IECL: An Intelligent Energy Consumption Model for Cloud Manufacturing

Zhou Zhou, Mohammad Shojafar, *Senior Member, IEEE*, Mamoun Alazab, *Senior Member, IEEE*, and Fangmin Li

***Abstract*—The high computational capability provided by a data centre makes it possible to solve complex manufacturing issues and carry out large-scale collaborative cloud manufacturing. Accurate, real-time estimation of the power required by a data centre can help resource providers predict the total power consumption and improve resource utilisation. To enhance the accuracy of server power models, we propose a real-time energy consumption prediction method called IECL that combines the support vector machine, random forest, and grid search algorithms. The random forest algorithm is used to screen the input parameters of the model, while the grid search method is used to optimise the hyperparameters. The error confidence interval is also leveraged to describe the uncertainty in the energy consumption by the server. Our experimental results suggest that the average absolute error for different workloads is less than 1.4% with benchmark models.**

***Index Terms*—Cloud manufacturing, power model, data centre, energy consumption prediction, support vector machine.**

I. Introduction

Cloud manufacturing is a new method of manufacturing that utilises network and cloud manufacturing service platforms to control online manufacturing based on the user's requirements, and to provide various on-demand manufacturing services. Cloud-based manufacturing allows enterprises to constantly improve the production process, adjust the production structure, and improve the efficiency and product quality.

As a critical component of the infrastructure of cloud manufacturing, the data centre plays an essential role. The high computational capability provided by a data centre makes it possible to solve complex manufacturing issues and carry out large-scale collaborative cloud manufacturing. To meet the growing demand for high-performance computing, the sizes of data centres are increasing. According to the latest data from Synergy Research [1], the total number of large-scale data centres operated by the world's 20 cloud service providers has increased to 597, representing double the total in 2015.

This rapid growth in the number of data centres gives rise to two issues [2]. The first is an increase in energy costs to service providers [3]. Statistical results [4] show that in 2018, the total power consumption of China's data centres was 150 billion kWh, accounting for 2% of the total social power consumption. This is expected to double to 4.05% by 2025. The second issue is the heavy social pressure in regard to environmental protection. The Global e-Sustainability Initiative (GeSI) reported that the global carbon emissions from data centres accounted for about 2% of the total global carbon emissions in 2020, equivalent to the carbon emissions of the global aviation industry [5].

Alo ngside this enormous energy consumption, there is low resource utilisation. Statistical reports [6], [7]suggest that resource utilisation by data centres is on average between 5% and 25%, leading to wastage of resources. In general, the energy consumption by data centres is mainly associated with servers, refrigeration systems, lighting, and other equipment. Power consumption by servers accounts for more than 50% of the energy required by the whole data centre [8],and the power consumption of the other components is also related to that of the servers. Hence, reducing server power consumption can help service providers to optimise the overall power consumption within a data centre [9].

A data centre may consist of a large number (hundreds or thousands) of servers. To improve resource utilisation and decrease the energy consumption costs, energy-aware optimisation algorithms are adopted [9], [10]. The issue of how to evaluate the quality of these energy-saving optimisation algorithms is an important one. In practice, an energy-aware optimisation algorithm is based on a specific power model, and hence the server power prediction model forms the basis of the energy-saving optimisation algorithm. The accuracy of the model is directly related to the quality of the optimisation algorithm [11], [12]. Accurate, real-time estimation based on the power consumption model for a data centre can therefore help resource providers to predict the total power consumption, improve resource utilisation, and reduce the energy consumption costs of the cloud manufacturing enterprise.

A. Goal and contributions of the paper

Energy consumption measurements of a data centre server can not only be used to establish a power model, but can also provide a guide for cloud manufacturing. The goal of this paper is to design a high-precision server energy consumption model. To achieve this, we propose a server energy consumption model called *IECL*. Unlike existing methods, our model can effectively predict the energy consumption for different types of workload. The main contributions of this paper are as follows:

Z. Zhou and F. Li are with the School of Computer Engineering and Applied Mathematics, Changsha University, Changsha, 410003, China (e-mail: zhzhou@ccsu.edu.cn, lifangmin@whut.edu.cn)

M. Shojafar is with the 5GIC & 6GIC, Institute for Communication Systems (ICS), University of Surrey, Guildford, GU27XH, United Kingdom (e-mail: m.shojafar@surrey.ac.uk)

M. Alazab is with the Charles Darwin University, Australia (e-mail: alazab.m@ieee.org)

Manuscript received 19 December 2021; revised 25 March 2022; accepted 02 April 2022

- We use the random forest (RF) algorithm for feature selection. We leverage this algorithm to screen feature parameters and select *the main representative parameter* for different types of tasks.
- We use the grid search (GS) method to optimise the hyper-parameters of the model.
- We propose a real-time energy consumption prediction method based on support vector machine (SVM). Unlike other approaches, the proposed method can effectively handle changes in the type of workload.
- We perform an extensive and comprehensive evaluation of our model on CPU-intensive, web transaction and I/O-intensive tasks. Our experimental results demonstrate that the prediction accuracy of the proposed energy consumption model is higher than other benchmark models.

B. Structure of the paper

The rest of the paper is organised as follows. Section II describes related works. Section III introduces our IECL model. Section IV presents our experimental results and an analysis. Finally, Section V concludes the paper and suggests directions for future work.

II. Related work

The construction of an accurate, real-time power consumption model can help cloud manufacturing enterprises predict and optimise the power consumption of a data centre, reduce energy consumption costs and increase the return on investment for the enterprise [10]. Existing approaches can be divided into *three* types based on the different modelling objects used: cloud platform, server, and virtual machine (VM) energy consumption models. In the following, we describe these three power modelling methods.

The first type is a cloud platform energy consumption model. Unlike the other approaches, in this method, a power model is built for the whole data centre. To obtain the total energy consumption of the data centre, this method does not need to calculate the energy consumption of each server, and is therefore suitable for testing and evaluation of the overall energy consumption of the system. Specifically, in [13], to measure the overall power consumption of the Hadoop platform, the authors used 12 parameters that directly reflected the energy used by the system.

The second approach is server energy consumption modelling. A data centre is composed of numerous servers that provide different services to users. If we can accurately build a power model of one server using this modelling method, we can then obtain the total energy consumption of all the servers. This modelling method is not as straightforward as the first approach. In addition, when the servers used by the data centre are heterogeneous, the energy consumption prediction error of this modelling method is large. More in detail, in [14], to improve the prediction accuracy of the energy consumption model, the authors proposed power models based on three deep learning methods: an Elman neural network, a BP neural network, and an LSTM neural network. To maximise the energy savings and minimise the consumption costs, the authors of [15]presented a power model based on the ENN strategy for cloud servers. Their experimental results suggested that the prediction accuracy was improved. In [16], the authors developed an energy consumption estimation model for computers in a data centre based on three parameters: page faults, memory used, and processor time. To further improve the energy consumption of the servers, the authors of [17] put forward several energy consumption models based on energy-related parameter selection and workload types. Experiments showed that their model surpassed alternative approaches in terms of prediction accuracy. To measure energy consumption, the authors of [18] created a power model based on the source-code structure. In [19], it was shown that the power consumption of a server has no intrinsic connection with CPU usage.

The third type of approach is VM energy consumption modelling. In general, a server in a data centre can run several VMs. By monitoring the resource utilisation and energy consumption of these VMs, the energy consumption of the server can be obtained. For instance, in [20], the authors introduced a power model based on the relationship between the resource utilisation of VMs and the energy consumption of the server, and their experimental results indicated that their approach outperformed other models in terms of relation error. To evaluate the energy consumption of servers, the author of [21] created a novel power model based on a performance monitor counter. More recently, the authors of [22] introduced a microservice placement strategy for edge-cloud collaborative smart manufacturing. Their approach tackled the solutions over semiconductor manufacturing case study and elaborated the construction of the latency metric. Although their work was promising, it *did not include an automatic analysis of cloud data which leads by machine learning methods that this paper is addressing them.*

III. IECL:An Intelligent Energy Consumption Model Based on Machine Learning

In this section, we will introduce an intelligent energy consumption model called IECL, based on a combination of the SVM, RF, and GS algorithms. It consists of six main steps: modelling the flow of energy consumption (Section III-A), data sampling (Section III-B), feature extraction (Section III-C), feature selection (Section III-D), feature analysis (Section III-E), and construction of the energy consumption model (Section III-F).

A. Modelling the flow of energy consumption

Fig. 1 illustrates the process flow for server power modelling. It consists of five steps. Data sampling is the first step in power modelling, and involves collecting related data from the cloud resources and applications. In the feature extraction step, the features related to energy consumption are identified, while in the feature selection step, a subset of features are chosen that are related to energy consumption. Model building and training are the final steps in developing the energy consumption model. In this paper, the server is leveraged to establish the energy consumption model and conduct training.

After training, the power model is evaluated to verify its robustness and effectiveness. Finally, our energy consumption model is compared with alternative approaches.

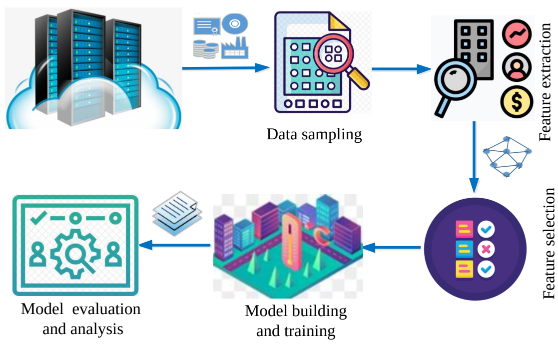


Fig. 1: Process flow of our energy consumption model

B. Data sampling

Data sampling is a prerequisite for the construction of an accurate energy consumption model. Fig. 2 illustrates the process of data sampling. The main components used are a power supply, a power meter, a test server, and a data centre manager (recording equipment). Fig. 2 shows that the power meter is connected to the testing server and the data centre manager (recording equipment). The power meter is used to record the power consumption data from the test server. The data centre manager (recording equipment) also collects parameter data. To obtain the working status of the server and collect experimental data in real time, it is essential to monitor and manage the performance indicators of the server. At present, the most commonly used monitoring software applications include Ganglia (http://ganglia.sourceforge.net/), Zabbix (https://www.zabbix.com/) and Nagios.

This paper presents a joint monitoring approach based on Ganglia and Zabbix. Ganglia is used to monitor the basic performance metrics of the server, while Zabbix is used to support secondary custom development. This approach effectively combines the strengths of each software, allowing us to jointly monitor more energy-related indicators and reduce the overall system overhead.

C. Feature extraction

In this section, we describe the feature extraction step. In the following sections, we introduce the feature selection and feature analysis processes. Feature analysis allows us to understand the relationship between the type of task and the usage of the sub-components of the server, such as the processor and disk. The feature extraction step is carried out to identify features that are related to power modelling. For instance, the disk time represents the percentage of disk time spent on input/output operations. All of the characteristic parameters can be captured by deploying Ganglia and Zabbix software on the server. When feature extraction is complete, we need to select a subset of suitable features.

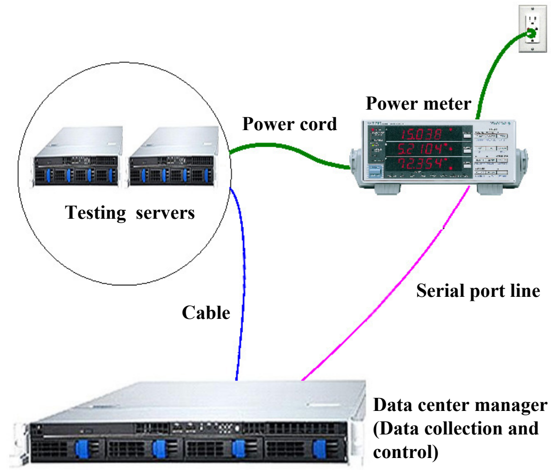


Fig. 2: Equipment used in the data sampling process

In the paper, a total of 29 features are extracted and used to establish the energy consumption model. Table I lists the parameters and characteristics used in our model. Tasks can be classified as CPU-intensive, I/O-intensive, or web transaction workloads.

D. Feature selection

Some of the features are associated with energy consumption, while others are not. We therefore need to choose a suitable set of features to build our energy consumption model. Since the RF algorithm has excellent overall classification performance and strong generalisability, it is suitable for feature parameter screening [23], [24]. We therefore adopt this approach to screen the feature parameters. A schematic diagram of the RF algorithm is shown in Fig. 3.

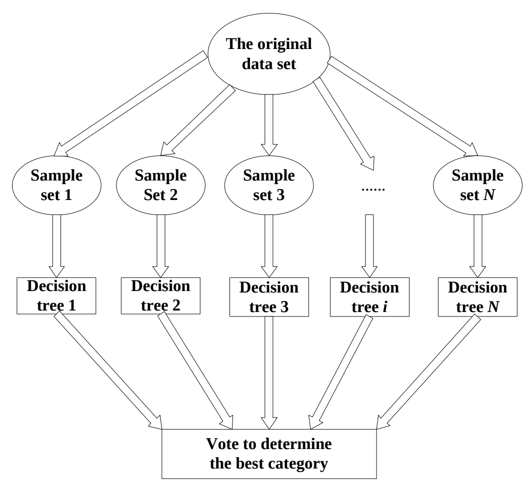


Fig. 3: Schematic diagram of the RF algorithm

The steps used by the RF algorithm to calculate the importance of each feature can be summarised as follows:
i) i) For each decision tree, we choose corresponding out-of-bag data (some remaining samples that have not been extracted) to calculate the out-of-bag data error, denoted as *error* 1.

TABLE I: Parameter name and characteristics

No	Parameter name	Software	Meaning
1	CPU user time	Zabbix	CPU usage percentage of user space
2	CPU idle time	Zabbix	Ratio of idle CPU time
3	Context switches/sec	Zabbix	Switching between processes or threads per second
4	Page faults/sec	Zabbix	Rate of page errors caused by threads executing in a process
5	bytes_out	Ganglia	Number of bytes per second going out
6	pkts_out	Ganglia	Packets going out per second
7	bytes_in	Ganglia	Number of bytes for coming in every second
8	pkts_in	Ganglia	Packets coming in per second
9	CPU softirq time	Zabbix	CPU percentage consumed by software interrupts
10	avgrq-sz	Ganglia	Average data size per input and output operation
11	I/O Data Bytes/sec	Ganglia	Input and output bytes for every second
12	CPU system time	Zabbix	CPU percentage used for kernel space
13	mem_free	Ganglia	Free memory size
14	Processor load(avg1)	Zabbix	Average system workload every one minutes
15	mem_cached	Ganglia	Cache memory size
16	cpu_idle	Zabbix	Percentage of idle CPU time after startup
17	disk_free	Ganglia	Free disk space
18	Disk Time	Zabbix	Percentage of time the disk is used for input and output operations
19	I/O Data Operation/sec	Ganglia	Number of input and output operations every second
20	svctm	Ganglia	I/O average service time
21	Processor load(avg5)	Zabbix	Average system workload every five minutes
22	await	Ganglia	Mean waiting time for input and output
23	avgqu-sz	Ganglia	Average I/O queue length
24	Processor load(avg15)	Zabbix	Average system workload every fifteen minutes
25	CPU iowait time	Zabbix	Maximum ratio of idle CPU I/O requests
26	proc.num	Ganglia	Total number of processes
27	part_max_used	Ganglia	Maximum ratio leveraged by all partitions
28	proc_run.num	Ganglia	Total number of processes running
29	cpu_nice	Zabbix	Proportion of processes whose priorities changed in the user process space

ii) ii) We randomly add interference to feature X of all samples of out-of-bag data, and calculate the out-of-bag data error again, denoted here as *error* 2.
iii) If there are N trees in the forest, the importance of feature $X = \sum(error2 - error1)/N$.
iiii) Finally, features with high importance are selected as the new dataset.

To choose a subset of features, we deploy three types of applications (CPU-intensive, web transaction and I/O-intensive applications) on a Dell server. The parameters of the server are shown in Table II. Table III shows the workload benchmark [17] used for each type of application.

TABLE II: Server parameters

Name	Value
CPU frequency	Intel Core i3-7100 3.9 GHZ
Mainboard	ASUS H110M-D3V
Memory size	8GB
Disk size	1TB
Network Interface Card (NIC)	RTL8168 Gigabit Enthernet controller

TABLE III: Workload benchmarks

Workload types	Benchmarks
CPU-intensive workloads	SPEC_CPU2006
I/O-intensive workloads	Iozone
Web transaction workloads	Loader Runner 11

After applying the RF algorithm and deploying the applications on the server, the importance ratio of each feature is obtained (see Table IV). If the importance ratio is lower than 1%, we assume the contribution of this feature to the server energy consumption can be ignored. Hence, 1% is chosen as the boundary value. By analysing the data in Table IV and selecting features with an importance ratio higher than 1%, we obtain the following results: for CPU intensive tasks, a total of 27 features are selected to build the power model; for I/O-intensive tasks, a total of 26 features are chosen; and for web transaction tasks, a total of 26 features are selected.

E. Feature analysis

Feature analysis allows us to understand the relationship between the task type and the power consumption of the sub-components of the server. Figs. 4-6 illustrate the usage of three server sub-components (the processor, memory, and disk) for three different types of tasks. Fig. 4 shows the utilisation of the CPU, I/O, and memory over time for a CPU-intensive task. It can be seen that the CPU utilisation remains relatively stable at 10–35%, the memory utilisation is relatively stable at 18–40%, and the disk utilisation increases repeatedly and gradually between 0–20%, 0–40%, and 0–100%, and then decreases to 0% and remains relatively stable. Fig. 4 also shows that the utilisation of the sub-components of the server (the processor, memory, and disk) varies over time, with no fixed pattern.

Similarly, Fig. 5 shows the CPU, memory and I/O usage over time for I/O-intensive tasks. It can be seen that the CPU usage is stable at 5–27%, the memory usage is stable at 40–50%, and the disk usage is stable at 54–100%. The figure also shows that the usage of the server sub-components varies over time.

Fig. 6 shows the CPU, memory, and I/O usage over time for Web transaction tasks. The CPU usage remains stable when it increases from 10% to 100%, and the memory usage remains stable when it ranges from 42% to 48%. The disk utilisation ranges from 0 to 44%.

Fig. 7 illustrates the server power consumption for different types of tasks. Fig. 7a shows that the server power ranges from

TABLE IV: Importance ratio for characteristic parameter

CPU-intensive task	Importance ratio (%)	I/O-intensive task	Importance ratio (%)	Web transactional task	Importance ratio (%)
CPU user time	7.122	avgqu-sz	6.0789	CPU_nice	7.2684
CPU idle time	6.4263	await	4.8794	CPU system time	6.7709
Context switches/sec	6.4076	I/O Data Bytes/sec	4.8683	Page faults/sec	5.9472
Page faults/sec	6.2208	CPU idle time	4.8597	pkts_in	5.8824
bytes_out	4.9083	bytes_out	4.8498	bytes_out	5.2759
pkts_out	4.8063	CPU iowait time	4.747	pkts_out	5.2399
bytes_in	4.5558	Page faults/sec	4.7417	bytes_in	5.2079
pkts_in	4.5018	CPU user time	4.6673	Context switches/sec	4.8678
CPU softirq time	4.1719	Context switches/sec	4.6551	CPU softirq time	4.3539
avgrq-sz	4.0342	pkts_out	4.649	CPU idle time	4.2793
I/O Data Bytes/sec	3.8358	pkts_in	4.2352	CPU user time	3.9839
CPU system time	3.7393	CPU system time	4.1981	proc_run.num	3.8259
mem_free	3.6801	bytes_in	4.1917	Processor load (avg1)	3.5158
Processor load (avg1)	3.6421	CPU softirq time	3.7109	avgrq-sz	3.4787
mem_cached	3.5938	Processor load (avg1)	3.6127	mem_free	3.4525
CPU_idle	3.1503	svctm	3.5583	mem_cached	3.1276
disk_free	2.9587	mem_free	3.4732	I/O Data Bytes/sec	3.1166
Disk Time	2.9231	Processor load (avg15)	3.2206	proc.num	3.1085
I/O Data Operation/sec	2.8142	Processor load (avg5)	3.1757	Processor load (avg5)	2.863
svctm	2.6496	IO avgrq-sz	3.0295	cpu_idle	2.7754
Processor load (avg5)	2.5151	CPU_idle	2.9944	Processor load (avg15)	2.6258
await	2.4926	mem_cached	2.975	I/O Data Operation/sec	2.0785
avgqu-sz	2.284	IO Data Operation/sec	2.9639	disk_free	1.895
Processor load (avg15)	1.9766	proc.num	1.5298	svctm	1.6994
CPU iowait time	1.597	proc_run.num	1.2876	await	1.3249
proc.num	1.3624	disk_free	1.1567	Disk Time	1.1181
part_max_used	1.1828	part_max_used	0.8763	avgqu-sz	0.654
proc_run.num	0.4474	Disk Time	0.8142	part_max_used	0.2627
CPU_nice	0	CPU_nice	0	CPU iowait time	0

46 to 64 W for CPU-intensive tasks, while Fig. 7b shows that the power range fluctuates continuously between 47 and 59 W for I/O-intensive tasks, with an average power consumption of 48 W. Fig. 7c shows that the range of server power for web transaction tasks is between 46 and 71 W. From Figs. 4-7, we can see that there is a correlation between CPU utilisation and power consumption.

To build an accurate energy consumption model, we not only need to consider the type of task, but also the resource utilisation rate of each component and the rule of change. To solve this problem, we adopt the SVM algorithm to build our energy consumption model, which is introduced in the next section.

F. Energy consumption model

In this section, we describe the SVM and GS methods, which are used to build our energy consumption model. In the following, we discuss these two strategies in detail.

1) SVM: In the last section, we showed that the server energy consumption is influenced by many factors and has nonlinear and uncertain characteristics. The SVM algorithm is a supervised machine learning method that can effectively deal with nonlinear classification and regression problems [25]. We therefore select this algorithm to establish our energy consumption model for a server. To deal with a regression problem, the SVM algorithm uses the data in the training set, and the regression function is as follows:

$$f(x) = \langle w, g(x) \rangle + b \tag{1}$$

where w is the weight vector, x is the input vector, $g(x)$ represents the mapping function, and b is a constant.

The SVM algorithm minimises the sum of the squares of the weight coefficients to ensure a smooth function relation. At the same time, the less error is allowed to improve the generalisation performance of the model. Hence, w and b are determined by solving the following quadratic convex programming problem:

$$\min_{w,b,\xi_i,\xi_i^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{2}$$

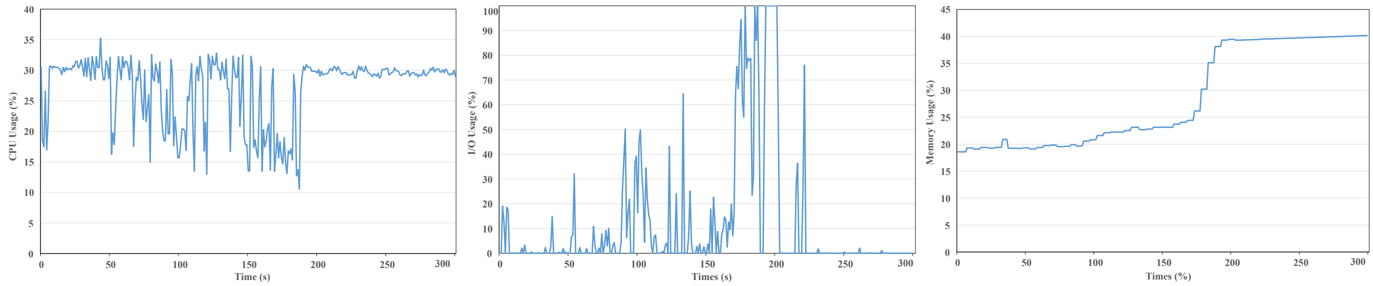
$$s.t. \begin{cases} y_i - \langle w, g(x) \rangle - b \le \varepsilon + \xi_i \\ -y_i + \langle w, g(x) \rangle + b \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0, i = 1, 2 \cdots n \end{cases} \tag{3}$$

where $||w||$ is the description function, and C means the penalty factor, $C > 0$, and ξ_i and ξ_i^* is the relaxation factor, and y_i refers to the output of the i-th sample, and ε is the fitting error.

By using an implicit kernel function rather than $g(x)$, a nonlinear problem can be mapped to a higher dimensional space, and the optimal linearly separable plane can be searched for and solved in this new space. To solve the nonlinear relation between the energy consumption and the impact factor of the data centre, a radial basis function (RBF) is selected as the kernel function in this paper. This is because the RBF is superior to other kernel functions in dealing with nonlinear problems and requires fewer parameters. The RBF equation is defined as follows:

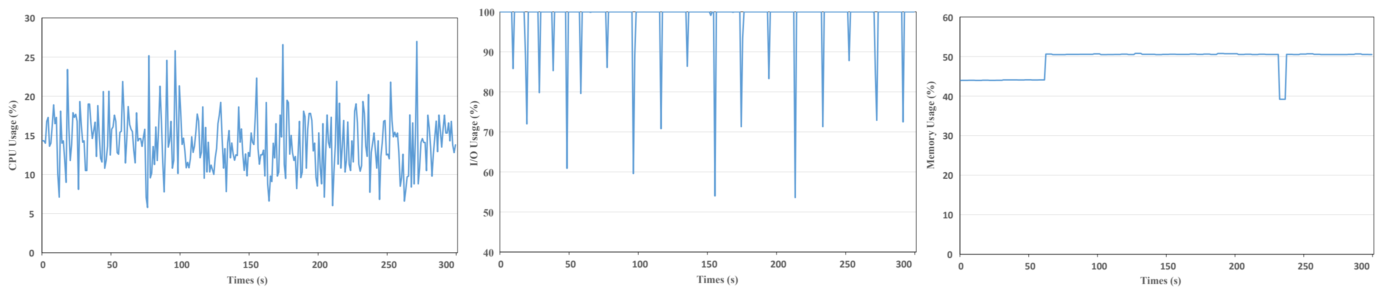
$$k_{RBF}\left(a', a\right) = e^{\frac{\left\|a' - a\right\|^2}{2\sigma^2}} \tag{4}$$

where a' and a are two low-dimensional vectors, and $\frac{1}{2\sigma^2}$



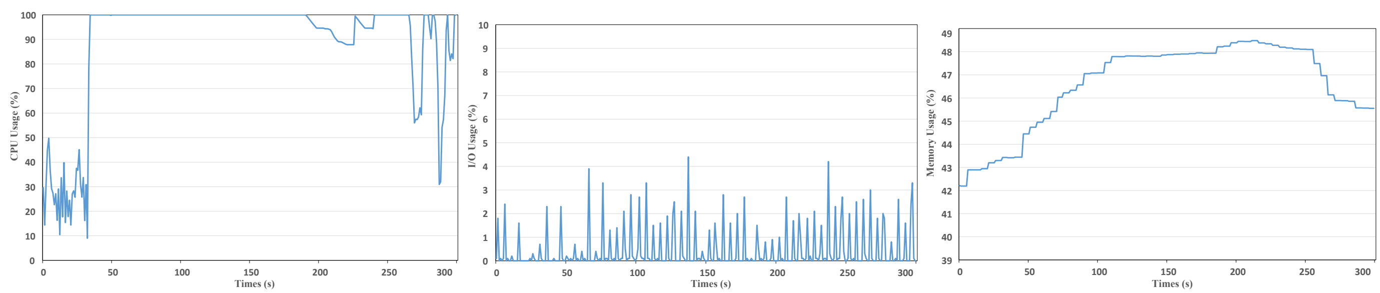
(a) CPU metric (b) I/O metric (c) Memory metric

Fig. 4: Resource utilisation under CPU-intensive workloads



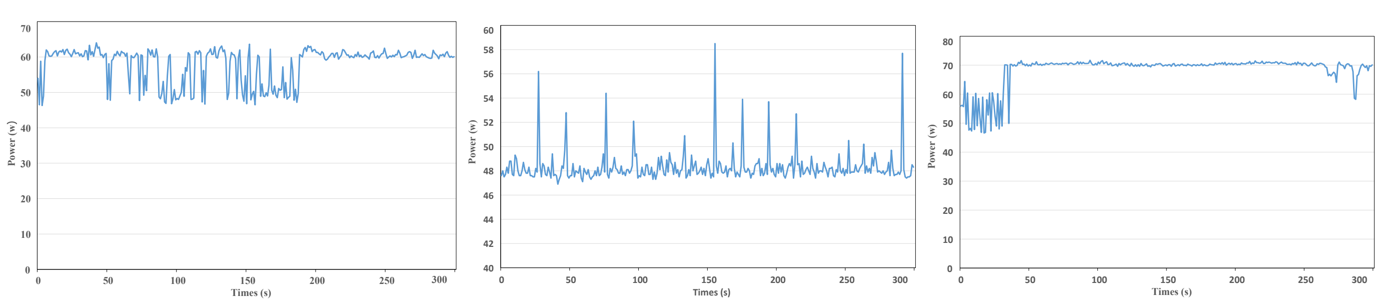
(a) CPU metric (b) I/O metric (c) Memory metric

Fig. 5: Resource utilisation under I/O-intensive workloads



(a) CPU metric (b) I/O metric (c) Memory metric

Fig. 6: Resource utilisation under Web transaction workloads



(a) CPU-intensive tasks (b) I/O-intensive tasks (c) Web transaction tasks

Fig. 7: Energy consumption under different types of workloads

is also known as γ parameter, which reflects the degree of separation of the mapping.

The three parameters (C, ε, and γ) are called model hyper-parameters and are constant. In the process of model training, adjusting model hyper-parameters can change model performance.

2) GS: This is an exhaustive search method for specifying parameter values [26]. It is a learning algorithm that optimises the hyperparameters of the model through cross-validation to obtain the optimal parameter combination. At present, the most commonly used methods for adjusting the hyperparameters include a random, the genetic algorithm (GA), GS, and particle swarm optimisation (PSO). However, GA and PSO take a long time to find an optimised solution, whereas the GS method can search a wide parameter space while controlling the calculated amount [27]. The GS algorithm is therefore selected as the method of adjusting the hyperparameters. The basic principle of GS can be summarised as follows: the parameters to be searched are divided into grids of the same size within a certain space range, and all points in the grid are traversed to find value. The performance of each point in the given interval can then be determined by passing it to the SVM system. The point at which the performance of the entire system is highest is called the optimal parameter. In other words, the performance of the model can be optimised by adjusting the hyperparameters to optimise the model evaluation index.

IV. Performance Evaluation

To measure the performance of our IECL model, we performed a series of experiments, and these are described in detail below.

A. Experimental environment and settings

Our experiments were carried out using a Dell server, with the parameter configuration shown in Table II. Three different types of workloads were used (CPU-intensive, I/O-intensive, and web transaction workloads), as shown in Table III. A dataset containing 3,000 items of data for CPU-intensive, I/O-intensive, and web transaction tasks was collected for the experiment. The steps in the experiment were as follows. First, the collected data were normalised, and the dataset was divided into training and test sets consisting of 80% and 20% of the data, respectively. Following this, the parameters of the training set were searched and modelled. Finally, the test set was used for evaluation, and the performance of the model was confirmed. The SVM package from the SciKit-Learn library was used to construct our energy consumption model. The default parameters of the algorithm package were set, and the default parameters were set as grid reference values, C= 1, γ = 0.036, and ε = 0.1, and the grid was built near the grid reference values [28]. In this paper, an exponential grid with a base of 10 was used for searching, and the parameter settings for each node were drawn from the following sets: $C \in \{0.01, 0.1, 1, 10, 100\}$, $\gamma \in \{0.001, 0.01, 0.1, 1, 10\}$, and $\varepsilon \in \{0.001, 0.01, 0.1, 1, 10\}$. The nodes intersected to form a 5×5×5 grid. The Java implementation of IECL is available in [29].

B. Comparison with alternative algorithms

To evaluate the effectiveness of the IECL model, we used the power regression [17], Cubic [24], CMP model [16], FSDL model [13], and AEC algorithms [25] for comparison. Of these, power regression [17] and CMP [16] are two typical linear regression models. The FSDL model in [13] is a typical supervised machine learning method that can effectively deal with nonlinear classification and regression problems. The cubic [24] and AEC [25] algorithms establish a power model based on the system resource utilisation. In the following section, we evaluate these in detail.

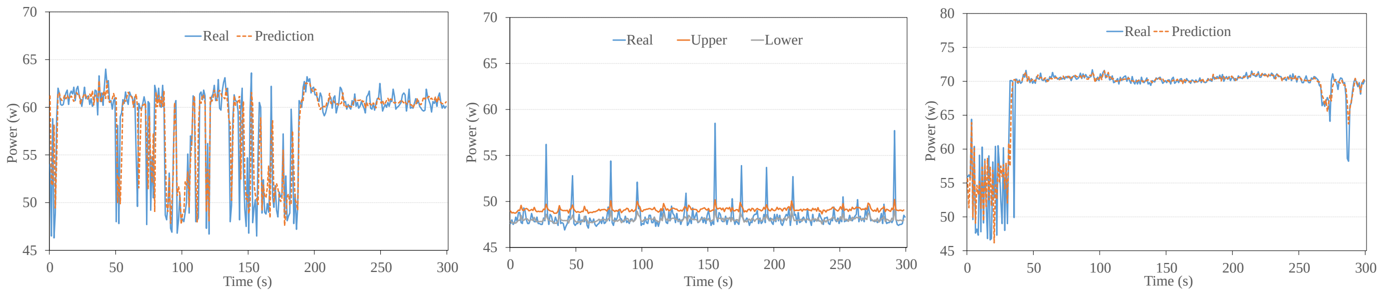
C. Evaluation of the prediction accuracy of the proposed model

Fig. 8 shows a comparison between the real-time energy consumption and the predicted energy consumption of the server under different types of workloads. Web transaction workloads require more power consumption than CPU-intensive and I/O-intensive workloads. This is because the processor consumes more power than the memory and I/O components. Fig. 8-9 show the prediction errors of the IECL model, using as evaluation criteria the MAPE (mean absolute error percentage) and RMSE (root mean square error). The data in Table IV show that for CPU-intensive workloads, the MAPE is 3.104% and the RMSE is 3.05. For I/O-intensive workloads, the MAPE is 1.793% and the RMSE is 1.18. For web transaction workloads, the MAPE is 1.426% and the RMSE is 2.195. Fig. 9 shows the predicted interval for the IECL model. Based on the results from Fig. 8-9, it can be concluded that our IECL model yields excellent performance and that there is little difference in the test results under different workloads, which can provide a reference for the prediction and optimisation of energy consumption for server operation.

D. Comparison with other baseline models

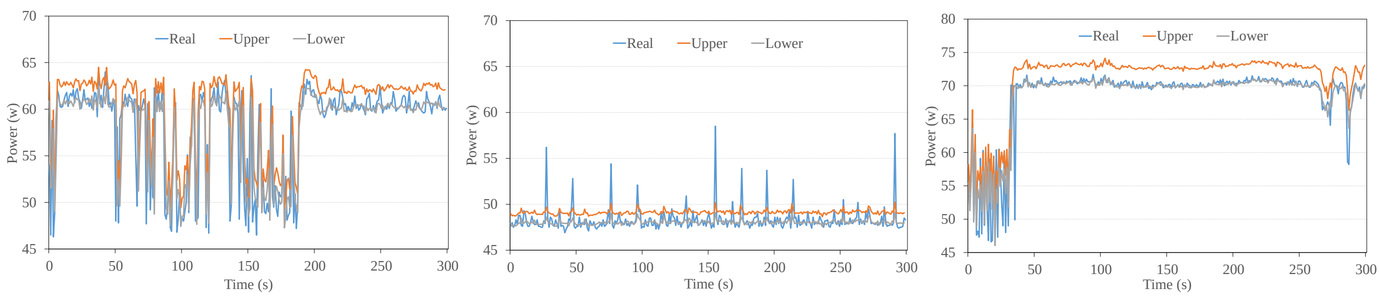
To evaluate the effectiveness of IECL, we compared it with other baseline power models as described above. The results are shown in Fig. 10-11. It can be seen that compared with the five alternative approaches, our IECL model achieves the highest prediction accuracy in most cases, and the average MAPE and RMSE are reduced by 2.9% and 2.7%, respectively. This is because IECL selects more features, uses the RF algorithm to filter features, and applies GS and SVM to conduct modelling, which improves the accuracy.

In terms of the MAPE and RMSE, the FSDL model is better than the other four power models (the power regression, CMP, AEC, and cubic algorithms). This is because it adopts a deep learning method to construct the power consumption model based on the Hadoop platform and to estimate its total power consumption. The power regression model is better than the other three approaches (the CMP, AEC, and cubic models). This is because power regression model considers the characteristics of the application and the power component (such as the processing unit, memory, and disk). Compared to the AEC and cubic models, CMP gives better performance, since it



(a) CPU-intensive loads (b) I/O-intensive loads (c) Web transaction loads

Fig. 8: Power consumption prediction under different types of workloads



(a) CPU-intensive loads (b) I/O-intensive loads (c) Web Transnational loads

Fig. 9: Power consumption prediction interval under different types of workloads

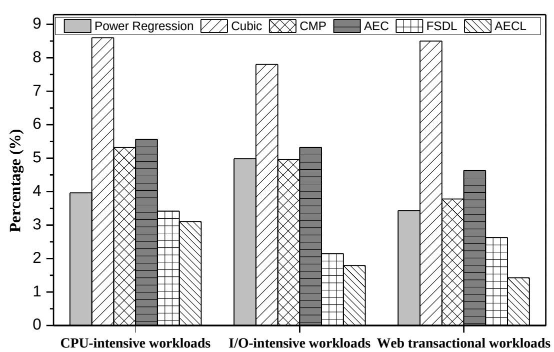


Fig. 10: MAPE results for each energy consumption model

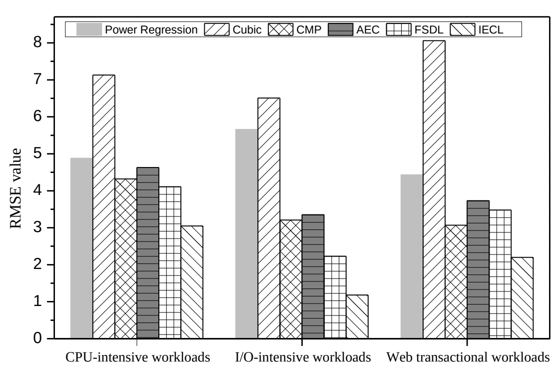


Fig. 11: RMSE results for each energy consumption model

considers the three main energy consumption parameters when constructing the power model. The cubic model generates the worst performance, since it only takes into account the processing unit, and is not suitable for varying workloads. Based on the results shown in Fig. 10 and 11, we can conclude that the IECL model achieves better performance than the other benchmark models.

The IECL model can be used in cloud data centres and can provide theoretical and practical guidance for cloud manufacturing. Cloud manufacturing enterprise systems are mutually bound to industrial information systems and smart modelling could automatically integrate this in deploying frequently used applications. The IECL model can predict the server energy consumption in real time, and can estimate the trends in this energy consumption. It is also better able to evaluate the advantages and disadvantages of the energy-aware algorithm, and hence can contribute to the optimisation of energy consumption for cloud manufacturing.

V. Conclusions and Directions for Future Research

Due to the nonlinear and uncertain characteristics of server energy consumption, achieving real-time evaluation of energy consumption is becoming increasingly difficult. This paper proposes a new server energy consumption model called IECL, based on a combination of the RF, SVM, and GS algorithms for different types of IoT tasks. Our experimental results suggest that the IECL model can predict server energy consumption with an accuracy of about 97%.

The proposed energy consumption model can be used in cloud data centers cloud data centres, and can provide theoretical and practical guidance for cloud manufacturing. Although our model is promising, it does not consider the energy consumption and time costs of training, and this is an important direction for future research on energy consumption models.

ACKNOWLEDGMENT

This work was supported by supported by scientific research project of education department of hunan province (Nos.21B0779), and in part by the Hunan Province Key Laboratory of Industrial Internet Technology and Security (nos. 2019TP1011). Mohammad Shojafar is supported by Marie Curie Global Fellowship funded by European Commission with grant agreement MSCA-IF-GF-839255.

REFERENCES

[1] J. Tao, "Research on sdn-based dynamic management of energy consumption of cloud data center," in *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, vol. 4. IEEE, 2021, pp. 1266–1270.

[2] C. Chen, K. Li, W. Wei, J. T. Zhou, and Z. Zeng, "Hierarchical graph neural networks for few-shot learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 1, pp. 240–252, 2021.

[3] X. Chang, W. Li, and A. Y. Zomaya, "A lightweight short-term photovoltaic power prediction for edge computing," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 4, pp. 946–955, 2020.

[4] K. Cao, S. Hu, Y. Shi, A. Colombo, S. Karnouskos, and X. Li, "A survey on edge and edge-cloud computing assisted cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7806–7819, 2021.

[5] 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, pp. 1–18, 2017.

[6] Z. Zhou, M. Shojafar, M. Alazab, J. Abawajy, and F. Li, "Afed-ef: An energy-efficient vm allocation algorithm for iot applications in a cloud data center," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 2, pp. 658–669, 2021.

[7] A. Israr, Q. Yang, W. Li, and A. Y. Zomaya, "Renewable energy powered sustainable 5g network infrastructure: Opportunities, challenges and perspectives," *Journal of Network and Computer Applications*, vol. 175, p. 102910, 2021.

[8] L. Bing, F. Zhu, J. Zhang, J. Chen, C. Xing, N. N. Xiong, and J. L. Mauri, "A time-driven data placement strategy for a scientific workflow combining edge computing and cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4254–4265, 2019.

[9] C. Chen, K. Li, S. G. Teo, X. Zou, K. Li, and Z. Zeng, "Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 14, no. 4, pp. 1–23, 2020.

[10] T. Baker, B. Aldawsari, M. Asim, H. Tawfik, Z. Maamar, and R. Buyya, "Cloud-senergy: A bin-packing based multi-cloud service broker for energy efficient composition and execution of data-intensive applications," *Sustainable Computing: informatics and systems*, vol. 19, pp. 242–252, 2018.

[11] M. Abdel-Basset, R. Mohamed, M. Elhoseny, A. K. Bashir, A. Jolfaei, and N. Kumar, "Energy-aware marine predators algorithm for task scheduling in iot-based fog computing applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5068–5076, 2020.

[12] Y. Wang *et al.*, "Industedge: A time-sensitive networking enabled edge-cloud collaborative intelligent platform for smart industry," *IEEE Transactions on Industrial Informatics*, 2021.

[13] Y. Liang, Z. Hu, and K. Li, "Power consumption model based on feature selection and deep learning in cloud computing scenarios," *IET Communications*, vol. 14, no. 10, pp. 1610–1618, 2020.

[14] W. Lin, G. Wu, X. Wang, and K. Li, "An artificial neural network approach to power consumption model construction for servers in cloud data centers," *IEEE Transactions on Sustainable Computing*, vol. 5, no. 3, pp. 329 – 340, 2020.

[15] W. Wu, W. Lin, L. He, G. Wu, and C. H. Hsu, "A power consumption model for cloud servers based on elman neural network," *IEEE Transactions on Cloud Computing*, vol. 99, pp. 1–12, 2021.

[16] Y. Junyang, H. Zhigang, Z. Zhou, and Y. Liu, "A cmp energy consumption estimate model for computer systems," *Journal of University of Electronic Science and Technology of China*, vol. 44, pp. 422–427, 2015.

[17] Z. Zhou, J. H. Abawajy, F. Li, Z. Hu, M. U. Chowdhury, A. Alelaiwi, and K. Li, "Fine-grained energy consumption model of servers based on task characteristics in cloud data center," *IEEE Access*, vol. 6, no. 1, pp. 27 080–27 090, 2018.

[18] H. Liu, F. Yan, S. Zhang, T. Xiao, and J. Song, "Source-level energy consumption estimation for cloud computing tasks," *IEEE Access*, vol. 6, pp. 1321–1330, 2017.

[19] X. Zhang, J.-J. Lu, X. Qin, and X.-N. Zhao, "A high-level energy consumption model for heterogeneous data centers," *Simulation Modelling Practice and Theory*, vol. 39, pp. 41–55, 2013.

[20] S. Chinprasertsuk and S. Gertphol, "Power model for virtual machine in cloud computing," in *2014 11th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2014, pp. 140–145.

[21] X. Peng, Z. Hu, D. Liu, G. Yan, and X. Qu, "Virtual machine power measuring technique with bounded error in cloud environments," *Journal of Network and Computer Applications*, vol. 36, no. 2, p. 818–828, 2013.

[22] Y. Wang *et al.*, "Mpcsm: Microservice placement for edge-cloud collaborative smart manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 5898–5908, 2020.

[23] S. Lakshmanaprabu *et al.*, "Random forest for big data classification in the internet of things using optimal features," *International journal of machine learning and cybernetics*, vol. 10, no. 10, pp. 2609–2618, 2019.

[24] Y. Chen, W. Zheng, W. Li, and Y. Huang, "Large group activity security risk assessment and risk early warning based on random forest algorithm," *Pattern Recognition Letters*, vol. 144, pp. 1–5, 2021.

[25] S. Kavitha, S. Varuna, and R. Ramya, "A comparative analysis on linear regression and support vector regression," in *2016 online international conference on green engineering and technologies (IC-GET)*. IEEE, 2016, pp. 1–5.

[26] Y. Sun, S. Ding, Z. Zhang, and W. Jia, "An improved grid search algorithm to optimize svr for prediction," *Soft Computing*, vol. 25, no. 7, pp. 5633–5644, 2021.

[27] S. Cui, Y.-W. Wang, and J.-W. Xiao, "Peer-to-peer energy sharing among smart energy buildings by distributed transaction," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6491–6501, 2019.

[28] R. Garreta and G. Moncecchi, "Learning scikit-learn : machine learning in python," *Packt Publishing*, 2013.

[29] 2022, "IECL source code," https://github.com/mshojafar/sourcecodes/raw/master/IECL_Sourcecode.zip.

Zhou Zhou received the Ph.D. degree from Central South University, Changsha, China, in 2017, majoring in computer science. He is currently an Associate professor at Changsha University. Also, he has accepted a post-doctoral position in Hunan University. His research interests include Cloud computing, energy consumption model, energy-efficient resource management, and virtual machine deployment. He is served as a Reviewer for some famous journals, such as IEEE Transactions on services computing, IEEE Transactions on Cloud Computing, IEEE Transactions on Industrial Informatics, Concurrency and Computation: Practice and Experience, Future Generation Computer Systems, Neural Computing and Applications, Journal of Supercomputing, Journal of Cluster Computing, and IEEE Systems Journal.

Mohammad Shojafar (M'17-SM'19) is a Senior Lecturer (Associate professor) is a Senior Lecturer (Associate Professor) in the network security and an Intel Innovator, Professional ACM member and ACM Distinguished Speaker, and a Marie Curie Alumni, working in the 5G & 6G Innovation Centre (5GIC & 6GIC), Institute for Communication Systems (ICS), at the University of Surrey, UK. Before joining 5GIC/6GIC, he was a Senior Researcher and a Marie Curie Fellow in the SPRITZ Security and Privacy Research group at the University of Padua, Italy. Dr. Mohammad is a PI of AutoTrust, a 750k euro 5G secure autonomous vehicular communication project supported by European Space Agency (ESA) in 2021 and was a PI of PRISENODE project, a 275k euro Horizon 2020 Marie Curie global fellowship project in the areas of Fog/Cloud security collaborating at the University of Padua, Italy. He also was a PI on an Italian SDN security and privacy (60k euro) supported by the University of Padua in 2018. He was contributed to some Italian projects in telecommunications like GAUChO, SAMMClouds, and SC2. He received his Ph.D. degree in ICT from Sapienza University of Rome, Rome, Italy, in 2016 with an "Excellent" degree. He is an Associate Editor in *IEEE Transactions on Network and Service Management*, *IEEE Transactions on Consumer Electronics*, *IEEE Systems Journal* and Computer Networks. For additional information: http://mshojafar.com

Mamoun Alazab (SM'15) is an Associate Professor at the College of Engineering, IT and Environment at Charles Darwin University, Australia. He received his PhD degree in Computer Science from the Federation University of Australia, School of Science, Information Technology and Engineering. He is a cyber security researcher and practitioner with industry and academic experience. Alazab's research is multidisciplinary that focuses on cyber security and digital forensics of computer systems with a focus on cybercrime detection and prevention. He has more than 150 research papers in many international journals and conferences. He is a Senior Member of the IEEE. He is the founding chair of the IEEE Northern Territory (NT) Subsection. For more information: https://sites.google.com/view/alazabm

Fangmin Li Fangmin Li received the B.Sc. degree from the Huazhong University of Science and Technology, Wuhan, China, in 1990, the M.Sc. degree from the National University of Defense Technology, Changsha, China, in 1997, and the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2001, all in computer science. He is currently a Professor with the School of Computer Engineering and Applied Mathematics, Changsha University. He has authored several books on embedded systems and over 50 academic papers in edge computing, cloud computing, wireless networks, and also holds ten Chinese patents. His current research interests include cloud/edge computing, wireless communications and networks, energy-efficient resource management, computer systems and architectures, and embedded systems. Dr. Li is a Senior Member of China Computer Federation (CCF), and a Committee Member of the Technical Committee on Sensor Network, CCF.