Load power estimation using a recurrent neural network for the purpose of computer power energy efficiency improvement

Shinichi Kawaguchi
Department of Clinical Engineering
Kanagawa Institute of Technology
Atsugi, Kanagawa, Japan
kawaguchi@cet.kanagawa-it.ac.jp

2021 IEEE Energy Conversion Congress and Exposition (ECCE) | 978-1-7281-5135-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ECCE47101.2021.9595258

***Abstract*— With the aim of improving the efficiency of processor power supplies under light load conditions, a method of controlling power supply efficiency in conjunction with processor load power has been investigated. To rapidly detect the load power, the authors proposed to use processor performance monitoring information to estimate the processor power. In previous studies, it was shown that power estimations via linear regression using performance measurement information enables good results for single application. However, estimation errors increase in combination cases of some applications because of differences in the proper power estimation regression parameters. In this study, a recurrent neural network (RNN) was used to enable nonlinear regression so that the power profiles in which errors occur can be estimated accurately. Herein, this paper shows that a RNN can estimate processor power more accurately using performance measurement information than can be achieved with linear regression estimation.**

Keywords—Neural network, DC-DC converter, POL, Estimation, Efficiency

I. INTRODUCTION

Due to digitalization and digital transfer (DX), energy consumption levels in information communication technology (ICT) systems are increasing sharply. As a result, energy-saving measures and the greening of ICT equipment are becoming significant issues. In ICT devices such as servers, processor power consumption is an outsized part of energy expenditures, so there are constant and ongoing efforts to improve processor power supply efficiency levels. In a normal server system that handles a large amount of data, data processing is performed continuously but there are often times when the processor is waiting data and its load is low for long periods due to frequent input/output (I/O) access [1-2].

However, since a server's processor power supply efficiency tends to be worse under light load conditions than when it is operating at its rated load capacity, much of the total power provided to such systems is wasted when the load is light. Therefore, as described above, it is necessary to further improve efficiency levels during light load periods while continuing to ensure high efficiency regardless of the load factor.

A number of previous studies have examined ways to improve light-load power efficiency by applying interleave number control (phase-shedding) based on the load applied to digitally controlled multi-phase power supplies [3-6]. As shown in Fig. 1, by selecting the most efficient phase configuration based on the load, efficiency can be improved over a wide load range. However, since it is difficult to make the phase control follow sudden processor load fluctuations in actual operations, the authors proposed an adaptive power efficiency control system that achieves rapid adaptive power supply control by directly receiving processor performance measurement information in digital form, and then using it to estimate the power load status [11]-[13]. As shown in Fig. 2, this system is designed to estimate the power consumed from the processing status of components in real-time, provide it as information to power supply control, and then perform control to maximize power supply efficiency. This allows processor efficiency to be improved regardless of the load. As for power estimation methods, previous studies have shown that linear regression models with appropriate parameters can precisely estimate power loads from processor performance information [11]-[13]. However, when various different applications are executed consecutively, power estimation errors tend to occur due to differences in the regression parameters for each application. In response to this issue, this study has confirmed that load power can be estimated correctly by applying a RNN, even in applications where linear regression errors occur. The following sections report on the power estimation RNN used in this study and the verification results obtained.

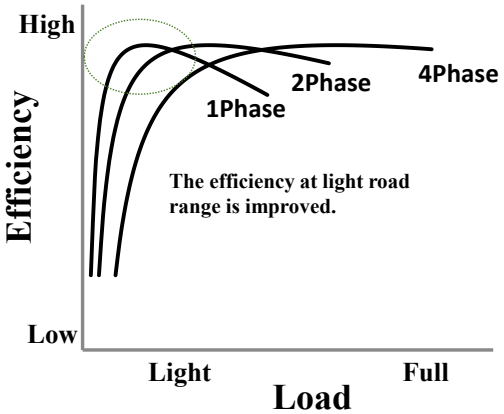

Fig. 1 . Efficiency of multiphase power supply

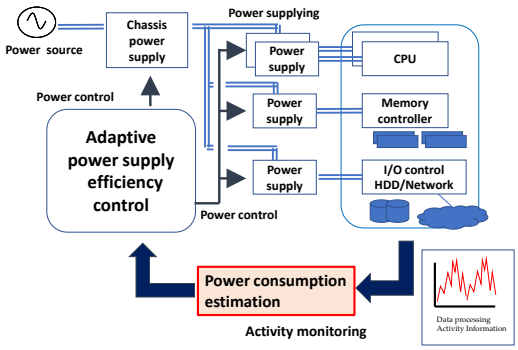

Fig. 2. Adaptive power efficiency control method

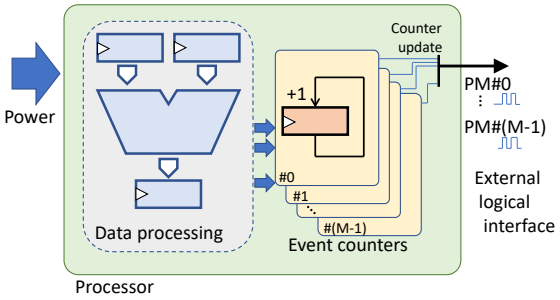

Fig. 3. Hardware event counters in processor

II. Processor Power Estimation Using RNN

A. Processor power and internal hardware event counters

The power consumption of the processor is proportional to the square of the operating voltage, the operating frequency, the internal operating rate, and the load capacity as (1) shows. In particular, reducing the operating voltage has a great effect on reducing power consumption, so the core voltage of the latest processor products has been lowered to about 1.0V. As shown in (1), the fluctuation of the power consumption load is mainly due to the processing activity rate α. This metric depends on what kind of data processing is being performed inside the processor and how often. Therefore, by monitoring the internal data processing status, it is possible to grasp the power consumption load status of the processor.

$$PW_{cpu} = \alpha f C {V_c}^2 + I_l V_c \tag{1}$$

PW_{cpu}: Processor consumption power
α: Processing activity
f: Clock frequency
C: Gate capacitance
V_c: Core voltage
I_l: Leak current

Generally, the processor has a hardware event counter function, and various hardware events (clock, instruction retirement, cache hit / miss, number of branches, etc.) that occur in the processor are counted by the hardware and performance analysis is performed. In general, the event counter data is used to analyze and tune software behavior on certain types of processors. Since the internal data processing status can be measured by the event counter of the processor in this way, the power consumption can be estimated from the operating rate obtained from the event counter as shown in (1).

To date, power estimation using the event counters has been used for the purpose of process scheduling of operating systems and DVFS control of processors. These were estimates with a time scale of a few seconds by collecting event counter information via software internal register access. In addition, the power was estimated by paying attention to the relationship showing the linearity or using stochastic analysis between the event counter and the power consumption[7-10]. On the other hand, in the literature of the precedent, an effort to apply a processor event counter has been made for the purpose of performing dynamic power supply control[11-13]. Furthermore, in order to estimate the power at a sub-microsecond time scale that enables power control, an external interface of processor that outputs updated information of the processor event counter was used.

As shown in Fig. 3, some processors have a function to output the status update of the event counter built in the processor. By incorporating this output information, it is possible to estimate the power at a time scale that can be applied to power supply control. In this case as well, a linear regression equation for power consumption was defined on the premise of the linearity between the event counter and power consumption[11-13]. An example of a linear regression equation of power estimation is (2).

$$P_{prediction}(t_k + \delta) = \sum_{i}^{n} a_i \times CNT_i(t_k) + b \tag{2}$$

$P_{prediction}(t_k + \delta)$ Power prediction at time $t_k + \delta$
$CNT_i(t_k)$ Value stored in performance counter i at time t_k
a_i Coefficient of performance counter i
b Constant

As shown in (1), theoretically, the magnitude of power consumption in a processor has a linear relation with the operating rate of the processor. In the case that high completeness of the processor arithmetic circuit covered by the performance measurement circuit can be realized and the optimum regression parameters for each application can be set, highly accurate power estimation can be performed by linear regression. On the other hand, a general processor has various arithmetic and control functions to execute different types of applications, and there are coverage limits on the implementation of the built-in performance measurement circuit. As a result, Unless the optimum parameters are adjusted for each application to be executed, the linearity between the performance measurement circuit output and the power load cannot be obtained. In such a case, if the power is estimated by linear regression based on the performance information measured internally by the processor, there is a problem that the estimation accuracy degrade. Fig. 4 shows power estimation using linear regression when several application are executed without fine parameter adjustment. As Fig. 4 shows, there is a deviation between the measured

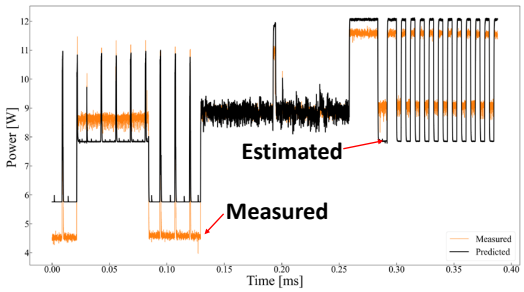

Fig. 4. Power estimation using linear regression (RMSE 12.5%)

power and the estimated power depending on the application being executed.

The deterioration of estimation accuracy suppresses improvements which load-linked power supply efficiency control brings. As a countermeasure to this problem, we propose the processor power load estimation by non-linear regression using the recurrent neural network shown next subsection.

B. Power estimation using RNN

As stated above, this study examines the use of a RNN as a processor load power estimation method in order to predict power profiles that cannot be precisely reproduced by linear regression. The input data is the event information obtained from the performance measurement circuit in the processor which are digital signals of PM#0 to PM#(M-1) received from processor as Fig. 3 shows. Fig. 5 shows an outline of our proposed RNN power estimation circuit. Multiple performance measurement items are performed in the processor, and the event occurrence for each performance measurement item is reported by a dedicated digital signal from the processor. The processor shown in Fig. 3 is equipped with M items performance measurement circuits, and each output signal (PM#0 to PM#M-1) is input to the power estimation circuit.

As can be seen in Fig. 5, each input performance measurement signal is averaged by the exponential moving averaging (EMA) process of the digital filter as preprocessing and then input to the subsequent neural network. Here, it should be noted that α_0 to α_{M-1}, which are the smoothing factors of the EMA, are constant values. EMA calculation is done as (3) and it is described by the recursive equation of (4). Then the EMA process can be executed with IIR digital filter using (4). The power estimation from the RNN is observed to have some unstable fluctuation without the EMA filter. This EMA performs to remove the oscillation noise of power estimation output.

$$\overline{CNT_i}(t_k) = \frac{\sum_{m=0}^{\infty}(1-\alpha_i)^m PM_i(t_k - m)}{\sum_{m=0}^{\infty}(1-\alpha_i)^m} \tag{3}$$

$$\overline{CNT_i}(t_k) = \alpha_i PM_i(t_k) + (1-\alpha_i)\overline{CNT_i}(t_{k-1}) \tag{4}$$

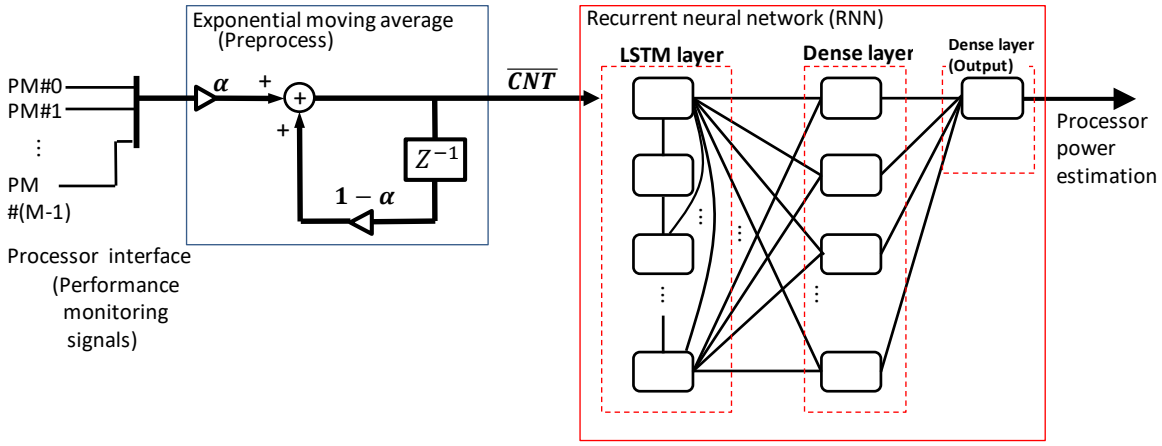

Fig. 5. Power estimation RNN

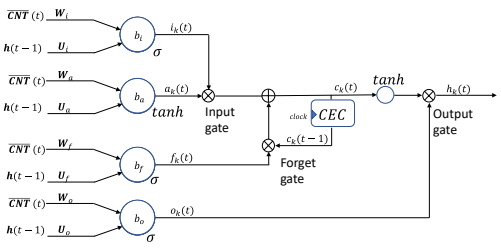

Fig. 6. LSTM cell

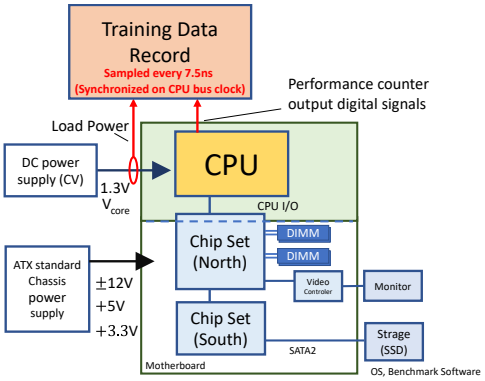

Fig. 7. Experimental computer configuration

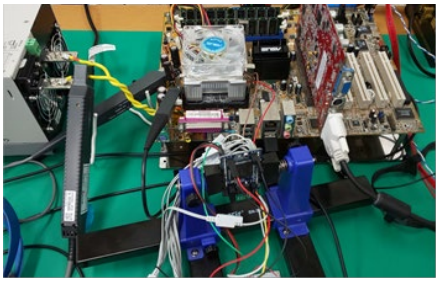

Fig. 8. Experimental computer

$$i_k(t) = \sigma(\overline{\boldsymbol{CNT}}(\mathrm{t})\boldsymbol{W}_{\mathrm{i}} + \boldsymbol{h}(\mathrm{t}-1)\boldsymbol{U}_{\mathrm{i}} + \mathrm{b_i}) \quad (5)$$

$$f_k(t) = \sigma(\overline{\boldsymbol{CNT}}(t)\boldsymbol{W}_f + \boldsymbol{h}(t-1)\boldsymbol{U}_f + b_f) \quad (6)$$

$$a_k(t) = \tanh(\overline{\boldsymbol{CNT}}(t)\boldsymbol{W}_g + \boldsymbol{h}(t-1)\boldsymbol{U}_g + b_g) \quad (7)$$

$$o_k(t) = \sigma(\overline{\boldsymbol{CNT}}(t)\boldsymbol{W}_o + \boldsymbol{h}(t-1)\boldsymbol{U}_o + b_o) \quad (8)$$

$$c_k(t) = f_k(t)c_k(t-1) + a_k(t)i_k(t) \quad (9)$$

$$h(\mathrm{t}) = \mathrm{o}_k(t)\tanh(c_k(t)) \quad (10)$$

Since the performance measurement information from the processor is time-series data, the long-short-term memory (LSTM) layer is placed in the hidden layer of the neural network. Fig. 6 shows a structure of LSTM cell. (5) to (10) show the calculation in cell #k of LSTM. In Fig. 5, it can be seen that there is one LSTM layer, and a dense layer is placed after that layer. The output cell aggregates the calculation data from neurons for each performance measurement and outputs power estimation. Training data was prepared for the power estimation RNN shown in Fig. 5, after which learning was conducted. Both the performance measurement information output by the processor and the processor load power value, which were simultaneously collected, were used for the RNN training data. The training data was loaded into the experimental computer shown in Fig. 7. The environment of this computer was designed to allow the performance measurement information and processor load power value to be collected simultaneously and in synchronization with the external signal interface output (every 7.5 ns) of the processor.

Fig. 8 shows an appearance of the experimental computer on which Pentium®M (1.86GHz) was installed for its CPU. The operating system(OS) was LINUX and Phoronix Test Suite benchmark programs were used as application to create various processing loads [14].

Each benchmark program in Table I was executed as an application program when collecting training data. The selected benchmark programs were classified into processor-, system-, and IO-based execution types. While running the benchmark program on the experimental computer, the configuration registers in the processor are set to operate the event counters in the processor. The performance measurement items are shown in TABLE II. Those items have a high correlation with power consumption are adopted[11][12].

By concatenating the data collected during the execution of each program, 18,000,000 training data patterns were prepared. The learning was performed for 200 epochs for all of the training data. Additionally, 200 steps of time-series data continuously output by the digital filter in the previous stage of the neural network were used as the input into the LSTM layer. The number of each cells of LSTM layer and next dense layer is 10. The RNN was modeled using TensorFlow/Keras framework in another computer [15][16].

TABLE I. EXECUTED BENCHMARK PROGRAMS

Program name	Program description	
	Benchmark type	*operation description*
Pgbench	System bound	Simple TPC-B like benchmark of PostgreSQL (ORDBS: object relational database system)
Ramspeed	Memory bound	System memory (RAM) performance tests
Stresscpu2	Processor bound	Series of GROMACS inner loops hand coded in assembly for speed and efficiency on SSE units.

TABLE II. EVEN ITEMS MONITORED IN COUNTERS

Counter #	*Name*	*Descriptions*
CNT#0	EMON_FUSED_ UOPS_RET	Number of retired fused micro-ops
CNT#1	L2_LINES_OUT	L2 lines allocated

III. RNN POWER ESTIMATION RESULTS

A. Result of power estimation using LSTM

Processor power estimation was performed using the trained RNN modeled by the TensorFlow/Keras. The performance measurement information data output from the processor of the experimental computer shown in Fig. 8 was used as the RNN input. Here, the performance measurement data that had not been included in the training data was used for the estimation.

Fig. 9 shows a comparison between the power estimation results for each program shown in Table I and the load power actually measured on the experimental computer. As can be seen in this figure, the load power was properly replicated. In this case, the root mean square error was 3.3%, which shows that the accuracy was improved from the case seen in Fig. 4.

B. Comparison with result of GRU architecture

Here shows a comparison with the power estimation results using RNN architectures other than LSTM. Power estimation results using simpleRNN and Gated recurrent unit (GRU) are compared. RNN cell structures of all architecture were same as the LSTM. GRU is a kind of recurrent network like LSTM, and is effective in time series data processing. Unlike LSTMs, GRUs have no internal storage cells and have a simpler architecture with fewer parameters. Therefore, the advantage is that the total amount of calculation is small.

The prediction accuracy when using the GRU cell for processor power prediction is compared with the case where LSTM is used. As shown in TABLE III, similar results were obtained with LSTM and GRU. An exact power profile could not be obtained with the simpleRNN and RMS error was the worst as shown in TABLE III.

TABLE III. ESTIMATION ACURACIES USING VARIOUS RNN ARCHITECTURE

Estimation method	*RMS error*
LSTM	3.3%
GRU	3.6%
Linear regression	12.6%
simpleRNN	31.7%

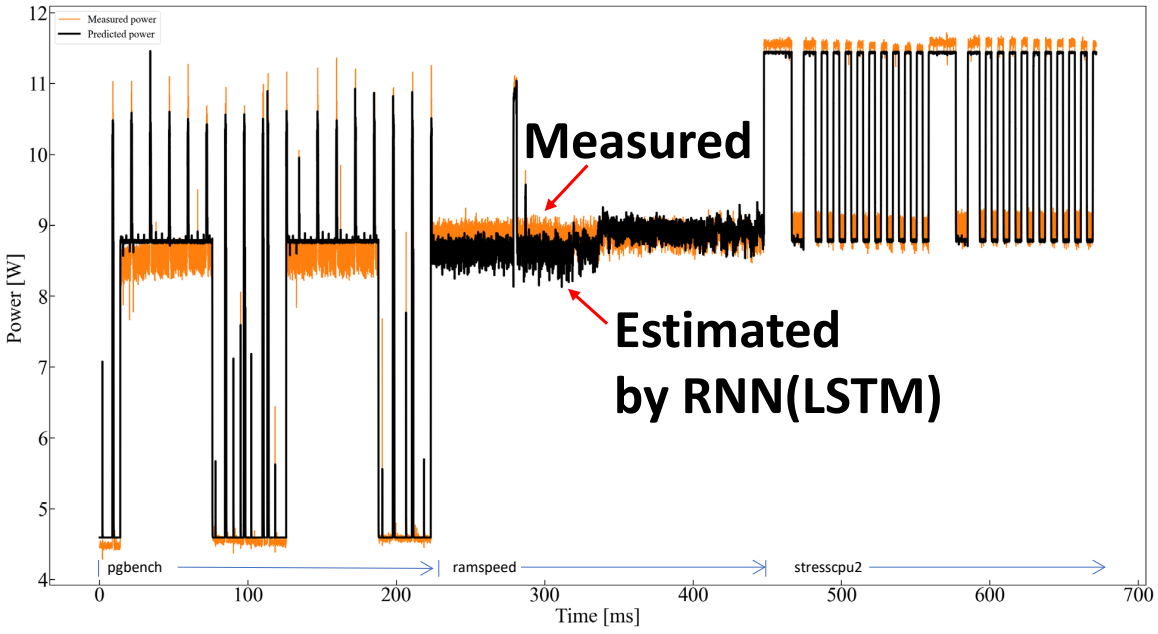

Fig. 9. Estimated power profile using LSTM (RMSE 3.3%)

IV. CONCLUSIONS AND FUTURE WORK

This paper reported on a RNN-based power estimation method for processor power loads that provides a way to improve power efficiency control. The results of experiments conducted using the proposed method showed that power estimation accuracy was significantly improved in contrast to linear regression-based power estimation methods. It was confirmed that equally precise power estimation results were obtained in LSTM and GRU. Therefore, by applying RNN for power estimation, additional efficiency improvements in power supply control can be expected.

As for our future research, we will address hardware implementations of the proposed power estimation RNN method. Also, the power estimation RNN circuit will be incorporated into an experimental computer and integration experiments will be conducted by enabling a synchronous interface between the processor output and the RNN.

REFERENCES

[1] Luiz Andre Barroso and Urs Holzle, The case for energy proportional computing, Computer, 40, pp33-37, Dec. 2007.

[2] J. Koomey, J.Taylor et al., “New data supports finding that 30 percent of servers are comatose, indicating that nearly a third of capital in enterprise data center is wasted,” TSO logic. 2015.

[3] R. Christen, J. Smajic, A. Sridhar and T. Brunschwiler, "Design and optimization of a wide dynamic range Programmable Power Supply for data center applications," 2019 IEEE Applied Power Electronics Conference and Exposition (APEC), Anaheim, CA, USA, 2019, pp. 2210-2217

[4] M. Singh and A. Fayed, "Imbalanced High-Current Multi-Phase Buck Converters for High-Performance CPUs," 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), Dallas, TX, USA, 2019, pp. 929-932.

[5] J. Lin, K. Hu and C. Tsai, "Digital multiphase buck converter with current balance/phase shedding control," TENCON 2015 - 2015 IEEE Region 10 Conference, Macao, 2015, pp. 1-5

[6] Y. Su, K. B. Cheng and W. Wu, "High-efficiency multiphase DC-DC converters for powering processors with turbo mode based on configurable current sharing ratios and intelligent phase management," 2017 IEEE Applied Power Electronics Conference and Exposition (APEC), Tampa, FL, 2017, pp. 191-196

[7] B. P. Singh and B. Thangaraju, "Power, Performance And Thermal Management Using Hardware Performance Counters," 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 2020, pp. 1-7

[8] X. Wu and V. Taylor, "Utilizing Hardware Performance Counters to Model and Optimize the Energy and Performance of Large Scale Scientific Applications on Power-Aware Supercomputers," 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2016, pp. 1180-1189

[9] Y. Kim, S. Park, Y. Cho and N. Chang, "System-Level Online Power Estimation Using an On-Chip Bus Performance Monitoring Unit," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 30, no. 11, pp. 1585-1598, Nov. 2011

[10] Minho Ju, Hyeonggyu Kim and S. Kim, "MofySim: A mobile full-system simulation framework for energy consumption and performance analysis," 2016 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 2016, pp. 245-254

[11] S.Kawaguchi and T.Yachi, “Computer Power Supply Efficiency Improvement by Power Consumption Prediction Procedure Using Performance Counter,” IEICE Transactions on Communications Vol.E97-B, No.2, pp. 408-415, Feb. 2014

[12] S.Kawaguchi and T.Yachi, "Adaptive Power Efficiency Control by Computer Power Consumption Prediction Using Performance Counters," IEEE Trans. on Industry Applications, vol. 52, no. 1, pp. 407-413, Jan.-Feb. 2016

[13] S.Kawaguchi, “Power Consumption Estimation by IIR digital filter for Computer Power Supply Efficiency Improvement”, IEEE Energy Conversion Congress and Exposition – Asia (IEEE ECCE-Asia 2021)

[14] phoronix-test-suites. https://www.phoronix-test-suite.com/ (accessed Jan.6, 2021)

[15] TensorFlow. https://www.tensorflow.org/ (accessed Jan.6, 2021)

[16] Keras https://keras.io/ (accessed Jun 27, 2021)