primary contributions of this paper include:

1) In-depth and fine-grained analysis on the performance state and power consumption characteristics of three main subcomponents of the cloud server with four different types of task loads (i.e., CPU-intensive load, memory-intensive load, I/O-intensive load, and mixed load). Moreover, an ANN-based approach to model the server power consumption is proposed in view of the feature that tasks load running on the cloud server are complex and changeable;
2) Collecting data by a set of performance counters, we implement three proposed server power consumption model respectively based on BP neural network (TW_BP_PM), Elman neural network (ENN_PM), and multi-layer LSTM neural network (MLSTM_PM);
3) Through four types of workload benchmarks, the three ANN-based proposed models are comprehensively and fully evaluated by experiments, and the characteristics and applicable scenarios of the three models are summarized.

The rest of this paper is organized as follows. Part 2 introduces related research results of this article. Part 3 mainly focuses on the quantitative analysis of the performance and energy consumption characteristics of the system under different workloads, as well as the modeling methodology of based on ANN. Part 4 conducts comparisons and analysis through experiments on the proposed power consumption model, and Part 5 draws a conclusion on this paper work.

2 Related Work

With the continuous expansion of cloud datacenter scale, the complexity of energy consumption characteristics of datacenter raises dramatically, and it attracts a lot of research attention on energy consumption monitoring and prediction of cloud servers in the datacenter. The literature [14] summarized the current establishment principles of energy consumption models for cloud computing datacenters and the basic flow of energy modeling. The authors divided the energy consumption models into two categories, i.e., system utilization-based models and performance counters-based models (i.e., PMC-based model). The system utilization rate-based models establish the mathematical relationship between resource utilization and system energy consumption by collecting system resource usage under different workloads, which has the characteristics of simple, direct, low computational overhead and high portability. The PMC-based model takes the performance information monitored by a set of performance counters as input to establish the prediction model, most of which have higher prediction accuracy. Besides, according to the difference of regression techniques, the paper [14] divided the energy consumption models into two categories from another aspect, i.e., the linear models and the nonlinear models, and implemented experimental comparison analysis, which shows that the nonlinear models have higher prediction accuracies than those of the linear models, but with greater computational overhead. Literature [15] proposed a Web server energy consumption model, which is also a PMC-based model. The total energy consumption of the server consists of the major subcomponents, such as processors, disks, memory, networks, and other board components. Among them, the authors established the energy consumption model of the CPU according to different P-states of the processor. In addition, the authors also used the CFS algorithm [16] to simplify the input parameter number of the model and the K-Means algorithm to mitigate the effects of nonlinear factors on the input parameters, and verified that the model can achieve the best average error within 2 percent on Intel i7 and AMD Opteron platforms through experiments. In recent years, more and more energy management strategies and energy-saving techniques have been applied to servers, and the energy consumption behavior of servers has also undergone tremendous changes. After a profound research on the changes of server energy consumption curves between 2007 and 2010 provided by SPECpower_ssj2008, Hsu et al. [17] found that the simple linear function is insufficient to describe the energy consumption behavior of the server under different CPU load. By carrying out different mathematical function fitting experiments, the author found that the exponential function can fit the energy consumption behavior with different CPU load better than others, and has low overhead. But this model is only suitable for computationally intensive because it just considers the impact of CPU load without modelling the energy consumption of other components. Basmadjian et al. [18] proposed a general energy consumption model for common datacenters, considering ICT resources and their associated attributes that contribute to datacenter energy consumption. Due to the complexity of the modeling process, the author divided the design of the model into four parts, i.e., ICT resources, servers, storage, and services. Different from other researches, the authors respectively build the energy consumption models in terms of ICT resource categories and its hierarchy, obtaining 2$\sim$10 percent error by experiment tests.

Virtualization technologies, widely applied in cloud computing environments, abstract and transform various physical resources in the datacenter, such as servers, networks, memory, and storage, so that users can flexibly configure these physical resources for better applying. In a virtualized environment, it is impossible to directly measure the energy consumption of a virtual machine (VM) through external instruments, because a VM consists of one or more running processes which produce energy consumption by occupying system resource [4], and it is obviously infeasible to detect the real-time power consumption of one or more processes through external devices. Therefore, many studies on VM-based energy modeling and measurement are proposed. Kansal et al. [19] put forward an energy consumption measurement tool for VM, i.e., Joulmeter, in which the energy consumption model is essentially based on component utilization. In particular, Joulmeter can achieve the training and parameter initialization of the target energy model by obtaining data from the API provided by the Windows system, without other external devices or software. Another similar work is the VMeter, a VM-oriented energy modeling method proposed by Bohra et al. [20]. The authors took the power consumptions of CPU, cache, disk and DRAM as the main considerations, and classify the system loads into CPU-intensive processes and I/O-intensive processes according to the degree of correlation between sub-components. Then, the energy consumption of the two parts is weighted and summed to obtain the final model, obtaining 93 percent accuracy in experiments running benchmarks. In a virtualized environment, the dynamic reconfiguration of a virtual machine is a major factor affecting the virtual machine's energy consumption, and the configurations of the vCPU in a large degree determine the virtual machine's energy consumption performance. Lin et al. [8] found that although the inherent energy consumption behaviors of physical CPU and vCPU in the same physical server is very different, the energy consumption curves of multi-core vCPUs are similar to those of physical CPU. Therefore, they put forward a vCPU energy model based on vCPU core number of the virtual machine.

According to the difference of performance behavior and energy consumption among different workloads in the server, task-based energy consumption modelling is a finer granularity method, which is conducted at software application level. Based on three types of tasks, i.e., computationally intensive tasks, data intensive tasks, and communication intensive tasks, Chen et al. [21] conducted experiment and analysis according to the energy consumption characteristics in cloud computing systems, and proposed task type-based fine-grained energy consumption model. Zhou et al. [22] took respective energy consumption performances of processor units, memory, disk and network interfaces under different task loads

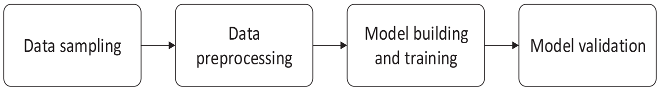


Fig. 1. The workflow of modeling.

into consideration, and established the fine-grained server energy consumption model by using the principal component analysis (PCA) and regression methods.

In recent years, more and more researchers are making their attempts to apply machine learning technologies to the studies of energy efficiency of datacenters, especially for server energy saving. In [23], the authors applied BP neural network (BPNN) and LSTM neural network to predict the server power of the datacenter. But the server energy consumption value in the experiment set is based on the analog value generated by other energy consumption models, rather than actual measured value, and it lacks certain credibility. Our works is a little bit similar to [23] (i.e., modeling by the BPNN and LSTM method), but the main difference is that we tried and applied three ANN structure to model and compare their performance and usability. In the literature [24], the authors believed that, with the passage of time, the fluctuation of energy consumption will have a certain impact on the subsequent system energy consumption, and proposed to establish the energy consumption model through deep learning method. By collecting the power consumption data, load fluctuation data, and system state data of the server in specified time unit, and performing the de-noising processing with the Detrended Wave Analysis (DFA) method, they respectively established two coarse- and fine-grained power consumption prediction models based on the autoencoder model (AE) and the recursive autoencoder model (RAE). Zhu et al. [25] used the Gaussian Mixture Model to cluster the energy consumption differences of resource characteristics in different degrees of utilization in the server, and adopted it to the regression prediction of energy consumption, with the experiment result showing that the model has higher accuracy, but takes a longer training time.

Regarding the researches on datacenter energy consumption modeling, the literatures [26], [27], [28] have conducted comprehensive analysis and summary about the existing research results. Compared with aforementioned energy modeling methods, this paper proposes three ANN-based energy consumption modeling methods, and validates these models under different types of workload through experiments.

3 ANN-Based Power Consumption Model

In this section, the general process of modeling the power consumption based on ANN will be briefly summarized. According to the process, we first outline how the feature extraction and selection are performed and we particularly analyze the performance of several major features under different workloads. Afterwards, the different power models accompanying their implementation details of three ANN-based structures, namely BPNN, ENN, and LSTM, are shown respectively.

3.1 The General Process of Modeling

The basic process of modeling power consumption is shown in Fig. 1, including four stages, i.e., data sampling, data preprocessing, model establishment and training, and model validation.

1) *Data Sampling*. L. Luo et al. [14] presented two basic data sampling methods, i.e., Processor performance counters and system utilization. In this paper, we employ the smart power meter and the performance counters provided by OS to collect data. As shown in Fig. 2, the testing server is connected to the power supply via the smart power meter (i.e., watts up? PRO), which can collect the server's power and log them inside its cache in a real-time manner. The logging device can fetch data by requesting from the meter. Besides, the testing server can also collect its own real-time performance status information via the performance counters.

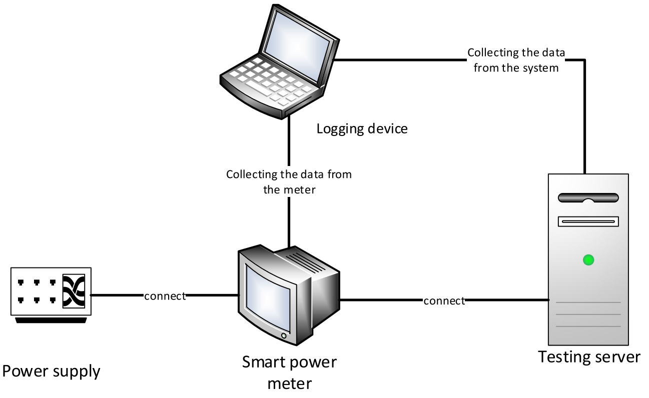


Fig. 2. Schematic diagram of data sampling process.

2) *Data Preprocessing*. During the preprocessing, the first step is to clean the raw data from the previous step, and remove the records with null value or abnormal value. Then, the two parts of cleaned data sets are merged together according to their timestamp as the original data set. In the end, feature filtering and analysis are conducted based on the original data, and a set of input features that have a great influence on the system power consumption are obtained. Besides, it should be noted that the data normalization is necessary and crucial because it can accelerate the speed of gradient descent [29] and may help improve the prediction accuracy [30]. Therefore, we employ the MinMax Normalization to normalize data with Eq. 1, where min and max respectively denote the minimum and maximum value in the data set, z_d denotes the value of original features, $\tilde{z}_d$ denotes the normalized value, and d denotes the number of input features.

$$\tilde{z}_d = \frac{z_d \ - \ min(z_d)}{max(z_d \ ) - min(z_d)} \ , \ d \ = \ 1, 2 \ldots n, \tag{1}$$

3) *Model Establishment and Training*. The detailed process of modeling will be described in Section 3.3. To explore the effectiveness of different ANN-based models for power consumption prediction, we develop three corresponding power models based on BPNN, ENN, and LSTM. We also implement model training with collected related data by simulating the actual production environments under different workloads with different types of benchmarks.

4) *Validation of Model and Power Prediction*. The experimental validation of the models will be discussed in Section 4. Comparisons and analysis will be conducted to verify the feasibility of the proposed ANN-based models from the following aspects, i.e., the prediction performance under different workloads, the training and running cost, comparison between proposed models and existing power consumption models.

TABLE 1
Table of Feature Parameters

Feature parameters	Description
Processor Time	The percentage of time the processor spends executing non-idle threads
User Time	The percentage of time the processor is in user mode
Privileged Time	The percentage of time the processor is executing code in privileged mode
Processor Utility	The amount of work the processor is doing
Priority Time	The percentage of time the processor spent executing non-low priority threads
Processor Performance	The average performance when the processor executes instructions
Commit Bytes in Use	The memory utilization
Available MBytes	The available memory capacity
Page/sec	The speed of reading or writing from disk to solve page errors.
Page Faults/sec	The average number of missing pages per second caused by interrupt
Disk Time	The percentage of time that the disk is busy reading or writing requests to provide services
Current Disk Queue Length	The current number of requests on the disk
Disk Bytes/sec	The bytes are transferred on the disk during read and write operations
Disk Transfer/sec	The read and write operation rate on disk
IO Data Bytes/sec	The bytes are written and read per second
IO Data Operation/sec	The number of I/O operations per second

3.2 Feature Extraction and Analysis

This section is divided into two parts to discuss: feature extraction and feature analysis. The results of the following analysis and discussion will help build up a clearer understanding between the characteristics of the performance parameters of main subcomponents in the server and its behavior of energy consumption, while running with different types of workloads.

3.2.1 Feature Extraction

The proposed models is a kind of PMC-based model which can reflect the system state more comprehensively and achieve better accuracy compared to the utilization based model. We will use a set of performance counters provided by Windows Operating System as input of our model (a total of 16 features). The specific input features are shown in Table 1.

According to the different resource requirements, we divide the workloads running on the servers into four types: CPU-intensive, memory-intensive, I/O-intensive, and mixed. Based on the above classification, we use different types of benchmarks to simulate the application load in the actual production environment, so as to obtain the performance state data of the system in the corresponding scenario.

As shown in Table 2, we use different benchmarks to simulate specific types of workloads. For the CPU-intensive workloads, we used the two benchmarks, Primeload and Grab-Ex. Primeload is developed by us to find all prime numbers in a range of N, supporting multithread execution, and Grab-Ex is a commonly used CPU stress testing tool, enabling to control the CPU load. For the memory-intensive workloads, the open source tool—RandMem, is applied, while IOzone is used in the experiment for I/O intensive load. Finally, PCMark7 is used to simulate mixed loads. The combination of productivity suite, computation suite, and system storage suite in PCMark7 can achieve an expectable simulation effect.

TABLE 2
Different Workloads' Corresponding Benchmarks

Workload types	Benchmarks
CPU-intensive	Primeload, Grad-Ex[1]
Memory-intensive	RandMem[2]
I/O-intensive	IOzone[3]
Mixed	PCmark7[4]

3.2.2 Feature Analysis

In this section, the performance of the three major subcomponents (CPU, memory, disk) in the server (here resource utilization is used as a reference indicator) and its power consumption characteristics are analyzed under four types of workloads. This will help better understand the impact of different types of workloads on the major energy-consuming components in the server and facilitate the next modeling. Fig. 3 shows the change in utilization of the three components (CPU, disk, and memory) in the server and the change in power consumption at the corresponding time under CPU-intensive load. We can see that in the case where disk utilization (maintain a range of 0 - 300) and memory utilization (37 - 38 percent) remain relatively stable, the CPU utilization changes from 0 to 100 percent and then returns to 0. The system's energy consumption fluctuates consistently and accordingly with the change of CPU utilization, which indicates that there is a strong positive correlation between them.

Fig. 4 shows the utilization change of the three components (CPU, disk, memory) and the power consumption change at the corresponding time in the case of running an IO-intensive load. When CPU utilization (from 5 to 10 percent) and memory utilization (from 38 to 43 percent) remain relatively stable, changes in disk I/O utilization (from 0 to 2000) have a relatively smaller impact on the power consumption. But it is observable that there existing some positive correlation in the three ranges of 0-200, 400-600, and 1200-1400.

Fig. 5 shows the utilization change of the three components (CPU, disk, memory) and the power consumption change at corresponding time when running a memory-intensive load. It is observed that the memory utilization change brings about the CPU utilization change, and the system power consumption fluctuates accordingly under the impacts of the both changes, indicating that the memory resources and CPU resources are closely related. Besides, the fluctuation of

[1] https://www.the-sz.com/products/cpugrabex/
[2] https://github.com/greenlsi/randmem
[3] http://www.iozone.org/
[4] https://benchmarks.ul.com/pcmark7

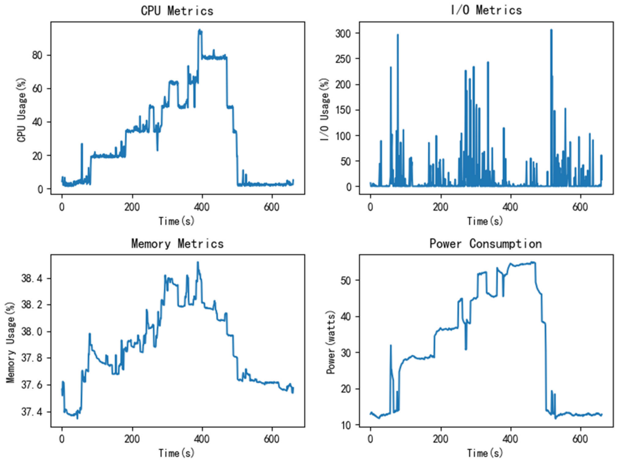


Fig. 3. The resource utilization and power consumption of the system under CPU-intensive workloads.

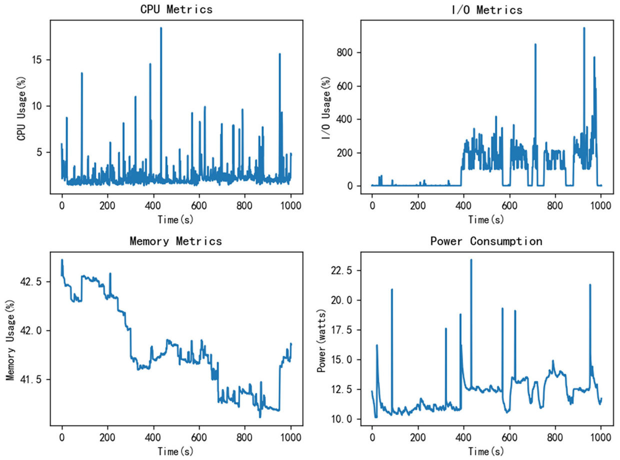


Fig. 4. The resource utilization and power consumption of the system under I/O-intensive workloads.

memory utilization will also drive the changes in disk performance (page breaks, paging operations), and ultimately take effect on the power consumption of the system.

Fig. 6 shows the utilization change of the three components and the power consumption change when running a Mixed workload. Compared with the above three experiments (more inclined to the stress test), the workload used in this set of tests is a hybrid superposition of the above three types of workloads, is mainly applied to simulate common production scenarios. It shows that more stable system performance and power consumption than those compare with experiments above. However, we can found that the contribution of CPU and memory to the overall system power consumption is larger, and the performance of I/O reach or exceed a threshold, there is a positive correlation between I/O and the corresponding power consumption.

In summary, this paper divides the task load of server in actual production environment into four types: CPU-intensive load, memory-intensive load, I/O-intensive load and mixed load. It is obvious that the power prediction model based on fixed mathematical formula and static parameters mentioned in the related work is difficult to adapt to the change of energy consumption characteristics caused by the change of load type in the server, poor in versatility, and rarely takes into consideration the impact of timing factors on the model prediction

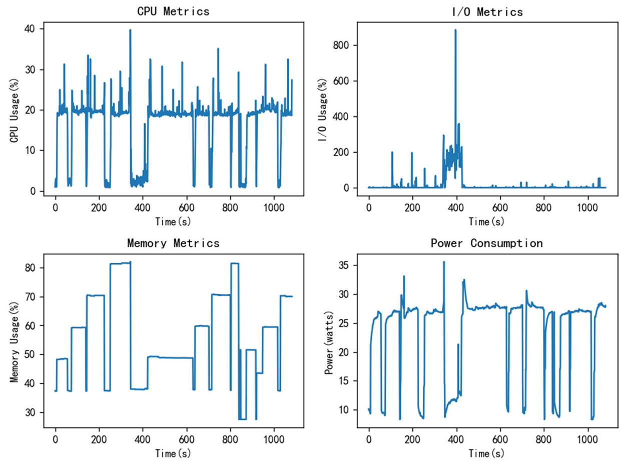


Fig. 5. The resource utilization and power consumption of the system under memory-intensive workloads.

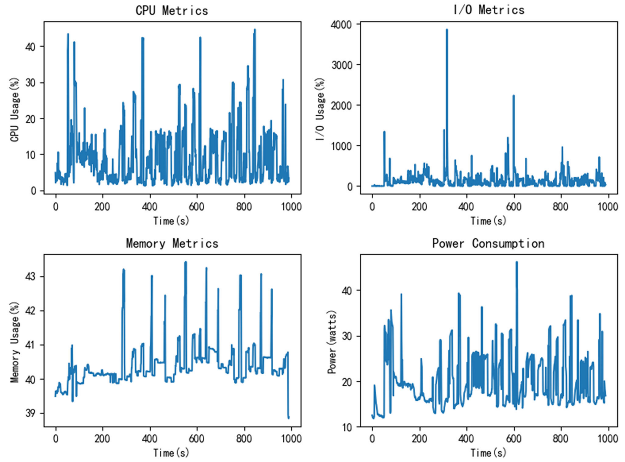


Fig. 6. The resource utilization and power consumption of the system under mixed workloads.

accuracy. Therefore, an ANN-based approach is proposed to establish power consumption prediction models and solve the problems above.

3.3 ANN-Based Cloud Server Power Consumption Model

In this section, we will elaborate on the three aspects of corresponding ANN's structural characteristics, computational logic and how to model the energy consumption of server.

3.3.1 *Power Consumption Model Based on Time Window and BP Neural Network*

In this subsection, TW_BP_PM, a power consumption prediction model, is developed based on time window and feed-forward neural network, shown in Fig. 7. Most of the power models mentioned in Section 2 take the system performance features of a single time as the model input, achieving favorable results. However, considering that the running process of the workload on the server is dynamically changing and time-correlated, which may also be reflected in the performance of the server and the power consumption changes, therefore, the concept of "time window" (TW) for model input is proposed.

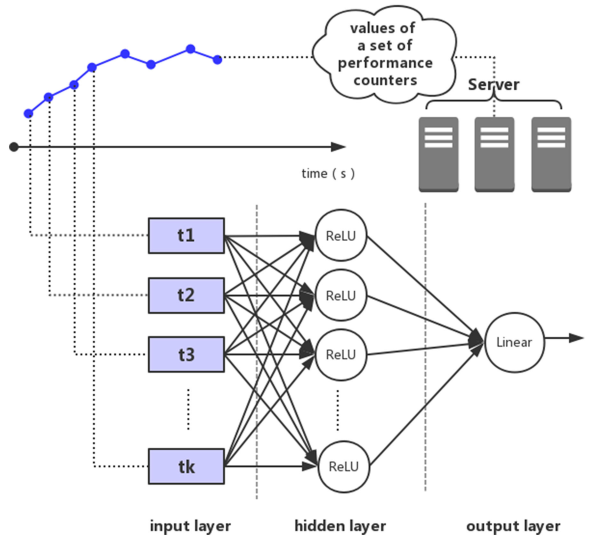


Fig. 7. The framework of TW_BP_PM power consumption model.

First, the symbol of n denotes the size of the time window (TW) and is an empirical constant. Second, a set of system state features collected at time t is defined as P_t, and the group of system state features in TW_t is defined as $[\mathrm{P_{t-n+1}},\ \mathrm{P_{t-n+2}},\ \ldots,\ \mathrm{P_t}]^T$, a column vector of size $n \times 16$. Then TW_t is utilized as the input of the model to predict the system power consumption at time t. In addition, we built a three-layer fully connected neural network, including input layer, hidden layer and output layer. The dimension d_{input} of the input layer is equal to the dimension of TW_t, and there are 25 neurons in the hidden layer. The connection weight between the input layer and the hidden layer is W_1, a matrix of $d_{input} \times 25$, and the output layer is a linear unit, i.e., the predicted power consumption value of outputs. The connection weight between the hidden layer and the output layer is denoted as W_2, and W_2 is a 25×1 matrix. Then, the feed forward computation process of the network is given, as shown in Eqs. 2, 3, and 4, where ${TW_t}^T$ is the transposed row vector of TW_t, B_1 and B_2 are respectively the bias, f is the activation function, and Out_2 represents the final output of the network.

$$L_1 = {TW_t}^T \times W_1 + B_1 \tag{2}$$

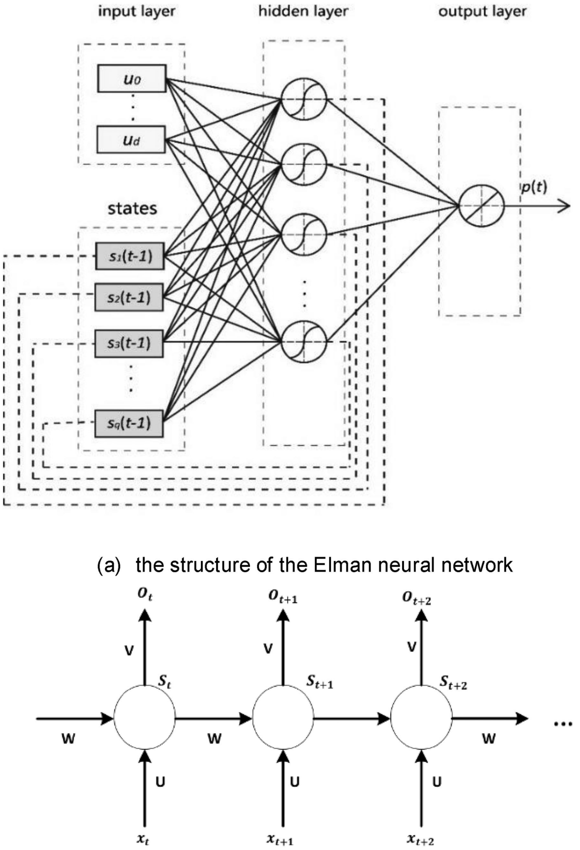
$$Out_1 = f(L_1) \tag{3}$$

$$Out_2 = Out_1 \times W_2 + B_2. \tag{4}$$

Krizhevsky et al. [31] found that when using ReLU, the convergence rate of the stochastic gradient descent algorithm (SGD) is faster than those of sigmoid and tanh, with lower the computational complexity of ReLU. In terms of the above two points, we chose ReLU as the activation function for the hidden layer, and the output layer is linearly output without activation function. We adopt back propagation algorithm to train our network, Mean Square Error (MSE) as the loss function, and L2 regularization and Early stopping to prevent model overfitting.

3.3.2 Power Consumption Model Based on Elman Neural Network

Recurrent Neural Network (RNN) is a kind of artificial neural network which is usually used to process time series data. Time series data, which is a sequence of data collected from different points of time in order, reflects the status or extent of a thing or phenomenon over time. The Elman neural network (ENN) is a common RNN and widely used in speech processing. As shown in Fig. 8a, unlike the BP neural network mentioned above, Elman neural network will use the output of the status layer as part of the next input in order to learn the information contained in the previous input sequence. Therefore, the previous step output from the state layer will be recycled as the part of next input in the process of forward propagation. When the size of input sample is large enough, the entire network can be unfolded in terms of the time dimension into a deep neural network as shown in Fig. 8b.



(b) the unfolding of Elman neural network in time dimension

Fig. 8. The realization of the power consumption model based on elman neural network.

The collected data from a group of performance counters and corresponding power consumption in time order is feed into the Elman network model (ENN_PM) for predicting the real-time power consumption of the server. As shown in Fig. 8a, U_t denotes the value of certain performance counter at moment t. Let $X_t = (U_t,\ U_{t+1} \ldots U_{t+d-1})$ be a part of network input, S_t and P_t be the output of hidden layer and output layer respectively. The weights of input layer, hidden layer and output layer are respectively denoted with U, W, and V, shown in Fig. 8b. The procedure can be described as follows:

$$O_t = UX_t + WS_{t-1} + B_1 \tag{5}$$

$$S_t = f(O_t) \tag{6}$$

$$P_t = VS_t + B_2, \tag{7}$$

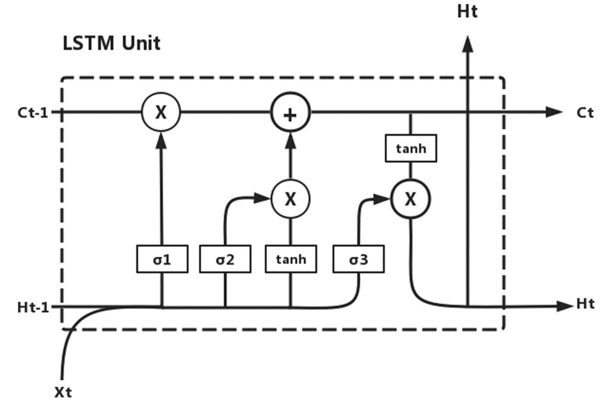


Fig. 9. The internal structure of the LSTM unit.

Where B_1, B_2 is bias terms. It's observed that the output of the network is related to not only the external input, but also the state layer output of previous step.

As a kind of RNN, ENN is usually trained by Back Propagation Through Time algorithm (BPTT). The main idea of BPTT is that searching for the optimal point of the trainable parameters in negative gradient direction until convergence, which is similar to the BP algorithm. In BPTT, the RNN structure will be first unfolded as a common deep neural network and then back propagation algorithm will be applied. However, the network has the problem of vanishing or exploding gradients when the size of input data is too large. To address the problem, we choose the truncated Back Propagation Through Time algorithm (TBPTT) to optimize training process, the general idea of it is that limit the gradient move distance (i.e., setting the time step to limit the number of propagation steps) during back propagation. In addition, we use L2 regularization and Early Stopping methods to avoid over-fitting.

3.3.3 Power Consumption Model Based on Multi-Layer LSTM Network

Long Short Term Memory Neural Network (LSTM) [32] is a variation of RNN, capable of learning long-term dependencies from input data which is a common problem [33] in general RNN methods. Fig. 9 provides the internal structure of an LSTM unit, containing three gates σ_1, σ_2, σ_3, and two tanh activation functions. X_t represents the external input of LSTM at time t, H_t is the state output of LSTM and C_t is the final output of LSTM at time t.

The update of an LSTM unit can be described as follows:

$$f_t = \sigma_1 \left(W_f[H_{t-1}, X_t] + b_f\right) \tag{8}$$

$$i_t = \sigma_2 \left(W_i[H_{t-1}, X_t] + b_i\right) \tag{9}$$

$$\tilde{C}_t = \tanh(W_c[H_{t-1}, X_t] + b_C) \tag{10}$$

$$o_t = \sigma_3 \left(W_o[H_{t-1}, X_t] + b_o\right) \tag{11}$$

$$\boldsymbol{C_t = f_t * C_{t-1} + i_t * \tilde{C}_t} \tag{12}$$

$$H_t = o_t * \tanh(C_t), \tag{13}$$

Where W_f, W_i, W_c, W_o and b_f, b_i, b_C, b_o are parameters to learn in LSTM units. Each LSTM unit saves the state output, use gates to control whether to drop old state information and add new state information.

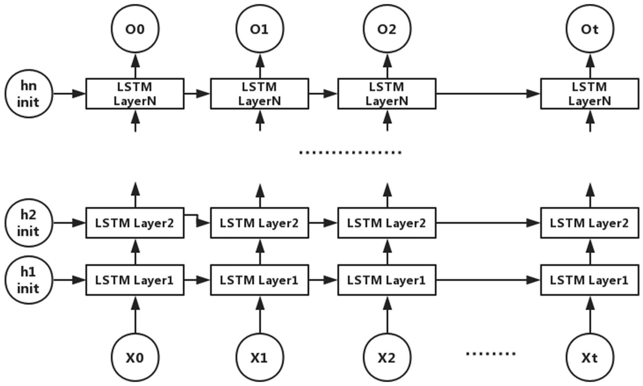


Fig. 10. The structure of the MLSTM_PM in time dimension.

In this paper, we proposed a multi-layer LSTM model for power consumption prediction, called MLSTM_PM, the structure of this model with time dimension unfolding shown in Fig. 10. It is seen that the process of this model consists of three steps, i.e., the initialization of each unit state in each LSTM layer, the feeding of the collected data from performance counters in time order, and the prediction of power consumption at each comment. In MLSTM_PM, the number of LSTM layer is 2 and each layer has 10 LSTM units. The truncated Back Propagation Through Time algorithm is chosen to optimize training process and L2 regularization, Early Stopping methods also used to avoid over-fitting.

4 Experiments

In this section, we will conduct experiments to test and analyze the prediction accuracy and usability of the proposed ANN-based models in Section 3.3. In order to evaluate the performance of the three models under different types of workloads, the benchmarks mentioned in Table 2 is adopted to simulate the characteristics of different workloads in the actual environment, and then the performance and power data is collected as the original data set of all the experiments. The experiments will be conducted in three aspects, including independently validation and analysis for each purposed model, comparative experiment with the introduction of existing power models and overhead comparison among the proposed models.

4.1 Experimental Setup

The experiments are conducted on the Dell Precision 3520 workstation, equipped with Intel Core I7-7700H processor, DDR4 8 GB of memory, 1T capacity of disk and 7200 rpm of speed. The power data of the server is collected by the external power meter connected between the power supply and server. We apply the benchmarks mentioned in Table 2 to stimulate the actual environment and obtain the data from a set of performance counters mentioned in Table 1 which is offered by Microsoft Windows 10 operating system. This data set contains 2247 records of CPU intensive workload, 1907 records of memory intensive load, 2847 records of I/O intensive workload and 4053 records of mixed workload. We implement the proposed models in the Tensorflow framework and divide the data set into training, validation and testing set by 75, 5, 20 percent respectively.

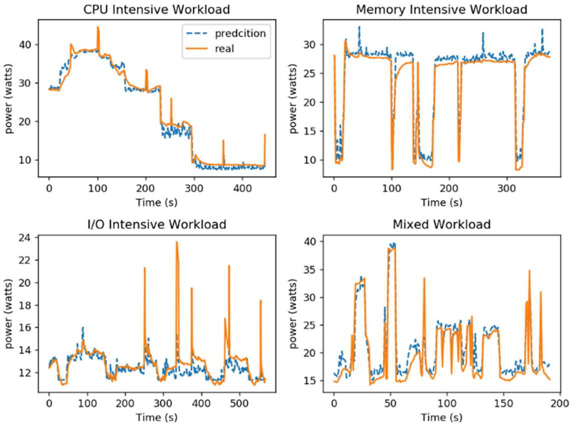


Fig. 11. Real-time power consumption prediction of TW_BP_PM under different types of workloads.

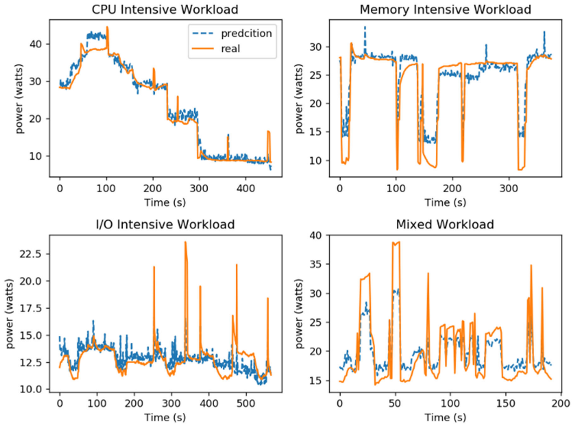


Fig. 12. Real-time power consumption prediction of ENN_PM under different types of workloads.

4.2 Experiment and Analysis of Each ANN-Based Model

In this part of the experiment, the proposed ANN-based models will be trained and tested under the different types of workload (CPU intensive workload, memory intensive workload, I/O intensive workload and mixed workload) and the result will be collected to evaluate the prediction accuracy of each ANN-based power model.

a) *TW_BP_PM*. The architecture of TW_BP_PM is shown as Fig. 7. The neural network has 3 layers, including input layer, hidden layer and output layer. The input layer has 16 neurons and time window, size of which is set to 2, is used to shape the input data. The hidden layer has 25 neurons. In the process of training, the maximum epoch is set to 300. Early stopping and L2 regularization, setting to 0.001, are used to prevent over-fitting.

TW_BP_PM is trained and tested under four types of workloads. As shown in Fig. 11, the results are collected and we compare the difference between the prediction value and the real value.

According to Table 3, it can be seen that the power model performs better under the CPU and I/O intensive workloads than the other two workloads with mean relative error of 6.7 and 4.1 percent. In general, the mean absolute error of TW_BP_PM is within 2W.

b) ***ENN_PM***. The architecture of ENN_PM is shown as Fig. 8. It has the same layers as TW_BP_PM, but the output of the hidden layer will be used as part of next input in ENN_PM. The number of neurons of hidden layer is set to 25. In the process of training, the maximum epoch is set to 50 and L2 regularization coefficient is set to 0.01. The TBPTT algorithm is applied to train the model, where the value of time step is set to 1. As shown in Fig. 12, the prediction value by ENN_PM and the real value are compared under four types of different workloads.

As shown in Table 4, the ENN_PM have mean relative error of 7.3 and 6.2 percent under the CPU intensive workload and I/O intensive workload, but the prediction error under memory intensive workload and mixed workload fluctuated greater than the other two. From the Fig. 12, it shows that ENN_PM cannot well predict the peak power consumption the idle power consumption, which affects the average prediction accuracy.

c) ***MLTSM_PM***. The architecture of MLSTM is shown as Fig. 10. The network consists of an input layer, two hidden layers and an output layer. Each neuron in the hidden layer is a LSTM unit, of which structure is shown in Fig. 9, as well as the number of neurons of each hidden layer is set to 10. In the process of training, the maximum epoch is set to 100 and the value of time step when running the TBPTT algorithm is set to 2. As shown in Fig. 13, the estimate power predicted by MLSTM_PM and the real power are compared under four types of different workloads.

As shown in Table 5, the mean relative error of the power consumption predicted by MLSTM_PM is within 10 percent, which indicates that MLSTM_PM can well adapt to the changes and fluctuations of various types of workloads. In addition, MLSTM_PM can reach the mean absolute error within 2W under four types of workloads.

Based on the above experiment, it can be seen that the three ANN based power models can almost reach the mean relative error of less than 10 percent and the mean absolute error of about 2W under different types of workload. The overall prediction performance of TW_BP_PM is slightly better than that of two RNN-based power models (ENN_PM and MLSTM_PM). Meanwhile, MLSTM has a better prediction accuracy

TABLE 3
Prediction Error of TW_BP_PM

Workload type	Mean Relative Error (MRE)	Mean Absolute Error (MAE)
CPU Intensive	6.7%	1.17W
Memory Intensive	7.1%	1.21W
I/O Intensive	4.1%	0.59W
Mixed	8.6%	1.60W

TABLE 4
Prediction Error of ENN_PM

Workload type	Mean Relative Error (MRE)	Mean Absolute Error (MAE)
CPU Intensive	7.3%	1.48W
Memory Intensive	13.6%	1.92W
I/O Intensive	6.2%	0.84W
Mixed	11.9%	2.49W

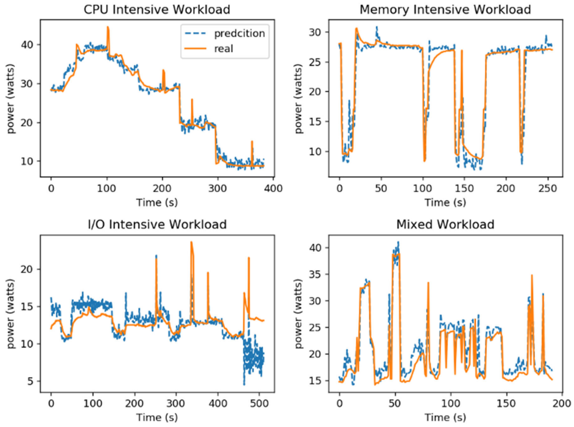


Fig. 13. Real-time power consumption prediction of MLSTM_PM under different types of workloads.

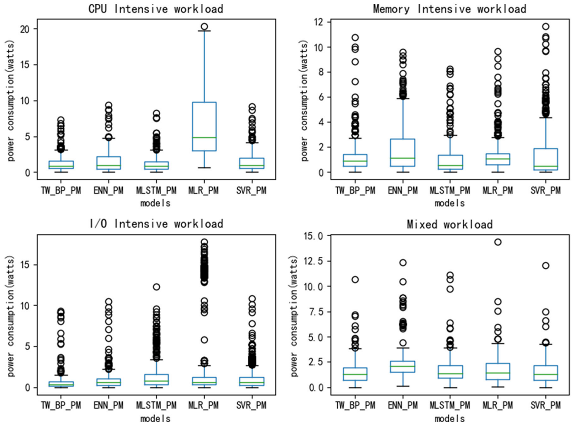


Fig. 14. The error distribution of each model under different types of workload.

TABLE 5
Prediction Error of ENN_PM

Workload type	Mean Relative Error (MRE)	Mean Absolute Error (MAE)
CPU Intensive	5.8%	1.15W
Memory Intensive	7.2%	1.14W
I/O Intensive	10%	1.4W
Mixed	9.3%	1.7W

than ENN_PM, which exist greater error when predicting the peak consumption and the idle power consumption of servers.

4.3 Comparative Experiment and Analysis

In this part of experiment, multiple linear regression (MLR, represent linear model) and support vector regression (SVR, represent non-linear model) are introduced to build the power consumption models for comparative experiment. Fig. 14 shows that the absolute error distribution of five power consumption models under four types of different workloads. Under the CPU intensive workload, the average error of the MLR based power models is larger than the other four models, and it can be seen that the average error of TW_BP_PM and MLSTM_PM is smaller (three-quarters of the data has an absolute error of less than 2.5w), and their distribution area of outliers is also smaller than the other three models. Under the memory intensive workload, all five models have a certain number of predicted outliers, which was related to the fluctuation of the workload. Among the five power models, the MLSTM_PM has the best prediction accuracy, followed by MLR based power model and ENN_PM. Under the predicted I/O intensive load, it can be seen that TW_BP_PM has the best prediction error distribution, followed by ENN_PM. The absolute error of most prediction results of these two models is less than 1.25w. However, it can be seen from Fig. 14 that all five models' results exist a certain number of predicted outliers. Under mixed load, the mean absolute error of the forecast results of the five models can reach less than 4W, and the mean error of TW_BP_PM and MLSTM_PM is smaller than that of the other three models (three quarters of the forecast data can reach the forecast error below 3W).

As shown in Figs. 15 and 16, it can be seen that under different workloads, the MRE of four other power models are

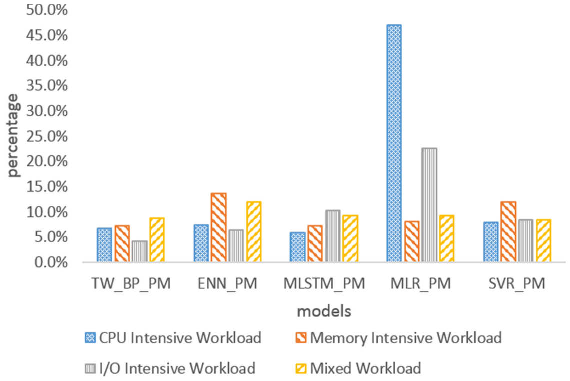


Fig. 15. Mean relative error of each power model.

below 10 percent as well as the MAE of ANN based power model is less than 3W. The ANN based power model has better accuracy than the MLR based and SVR based power models and can better adapt to the changes and fluctuations of various workloads.

4.4 Comparison of Models Overhead

In the actual production environment, the usability of the proposed model (e.g., the training and execution time, the needed CPU load when training and running) is one of the key points of concern besides the prediction accuracy. As shown in Table 6, we select the elapsed time of training process and its corresponding CPU load, as well as the execution time of a single input and its corresponding CPU load as the indicators to evaluate the usability of the model. The three ANN-based models were trained and tested on the same machine with Intel i7-6498 processor and 8G memory, based on the same data set mentioned above.

As shown in Table 6, it can be seen that ENN_PM has a faster convergence speed compared with the other two models in the training process, and the training time is less than 10s. In addition, the training time of MLTSM_PM is second only to ENN_PM, and the model convergence speed of TW_BP_PM is the slowest among them. All the trained ANN-based power models require less than 10^{-4} seconds to complete a single prediction, which indicates that the model proposed in this paper can realize real-time prediction of server power consumption. In addition, CPU load during the training and execution is also considered. In the process of the

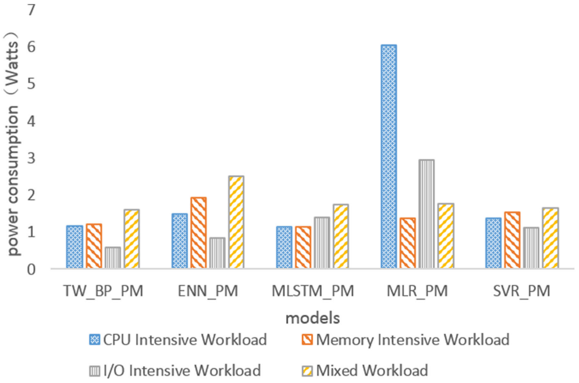


Fig. 16. Mean absolute error.

training, the CPU load of ENN_PM is smaller than the TW_BP_PM's and MLSTM_PM's, because of the simpler network structure and smaller input dimensions of ENN_PM. The CPU loads of the three ANN- based models in this paper remain at an average of about 10 percent, and the CPU loads of ENN_PM and MLSTM_PM are about 7 percent on average.

5 Conclusion

In this paper, we proposed a datacenter cloud server-oriented energy consumption model based on three different types of ANNs (i.e., TW_BP_PM, ENN_PM and MLSTM_PM). First, we divide the workloads of cloud server operations in actual production scenarios into four categories, namely CPU-intensive load, memory-intensive load, I/O-intensive load, and mixed load. On the basis of the above classification, we generated and simulated the running status of these loads in the system with corresponding benchmarks, collected the system performance status in real time through a set of performance counters, and analyzed the characteristics of performance and energy consumption of sub-components in the server under different workloads. Among the established three ANN-based power consumption models, TW_BP_PM is a real-time power prediction model using a combination of time window and BP neural network. ENN_PM is based on Elman neural network, a kind of RNN, which takes the state layer output of the network in the last moment as a part input of the current time model, and implements power consumption prediction by cycling this process. MLSTM_PM established the model based on LSTM unit, which can effectively avoid the long-term dependence of the general RNN with better prediction accuracy, but the complex computational logic inside LSTM makes the computational overhead of the entire model larger. In the end, we conducted some experiments on the three ANN-based power consumption models, i.e., the evaluation of prediction accuracy of each single model under different workloads, the comparison between the ANN model and other typical power consumption prediction models, and the usability comparison of the ANN model. Among them, TW_BP_PM and MLSTM_PM have better performance in overall prediction accuracy, with average prediction error less than 1W. But the training convergence speed of the former is slower, and the latter's operation logic is more complicated, resulting in longer training time or occupying larger CPU operation resources, having the characteristics of larger overhead but higher prediction accuracy. Owing to the faster convergence speed and simpler network structure, although the fluctuation of prediction error is large while running memory-intensive load and mixed load, the overall average relative error of the Elman neural network-based power consumption model can be controlled within 10 percent, and its average absolute error is less than 3W, with lower overhead.

TABLE 6
Comparison of Three ANN Based Model's Overhead

	TW_BP_PM	ENN_PM	MLSTM_PM
Training time (sec)	$\approx$27.3	$\approx$5.6	$\approx$11.5
The execution time of a single input (sec)	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$
The CPU load during training (%)	63%	25%	68%
The CPU load during execution (%)	12%	8%	7%

Acknowledgments

This research work is partially supported by the National Natural Science Foundation of China (Grant Nos. 61772205, 61872084), Science and Technology Planning Project of Guangdong Province (Grant Nos. 2017B010126002, 2016A010101018, and 2016A010119171), Guangzhou Science and Technology Projects (Grant Nos. 201907010001, 201807010052 and 201610010092), Nansha Science and Technology Projects (Grant No. 2017GJ001) and the Fundamental Research Funds for the Central Universities, SCUT.

References

[1] There are Now Close to 400 Hyper-Scale Data Centers in the World [EB/OL]. 2017-12-22. Available from: https://www.datacenterknowledge.com/cloud/research-there-are-now-close-400-hyper-scale-data-centers-world.

[2] America's Data Centers Consuming and Wasting Growing Amounts of Energy [EB/OL]. 2015-02-06. [Online]. Available: https://www.nrdc.org/resources/americas-data-centers-consuming-and-wasting-growing-amounts-energy

[3] M. Avgerinou, P. Bertoldi, and L. Castellazzi, "Trends in data centre energy consumption under the European code of conduct for data centre energy efficiency," *Energies*, vol. 10, no. 10, 2017, Art. no. 1470.

[4] W. W. Weiwei Lin, "Energy consumption measurement and management in cloud computing environment," *J. Softw.*, vol. 27, no. 4, pp. 1026–1041, 2016.

[5] J. C. McCullough, Y. Agarwal, J. Chandrashekar, S. Kuppuswamy, A. C. Snoeren, and R. K. Gupta, "Evaluating the effectiveness of model-based power characterization," in *Proc. USENIX Annu. Tech. Conf.*, 2011, Art. no. 12.

[6] W. Wu, W. Lin, and Z. Peng, "An intelligent power consumption model for virtual machines under CPU-intensive workload in cloud environment," *Soft Comput.*, vol. 21, no. 19, pp. 5755–5764, 2017.

[7] C.-H. Hsu and S. W. Poole, "Power signature analysis of the SPECpower_ssj2008 benchmark," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw.*, 2011, pp. 227–236.

[8] W. Lin, W. Wu, H. Wang, J. Z. Wang, and C.-H. Hsu, "Experimental and quantitative analysis of server power model or cloud data centers," *Future Gen. Comput. Syst.*, vol. 86, pp. 940–950, 2018.

[9] Y. C. Chang, R. S. Chang, and F. W. Chuang, "A predictive method for workload forecasting in the cloud environment," *Adv. Technol. Embedded Multimedia Hum.-Centric Comput.*, vol. 260, pp. 577–585, 2014.

[10] S. Gupta, V. Singh, A. P. Mittal, and A. Rani, "Weekly load prediction using wavelet neural network approach," in *Proc. 2nd Int. Conf. Comput. Intell. Commun. Technol.*, 2016, pp. 174–179.

[11] J. Kumar and A. K. Singh, “Workload prediction in cloud using artificial neural network and adaptive differential evolution,” *Future Gen. Comput. Syst.*, vol. 81, pp. 41–52, 2017.

[12] Y. Zuo, Y. Wu, G. Min and L. Cui, “Learning-based network path planning for traffic engineering,” *Future Gen. Comput. Syst.*, vol. 92, pp. 59–67, 2019.

[13] C. Huang, G. Min, Y. Wu, Y. Ying, K. Pei, and Z. Xiang, “Time series anomaly detection for trustworthy services in cloud computing systems,” *IEEE Trans. Big Data*, to be published, doi: 10.1109/TBDATA.2017.2711039.

[14] L. Luo, W.-J. Wu, and F. Zhang, “Energy modeling based on cloud data center,” *J. Softw.*, vol. 25, no. 7, pp. 1371–1387, 2014.

[15] L. Piga, R. A. Bergamaschi, and S. Rigo, “Empirical and analytical approaches for web server power modeling,” *Cluster Comput.*, 2014, vol. 17, no. 4, pp. 1279–1293.

[16] M. Hall, “Correlation-based feature selection for machine learning,” PhD Thesis, Waikato University, 1998, 19.

[17] C. H. Hsu and S. W. Poole, “Power signature analysis of the SPECpower_ssj2008 Benchmark[C],” in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw.*, 2011, pp. 227–236.

[18] R. Basmadjian, N. Ali, F. Niedermeier, H. D. Meer, and G. Giuliani, “A methodology to predict the power consumption of servers in data centres,” in *Proc. ACM SIGCOMM Int. Conf. Energy-Efficient Comput. Netw.*, 2011, pp. 1–10.

[19] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, “Virtual machine power metering and provisioning,” in *Proc. ACM Symp. Cloud Comput.*, 2010, pp. 39–50.

[20] A. E. H. Bohra and V. Chaudhary, “VMeter: Power modelling for virtualized clouds,” in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Workshops Phd Forum*, 2010, pp. 1–8.

[21] F. Chen, J. Grundy, Y. Yang, J. G. Schneider, and Q. He, “Experimental analysis of task-based energy consumption in cloud computing systems,” *Proc. ACM/Spec Int. Conf. Perform. Eng.*, 2013, pp. 295–306.

[22] Z. Zhou, J. H. Abawajy, F. Li, Z. Hu, M. U. Chowdhury, A. Alelaiwi, et al., “Fine-grained energy consumption model of servers based on task characteristics in cloud data center,” *IEEE Access*, vol. 6, no. 99, pp. 27080–27090, 2018.

[23] N. Liu, X. Lin, and Y. Wang, “Data center power management for regulation service using neural network-based power prediction,” in *Proc. IEEE 18th Int. Symp. Quality Electron. Des.*, 2017, pp. 367–372.

[24] Y. Li, H. Hu, Y. Wen, and J. Zhang, “Learning-based power prediction for data centre operations via deep neural networks,” *Proc. 5th Int. Workshop Energy Efficient Data Centres*, 2016, Art. no. 6.

[25] H. Zhu, H. Dai, S. Yang, Y. Yan, and B. Lin, “Estimating power consumption of servers using gaussian mixture model,” in *Proc. 5th Int. Symp. Comput. Netw.*, 2017, pp. 427–433.

[26] H. Cheung, S. Wang, C. Zhuang, and J. Gu, “A simplified power consumption model of information technology (IT) equipment in data centers for energy system real-time dynamic simulation,” *Appl. Energy*, vol. 222, pp. 329–342, 2018.

[27] M. Dayarathna, Y. Wen, and R. Fan, “Data center energy consumption modeling: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 732–794, Jan.-Mar. 2017.

[28] J. C. Mccullough, Y. Agarwal, J. Chandrashekar, S. Kuppuswamy, A. C. Snoeren, and R. K. Gupta, “Evaluating the effectiveness of model-based power characterization,” *Proc. Usenix Conf. Usenix Tech. Conf.*, 2011, Art. no. 12.

[29] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

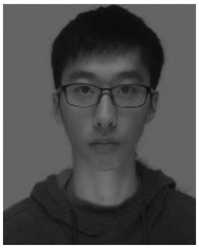
[30] P. Juszczak, D. Tax, and R. P. Duin, “Feature scaling in support vector data description[C],” in *Proc. ASCI*, 2002, pp. 95–102.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 84–90.

[32] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[33] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

Weiwei Lin received the BS and MS degrees from Nanchang University, in 2001, and 2004, respectively, and the PhD degree in computer application from the South China University of Technology, in 2007. Currently, he is a professor in the School of Computer Science and Engineering, South China University of Technology. His research interests include distributed systems, cloud computing, big data computing, and AI application technologies. He has published more than 80 papers in refereed journals and conference proceedings. He is a senior member of CCF.



GuangXin Wu received the BE degree in computer science from the South China University of Technology, in 2018. He is currently working toward the master’s degree in computer science at the South China University of Technology. His research interests include search engine and cloud computing.

Xinyang Wang received the PhD degree from the South China University of Technology. He has authored and coauthored some papers in areas of parallel and distributed system architecture, parallel computing, and network topology properties. His research interests are mainly on computer network topology, parallel computing, cloud computing, heterogeneous data integration, and network fault-tolerance.

Keqin Li is a SUNY distinguished professor of computer science at the State University of New York. He is also a distinguished professor of the Chinese National Recruitment Program of Global Experts (1000 Plan) at Hunan University, China. He was an Intellectual Ventures endowed visiting chair professor at the National Laboratory for Information Science and Technology, Tsinghua University, Beijing, China, during 2011-2014. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU-GPU hybrid and cooper-ative computing, multicore computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, Internet of things, and cyber-physical systems. He has published more than 530 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He is currently or has served on the editorial boards of *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Computers*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Services Computing*, and *IEEE Transactions on Sustainable Computing*. He is a fellow of the IEEE.