IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 July 2023, accepted 26 August 2023, date of publication 6 September 2023, date of current version 13 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3312554

RESEARCH ARTICLE

Flexible Carbon Neutralization Strategy: Customized Dynamic Server Management for Energy Efficiency Optimization

SANG-GYUN MA, DONG-GUN LEE, AND YEONG-SEOK SEO, (Member, IEEE)
Department of Computer Engineering, Yeungnam University, Gyeongsan 38541, Republic of Korea

Corresponding author: Yeong-Seok Seo (ysseo@yu.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2023R1A2C1008134).

ABSTRACT In recent years, large data centers have increased significantly as data usage has increased because of digital innovations. However, data centers are 24-hour operation facilities that consume large amounts of power, thus causing environmental problems. Recently, research has been conducted using deep learning methods from various perspectives to predict traffic and reduce power consumption in data centers and servers. However, the traffic processed by servers is highly variable, which is a factor that makes server management difficult. Thus, the traffic processed by servers is irregular, and more research is required on dynamic server management. This study proposes Customized Dynamic Server Management (CDSM) based on Long-Term Short Memory (LSTM), a neural network that is effective in predicting time-series data, to address the aforementioned problem. The proposed method can more effectively save the power used by servers, thereby managing servers more reliably and efficiently than before in the current operating environment. To validate the proposed model, we collected the traffic data at six Wikipedia data centers. We then analyzed the relationship between each traffic data using statistical analysis and conducted experiments. Furthermore, we calculated the server power consumption based on the actual power consumption according to the CPU usage of different servers provided by SPECpower, a benchmark for evaluating server power efficiency. Additionally, we calculated the amount of computation required for the program and deep learning model of the proposed. Based on this, practical results were derived considering the trade-off between the server's power saving and computation performance. The experiment results showed that the server power consumption could be reduced by an average of 68% and a minimum of 32% with CDSM compared to without. This shows that CDSM can effectively reduce server power, hence saving energy in data centers and contributing to carbon neutrality.

INDEX TERMS Data center, deep learning, carbon neutralization, quality of service, server management, traffic prediction.

I. INTRODUCTION

In recent years, large data centers have increased significantly as data usage has increased because of digital innovations. However, owing to the nature of data centers, numerous servers and network devices are operated simultaneously, requiring a significant amount of power, and cooling devices are used to address the problem of heat generation. Moreover, data centers operate round-the-clock, resulting in high power usage [1], [2], [3]. Therefore, the amount of electricity used in data centers has increased, resulting in sharp increase in greenhouse gas emissions [3], [4], [5], [6]. The global trend of achieving carbon neutrality requires undertaking measures to reduce greenhouse gas emissions [7], [8], [9]. Companies are endeavoring to pursue the interests of society from the perspective of Environmental Social Governance (ESG) and Corporate Social Responsibility (CSR) [10], [11], [12].

The associate editor coordinating the review of this manuscript and approving it for publication was Abderrahmane Lakas.

Carbon neutrality is the concept of making net emissions zero by reducing greenhouse gas emissions caused by human activities as much as possible and increasing carbon absorption and reproducing oxygen through forests for the already emitted greenhouse gases. Therefore, it is essential to reduce the power consumption of data centers to achieve carbon neutrality [13].

Improper management of servers in the data center can cause a waste of power as there are more servers turned on than needed when the traffic is low. This is why some servers are turned off during low-traffic hours and turned back on during high-traffic hours. However, if there is a sudden surge in traffic during off-peak hours, the servers may fail to respond and may go down [14], [15]. As such, the traffic handled by the server is irregular, and hence, more research is required on Dynamic Server Management.

Various studies have been conducted in the past to reduce the electricity consumption of servers in data centers. Various methods have been studied to optimize server power consumption, such as concentrating the load on some servers and turning off the rest of the servers depending on the situation of the system or analyzing the power consumption trend and selecting the server with the smallest power consumption and turning it off [16], [17], [18].

However, these methods are inefficient because they do not dynamically control the servers and generate a lot of heat from a large number of servers. They are also unable to respond quickly to traffic overloads caused by rapid fluctuations in traffic. Therefore, this study proposes Customized Dynamic Server Management (CDSM) using a deep learning-based traffic prediction model for power saving and server power control in data centers. CDSM is a method that reduces power consumption by dynamically managing servers for a specific data center. Traffic is suggested through CDSM considering the CPU usage and the scope of guaranteeing the quality of service (QoS) for network communication. The traffic suggested by CDSM is used to control the server and it is varied depending on the server situation. The proposed method can transfer the work of low-traffic servers to other servers, and then power off low-traffic servers. By controlling the power of the servers with the proposed method, the server power consumption can be reduced more effectively, and this will contribute to carbon neutrality by saving the energy used in the data center.

We implement the proposed method using a Long-Term Short Memory (LSTM)-based model that is robust to time-series data prediction [19], [20], [21], [22]. For training and performance evaluation, we collect the transmission and reception data traffic from six data centers of Wikipedia and examined the correlation between the data via statistical analysis to select input variables.

CDSM provides the maximum traffic throughput that a server can reliably handle based on the traffic predicted by a deep learning-based traffic prediction model. This is determined by considering that the QoS is ensured while using energy efficiently when the threshold of CPU usage is 70% [23], [24]. This study suggests the server free resource as an evaluation metric to validate the CDSM. In addition, we use statistical methods to evaluate the consistency of the model's predicted results with actual values. Furthermore, we calculate the server power consumption based on the actual power consumption according to the CPU usage of different servers provided by SPECpower, a benchmark for evaluating server power efficiency. Additionally, we calculate the amount of computation required for the program and deep learning model of the proposed. Based on this, we derive practical results considering the trade-off between the server's power saving and computational performance.

The contributions of this study are as follows:

- This study proposes a novel model for predicting irregular traffic using a time-series deep learning model to reduce the servers' energy consumption.
- For practical validation of the CDSM, we use real traffic and servers' power consumption based on their CPU usage.
- To provide a realistic basis for the management of server free resources and energy, we consider their CPU usage and the scope of guaranteeing QoS for network communication.
- We use statistical methods to validate the proposed time-series deep learning-based traffic prediction model.
- We propose a new evaluation metric for the validation of CDSM.
- To validate the proposed method practically, we calculate the amount of computation for the program and deep learning model and consider the trade-off of server power savings.

The remainder of this paper is organized as follows. Section II reviews research papers related to this study. Section III describes in detail the proposed CDSM management of server resources. Section IV describes the experimental design, including the details of the experiments to be conducted in this study and the evaluation metrics. Section V evaluates whether the CDSM shows significant results compared to conventional management approaches. Section VI examines three aspects that can be additionally created in the process of this study. Finally, Section VII summarizes the findings of this study to present the conclusions and describes future research.

II. RELATED WORK

This section introduces previous studies on deep learning-based time-series prediction, server control for energy saving, and reduction of electricity usage in data centers. Studies have been conducted on deep learning-based anomalous traffic detection, data placement and power management on servers, and dynamic server power mode control [16], [17], [25], [26], [27], [28].

A. NETWORK MANAGEMENT BASED ON A TIME-SERIES PREDICTION MODEL

A hybrid model that combines Convolutional Neural Network (CNN), which extracts input features, and LSTM, which exhibits high performance on time-series data, was proposed [25]. Time-series traffic provided by Yahoo! was analyzed using the proposed model. The proposed model was able to detect abnormal traffic such as Denial of Service, Probe, User to Root, and Remote to User, and showed superior performance over other machine learning models. It showed 98.6% accuracy and 89.7% recall on test data. Network traffic prediction models were also proposed using LSTM, CNN, and Seasonal ARIMA (SARIMA) models. The SARIMA model added seasonal components to the conventional AutoRegressive Integrated Moving Average (ARIMA) model [26]. The proposed models contributed to efficient network traffic management with improved performance over the conventional models. The LSTM and CNN-LSTM hybrid models showed higher precision than the SARIMA model and reduced the error rate by 11%.

B. SERVER CONTROL FOR ENERGY SAVING

To reduce the energy consumption of servers, this study proposes a method of concentrating the load on specific servers and then shutting them down based on the system situation [16]. Conventional methods require high energy consumption to support load balancing between storage devices in a multimedia server. The proposed study showed that the number of servers running was reduced compared to that of the conventional methods, reducing the energy consumption significantly.

A method was proposed to reduce energy consumption by analyzing servers' power consumption history to predict whether servers' power consumption will increase at particular times, and then operating or stopping the servers based on this method [17]. After collecting power consumption data from each server for a certain duration, it predicts whether the power consumption would increase compared to before. The proposed study showed a 29% reduction in energy consumption while maintaining the same performance.

C. COOLING CONTROL FOR ENERGY SAVING

An artificial neural network (ANN)-based optimal cooling water flow control algorithm for data centers has been proposed for efficient cooling systems in data centers [27]. The ANN-based control algorithm demonstrated performance improvement in terms of accuracy, stability, and energy saving compared to conventional methods.

A multi-outside air-cooling system that uses outside air to save cooling energy in data centers was proposed [28]. The multi-outside air-cooling system was found to save 26.7% of the total cooling energy.

D. DYNAMIC SERVER CONTROL FOR SERVER POWER CONSUMPTION REDUCTION

A resource allocation system has been proposed to reduce the number of servers while avoiding overloads in the system operation of a data center [29]. The resource allocation system predicted future resource usage through a load prediction algorithm to prevent system overload. Furthermore, resource skewness for server p was introduced, which measures the degree of uneven utilization of the server. The more uneven the server's utilization, the larger is its value. By minimizing the server's resource skewness, the overall utilization of the server is improved, showing that the proposed algorithm performs well. An optimal power minimization approach with improved task scheduling was proposed for an efficient dynamic resource allocation process [30]. The proposed approach used a prediction mechanism and a Dynamic Resource Allocation Table updating algorithm to improve task scheduling. The simulation results showed that the proposed approach allocates resources efficiently with an improvement of 8% over conventional methods in terms of task completion and response time. This improved task scheduling and contributed to reducing power consumption in data centers.

By reviewing related studies, we determined that various studies have been conducted to reduce the energy consumption of servers. However, in terms of traffic prediction, resource trend prediction, and server control in data centers, still more improvement is required to respond quickly to traffic overload caused by sudden fluctuations in traffic. By contrast, CDSM facilitates response to traffic overload caused by sudden fluctuations in traffic as it uses a time-series deep learning model to predict traffic and, simultaneously, dynamically manages it according to the server situation. Furthermore, the methods proposed in previous studies do not dynamically control servers and are thus inefficient in terms of cooling in the data center because of a lot of heat being generated by many servers. That is, CDSM can be used together to reduce the energy consumption in the data center more effectively. Therefore, research on dynamic server management methods is still a challenging issue. Accordingly, this study focuses on a method of dynamically managing servers according to the server situation through a time-series deep learning-based traffic prediction model. Based on the proposed study, energy consumption can be reduced more efficiently by managing servers reliably and efficiently when traffic surges or decreases.

III. CUSTOMIZED DYNAMIC SERVER MANAGEMENT (CDSM)

The proposed method consists of four steps, as described in Fig. 1.

A. STEP 1: PREDICTING TRAFFIC USING TIME-SERIES DEEP LEARNING

CDSM uses a time-series deep learning model to predict irregular traffic to reduce the energy consumed by servers and uses only the servers needed for the traffic generated. To reduce the energy used by servers, some servers are turned off during low-traffic hours and others are turned on during high-traffic hours. However, during a sudden surge in traffic

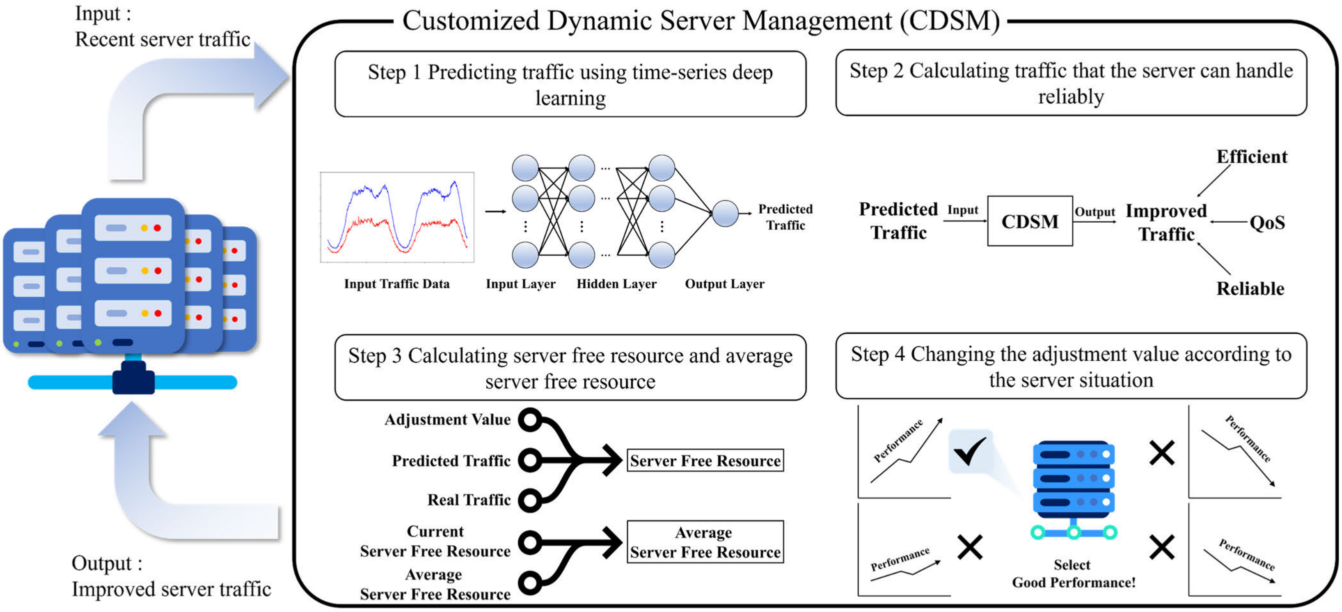

Input :
Recent server traffic
Customized Dynamic Server Management (CDSM)
Step 1 Predicting traffic using time-series deep learning
Step 2 Calculating traffic that the server can handle reliably
Input Traffic Data
Input Layer
Hidden Layer
Output Layer
Predicted Traffic
Predicted Traffic
Input
CDSM
Output
Improved Traffic
Efficient
QoS
Reliable
Step 3 Calculating server free resource and average server free resource
Step 4 Changing the adjustment value according to the server situation
Adjustment Value
Predicted Traffic
Real Traffic
Server Free Resource
Current Server Free Resource
Average Server Free Resource
Average Server Free Resource
Performance
Performance
Performance
Performance
Select Good Performance!
Output :
Improved server traffic

FIGURE 1. Overview of CDSM.

at a certain time, responding immediately is difficult because predicting how much server capacity should be secured on the server side is challenging. This situation occurs quite frequently. For example, Fig. 2 shows a graphical representation of server traffic trends from August 2 to August 3, 2022 collected from the actual Wikipedia servers located in Netherlands [31]. In Fig. 2, the Y-axis represents traffic (byte/s), and the X-axis represents time in 4-min increments. Fig. 2 shows a spike in traffic at a specific time, in the time periods of 500s on the X-axis. In this graph, the average rate of change is approximately 0.001e+9; however, the instantaneous rate of change between 500 and 521 is approximately 0.074e+9, which is 74 times higher than the average rate of change. Therefore, this spike corresponds to a level of traffic overload that can be considered an outlier. As sudden surge in traffic is common, a time-series deep learning model can be trained on traffic in such situations to predict sudden traffic overloads in advance. Therefore, if servers are managed by predicting traffic through a trained time-series deep learning model, coping with sudden traffic overloads in advance and responding quickly to sudden traffic surges will be possible. To predict traffic for server management, a time-series deep learning model is trained on the servers' traffic. However, if there is highly correlated data in the servers' traffic, it indicates an imbalance of data classes, which can lead to a multicollinearity problem [32], [32], [34]. This is a factor that can reduce the model's performance. Therefore, to improve the traffic prediction model's performance in this study by avoiding the multicollinearity problem, we select input variables by examining the correlation between the data through statistical analysis. Then, using the traffic as input after removing the data imbalance based on statistical analysis, we train the prediction model and predict the traffic to manage the servers according to the system situation. For the predicted traffic, CDSM suggests the maximum traffic throughput that the server can handle reliably.

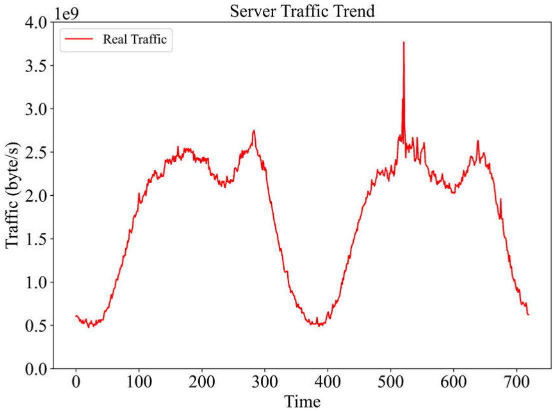


FIGURE 2. Graph of server traffic trend.

B. STEP 2: CALCULATING TRAFFIC THAT THE SERVER CAN HANDLE RELIABLY

In the previous step, traffic was predicted to control the servers based on the amount of traffic generated. To control the servers reliably, we must determine the maximum traffic throughput that the server can handle reliably for the predicted traffic. If this is calculated, the server can be controlled reliably as it can be prevented from going down even if the traffic is somewhat underestimated. To manage the server reliably and efficiently, an efficient CPU usage should be considered along with QoS guarantees. For example, in cases where the traffic is predicted but the server is managed for a CPU usage of 90% to reduce the number of servers blindly, a problem will occur where QoS will not be guaranteed [23], [24]. Therefore, a process is required to convert the predicted traffic into the maximum traffic throughput of the server for server control considering an efficient CPU usage along with QoS guarantee. Therefore, the maximum traffic throughput that the server can reliably handle for the traffic is suggested by calculating the adjustment value from the traffic predicted by the deep learning-based traffic prediction model. As mentioned earlier, the server runs efficiently with QoS guarantee when its CPU usage is less than or equal to 70% [35], [36], [37]. Therefore, to increase the CPU usage to reduce the number of servers running with QoS guarantees, we change the adjustment value dynamically between 1.4 (1/1.4=71.4%) and 1.8 (1/1.8=55.6%) in the predicted traffic depending on the server situation [35], [36], [37]. For example, when the adjustment value is 1.4, the server's CPU usage is 71.8% (Traffic/1.4∗Traffic=71.8%). When the adjustment value is 1.8, it is approximately 55.6% (Traffic/1.8∗Traffic=55.6%). As the CPU usage is below 70%, the server runs efficiently while guaranteeing QoS. For reliable and efficient management, this method changes the adjustment value dynamically; it increases the number of servers by increasing the adjustment value when the server situation is unstable and decreases the number of servers by decreasing the adjustment value when the server situation is stable. Servers can be managed more reliably and efficiently by reflecting the server situations in the case of dynamically changing the adjustment value compared to the case of fixing it. As the adjustment value is used, servers can be managed smoothly and efficiently. If the predicted traffic is somewhat lower than the actual value, the servers can be managed smoothly by increasing the number of servers by increasing the adjustment value. If the predicted traffic is higher, the servers can be managed efficiently by decreasing the number of servers by decreasing the adjustment value.

C. STEP 3: CALCULATING SERVER FREE RESOURCE AND AVERAGE SERVER FREE RESOURCE

In the previous step, the maximum traffic throughput that the server can reliably handle for a given traffic was suggested. This was calculated from the predicted traffic using an adjustment value. The adjustment value is changed dynamically to manage the server reliably and efficiently to reflect the server situation. Therefore, we require a baseline to identify the server situation. Accordingly, we calculate the server free resource and the average server free resource, which are the basis for changing the adjustment value according to the server situation. To calculate the server free resources, we propose Eq. (1) in this study. The traffic value suggested by CDSM minus the actual traffic is called the server free resource, as shown in Eq. (1), which is used to validate the CDSM.

Here, CDSM is validated with the minimum, maximum, and average values of the server's free resources. The higher the minimum value of the server free resource, the more sufficient the capacity to handle traffic reliably, indicating that the server is managed stably. For example, suppose the minimum value of the server free resource is 0.1 GB, which means that the server is not managed reliably because if the actual traffic was higher than that even slightly, the server could go down. Also, the lower the maximum and average values of the server free resource, the less capacity is wasted, indicating efficient management of the server. The maximum value of the server free resource is a metric for evaluating the degree of efficient management under high traffic conditions. On the other hand, the average value of the server free resource is a metric for evaluating the degree of efficient management in the overall situation. For example, suppose the average value of the server free resource for a reliable operating server is 1 GB, and the maximum and average values of the server free resource obtained through CDSM are 3 GB and 2 GB, respectively. The server is then judged to be managed inefficiently because their difference in capacity is wasted while managing the server. These values are real numbers, and the higher the minimum value of the server free resource and the lower the maximum and average values of the server free resource, the higher the server management performance.

Therefore, to obtain the average value of server free resources, this study proposes and calculates Eq. (2). If the average server free resource is calculated by storing all the server free resources, the amount of computation required is too large. Therefore, we use Exponentially Weighted Averages to reduce the amount of computation. Using Exponentially Weighted Averages results in a slightly inaccurate average server free resource, but over time, as the average free resource is calculated repeatedly using Exponentially Weighted Averages, it gets closer to the exact value of the server free resource. Therefore, the average free resource calculated by Exponentially Weighted Averages is sufficient as a baseline for identifying the server situation. Furthermore, if the average server free resource is calculated accurately by storing and computing all server free resources, it will increase the amount of computation and power consumption, reducing the advantages of using CDSM. The server free resource and the average server free resource are used in changing the adjustment value to manage servers dynamically. The average server free resource is used as a baseline to evaluate the state of the server up to this point, and the server free resource is used to evaluate the situation according to this baseline. For example, if the server free resource is higher than the baseline, the server is considered to be in a state with spare resources; conversely, if it is lower, the server state is considered to be unreliable.

After determining the server's situation through the server free resource and average server free resource, the servers can be managed reliably and efficiently by dynamically changing the adjustment value. The number of servers can be reduced by decreasing the adjustment value if there are servers with wasted resources. The number of servers can be increased by increasing the adjustment value if it is difficult to process traffic reliably due to traffic overload. Therefore, by changing the adjustment value with the deep learning model according to the server situation, the servers can be managed more reliably and efficiently, even with rapid fluctuations in traffic.

$$S_{\text{free}}(\mathrm{T}) = c \times \text{Predicted}(\mathrm{T}) - -\text{Real}(\mathrm{T}) \tag{1}$$

S: server free resource , *c:* adjustment value*, T:* traffic

$$S_{\text{avg}}(\mathrm{T}) = 0.01 \times S_{\text{now}}(\mathrm{T}) - -0.99 \times S_{\text{avg}}(\mathrm{T}) \tag{2}$$

S: server free resource, now: current server free resource, avg: average server free resource

D. STEP 4: CHANGING THE ADJUSTMENT VALUE ACCORDING TO THE SERVER SITUATION

In the previous step, the server free resource and the average server free resource were calculated, and they were termed as the basis for changing the adjustment value according to the server situation. This is to change the adjustment value dynamically by reflecting the server situation to manage the server reliably and efficiently. If the server free resource is higher than the baseline, the server is considered in a state with spare resource, and if it is lower, the server is considered in an unstable state. We require a method to change the adjustment value dynamically based on this result: if there are servers that are wasting resources, the number of servers is reduced by decreasing the adjustment value; if it is difficult to handle traffic reliably due to traffic overload, the number of servers is increased by increasing the adjustment value. Accordingly, the adjustment value is calculated to suggest the maximum traffic throughput that the server can reliably handle. The initial adjustment value is set to 1.5, and the server situation is identified based on the average of server free resources up to this point. If the server free resource is greater than or equal to the baseline, 0.001 is subtracted from the adjustment value. Afterward, if the server free resource is greater than or equal to the baseline successively, 0.001 is multiplied by 2 repeatedly to subtract the result from the adjustment value. Conversely, if the server free resource is less than the baseline, 0.001 is added to the adjustment value. Afterward, if the server free resource is less than the baseline successively, 0.001 is multiplied by 2 repeatedly to add the result to the adjustment value. The adjustment value is calculated based on 0.001 to provide a fine-grained change. Here, the lower and upper limits of the adjustment value are set to 1.4 and 1.8, respectively. This demonstrates that the server operates efficiently with QoS guarantees when CPU usage is less than or equal to 70% [35], [36], [37]. Algorithm 1 shows the pseudocode for this algorithm, and Algorithm 2 shows the pseudocode for the entire algorithm of CDSM.

If the adjustment value is dynamically adjusted to a value between 1.4 and 1.8 depending on the server situation, the servers can be managed more reliably by increasing the number of servers by increasing the adjustment value when traffic increases sharply. Moreover, when traffic declines sharply, the number of servers can be reduced by decreasing the adjustment value. That is, the effect of reducing energy consumption can be expected as the servers are managed more efficiently.

Algorithm 1 Changing the Adjustment Value According to the Server Situation

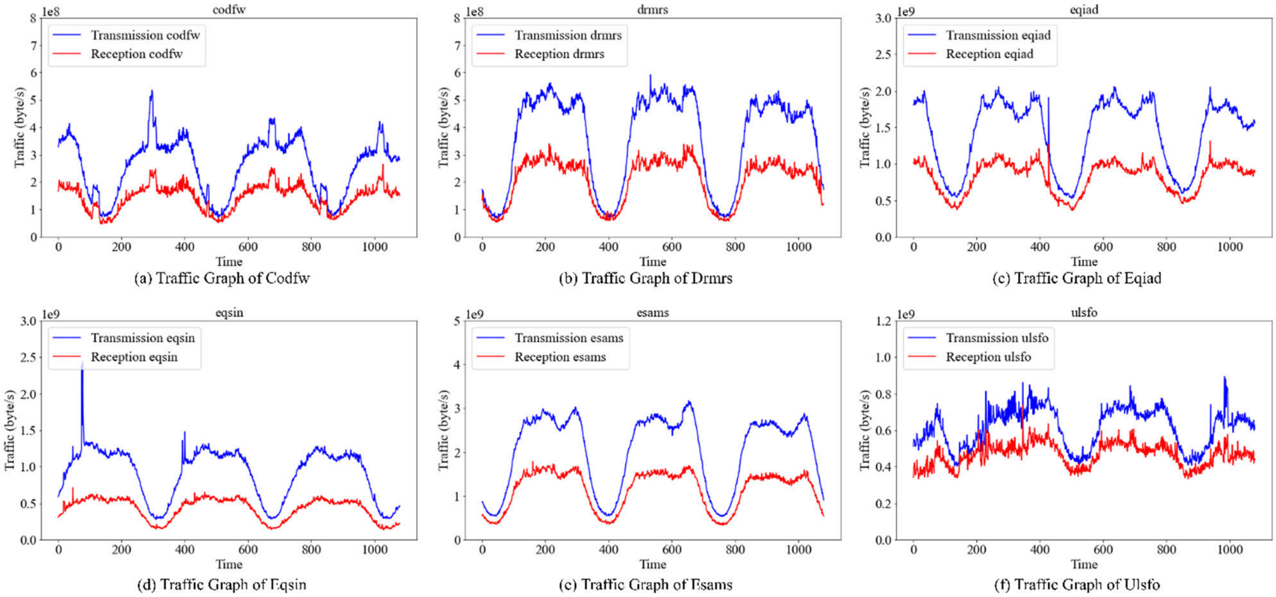
```
function ModifyAdjustmentValue(Stable, Danger, Free_Resource,
Free_Resource_Avg, Adjustment_Value)
begin
if Free_Resource>=Free_Resource_Avg then
    // The adjustment value is reduced if the server is in a state with spare
    resources.
    Stable * = 2
    Danger = 1
    Adjustment_Value -= Stable*0.001
else
    // The adjustment value is increased if the server is in an unstable state.
    Stable = 1
Danger * = 2
    Adjustment_Value + = Danger*0.001
Endif
    // The adjustment value is prevented from exceeding the lower and upper
    limits.
if Adjustment_Value<1.4 then Adjustment_Value = 1.4
    if Adjustment_Value>1.8 then Adjustment_Value = 1.8
    return Adjustment_Value, Stable, Danger
end
```

Algorithm 2 CDSM

```
function CDSM()
begin
Stable = 1 // The adjustment value calculation variable used when the server is in
        a state with spare resource
Danger = 1 // The adjustment value calculation variable used when the server is in
        an unstable state
While True
    // Calculate server free resources and average of server free resources after
    predicting network traffic
    Traffic_Predict = Result of Traffic Prediction
    Free_Resource = Traffic_Predict * Adjustment_Value - Real_Traffic
    Free_Resource_Avg = 0.01*Free_Resource + 0.99*Free_Resource_Avg
    // Calculating Adjustment Value
    Adjustment_Value, Stable, Danger = ModifyAdjustmentValue(Stable,
    Danger, Free_Resource, Free_Resource_Avg, Adjustment_Value)
Endwhile
end
```

IV. DESIGN OF EXPERIMENTS

In the previous Section III, we proposed CDSM and the traffic prediction model used in CDSM. It is necessary to verify whether the actual traffic prediction model is accurate and whether the CDSM is stable and efficient. First, it is necessary to select input variables that were used in training the traffic prediction model and verifying them through performance measurement of the time-series prediction model. Furthermore, it is necessary to verify the server management performance of CDSM, as well as the trade-off between the amount of server power reduction and the amount of computation in CDSM. To validate the proposed method, we use correlation analysis and a variance inflation factor (VIF) for input variable selection and R2 Score for model



(a) Traffic Graph of Codfw (b) Traffic Graph of Drmrs (c) Traffic Graph of Eqiad

(d) Traffic Graph of Eqsin (e) Traffic Graph of Esams (f) Traffic Graph of Ulsfo

FIGURE 3. Traffic graphs of data centers.

performance measurement. As for the server management performance of CDSM, we use the server power consumption measured based on the actual power consumption according to the CPU usage(provided by SPECpower). Additionally, we use the minimum, maximum, and average values of server free resources. Finally, we use floating point operation per second (FLOPs) to calculate the trade-off between the server power reduction and the amount of computation in CDSM.

TABLE 1. Descriptions of data centers.

Data Center	Description
Eqiad	Data center located in Ashburn, USA
Codfw	Data Center in Carrollton, Texas, USA
Esams	Data center located in Amsterdam, The Netherlands
Ulsfo	Data Center located in San Francisco
Eqsin	Data Center located in Singapore
Drmrs	Data Center located in Marseille, France.

A. DATA USED

In this study, we collect the traffic data transmission and reception at Wikipedia's six data centers (Eqiad, Codfw, Esams, Ulsfo, Eqsin, and Drmrs) in Wikitech [31]. Table 1 provides the descriptions of the data centers. We use 3.600 data, which are 10-day data of 4-minute intervals from April 1 to April 10, 2022, as the train set, and 1.800 data, which are 5-day data of 4-minute intervals from August 1 to August 5, 2022, four months later, as the test set.

TABLE 2. Traffic data.

Time	Transmission data in the Esams Data Center (byte/s)
2022-08-01 0:00	876,219,108
2022-08-01 0:04	914,283,844
2022-08-01 0:08	936,286,381
2022-08-01 0:12	878,264,774
2022-08-01 0:16	826,663,298

Table 2 shows some of the test set data, and Fig. 3 shows the graphs of transmission and reception traffic that we collected from the six data centers. Traffic is high in the afternoon and low at dawn depending on the activity of the people. In each graph, the horizontal axis represents time and the vertical axis represents traffic at that time.

B. DATA PREPROCESSING

In this section, we report statistical experiments to examine how the transmission and reception traffic of the six data centers affects the transmission traffic of the Esams data center. The Pearson correlation coefficient for the traffic is used to analyze and validate the correlation between each traffic [38].

Fig. 4 shows the Pearson correlation coefficients for the transmission and reception traffic of the six data centers. The correlation coefficients between the transmission and reception traffic of the Drmrs data center/the reception traffic of the Esams data center, and the transmission traffic of the Esams data center have values between 0.97 and 0.99. A very strong correlation was found in the Esams data center's transmission traffic, with no significant deviation in values and an almost perfect linear relationship. The correlation coefficients between the Ulsfo data center's transmission and reception traffic and the Esams data center's transmission traffic are −0.62 and −0.47, respectively, indicating that impacts are not small overall. Finally, the Pearson correlation coefficients between the transmission and reception traffic of the Codfw, Eqiad, and Eqsin data centers and the transmission traffic of the Esams data center are between -0.33 and 0.044, respectively, indicating almost no correlation.

Table 3 shows the VIF between traffic. If its value is greater than or equal to 10, it means that the variable is not independent [39]. As all the VIF values are below 10, these variables are mutually independent. Thus, there is no multicollinearity between all the traffic, and the variables are mutually independent.

Based on the combined results of the two experiments, we used the transmission and reception traffic of Drmrs and Esams data centers, among the transmission and reception traffic of the six data centers, as input variables since they have very strong correlations with the transmission traffic of Esams data center and are mutually independent variables.

TABLE 3. **Result of VIF.**

Feature	VIF Factor
Reception codfw	0.345307
Reception drmrs	-8.14538
Reception eqiad	0.363808
Reception eqsin	2.389091
Reception esams	0.665438
Reception ulsfo	-1.50497
Transmission codfw	3.05752
Transmission drmrs	-4.15771
Transmission eqiad	0.127355
Transmission eqsin	0.043306
Transmission esams	0.3222
Transmission ulsfo	-1.47156

C. CONSTRUCTION OF TRAFFIC PREDICTION MODEL

In this study, we used a deep learning model based on TensorFlow 2.7.0 and Keras 2.7.0, and used LSTM, which has a high performance in predicting time-series data [19], [20], [21], [22]. The transmission and reception traffic of Drmrs and Esams data centers are used as input variables, and the final output is the future traffic. For the hyper-parameters, we experimented with various combinations to set the values that show optimal performance and low time cost. Table 4 shows the hyper-parameters used in the experiments and the parameter values showing the best performance among them.

TABLE 4. **Values of LSTM Hyper-parameters.**

Hyper-parameter	Value
Unit	[25-100] 50
Dropout	[0.05-0.5] 0.2
Activation	[Relu, Sigmoid, Tanh] Tanh
Optimizer	[SGD, RMSprop, Adam] Adam
Learning_rate	0.001
Batch_size	32
Epoch	1000
Window size	[1-45] 15

Window size is a way to represent temporal characteristics. As the window size increases, more temporal characteristics are reflected, but at a higher time cost. Conversely, as the window size decreases, less temporal characteristics are reflected, but at a lower time cost. Fig. 5 is a graph showing the experimental results for the window size hyper-parameter. As the traffic is collected in 4-min intervals, 15 traffic samples are collected in an hour. Accordingly, we conducted the experiment by increasing the window size by 15. When the window size is not set, the traffic is not predicted accurately compared to other results. This can be confirmed through a lower R2 score compared to the other results. When the window size is set, traffic is predicted more accurately unlike when the window size is not set. While traffic is predicted more accurately compared to the case of not setting the window size, an overfitting trend is seen as the window size increases, resulting in lower performance. Therefore, we set the window size to 15, which shows the best performance.

D. SETTING THE SERVER POWER CONSUMPTION MEASUREMENT ENVIRONMENT

This section describes an experiment to measure the server power consumption for the proposed method based on the actual power consumption according to the CPU usage of different servers provided by SPECpower. To include a wide range of situations, we assumed a total of 250 servers in the experiment. Assuming that 20, 50, 100, 150, and 200 MB/s of traffic throughput per server, we conducted the experiment for four server types. For the servers, there is no actual power consumption at all CPU utilization rates as shown in Table 5. Therefore, if a CPU usage is not in the table, the power consumption is calculated using linear interpolation with the two closest values among the CPU usages in the table. For example, if the CPU usage is 65%, we use linear interpolation with 60% and 70% to calculate the power consumption equivalent to 65%, as shown in Table 5. CPU

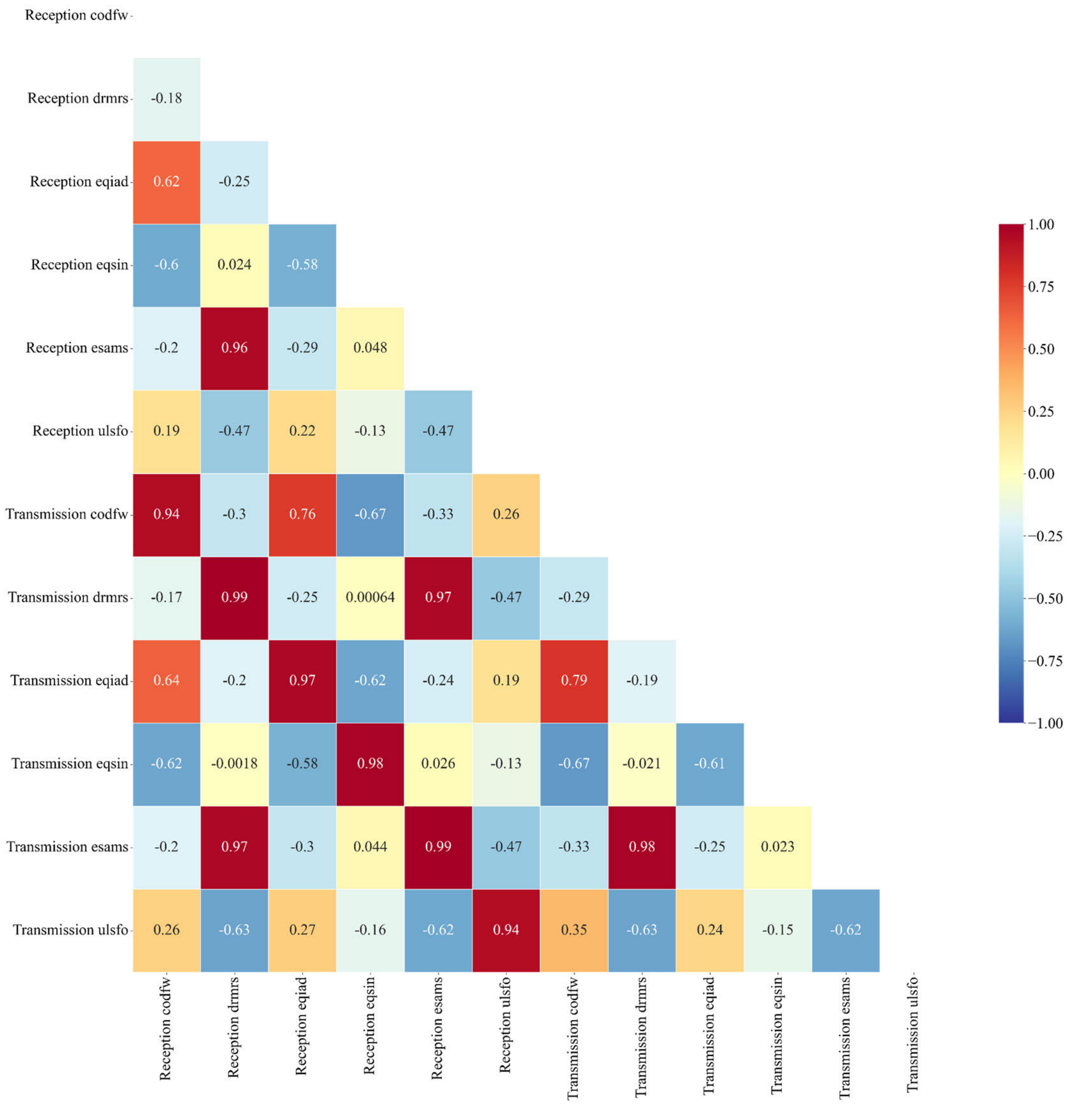


FIGURE 4. Results of the correlation analysis.

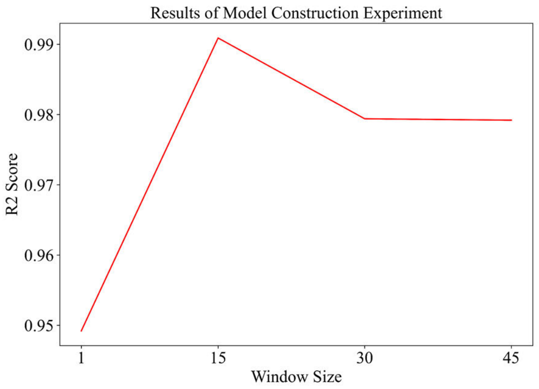


FIGURE 5. Experimental results by changing the window size (R2 Score).

utilization U(T) is calculated as shown in Eq. (3). Among the existing studies, formulas for CPU utilization using time metrics have been proposed, but since the experiments in this study focus on using traffic metrics rather than time metrics, a new formula for traffic-based CPU utilization has been proposed and utilized. Max(T) is the maximum traffic that can be handled by the minimum server for the traffic suggested by CDSM, and to calculate it, we propose Eq. (4). For example, if traffic throughput per server is 50 MB/s while the traffic suggested by CDSM is 345 MB/s, then Max(T) is 350 MB/s. The servers used in the experiment are as follows, and Table 5 shows the actual power consumption according to the CPU usage for each server [40].

a) Inspur Corporation Inspur NF5280M5
b) Lenovo Global Technology Think System SR645 V3
c) Dell Inc. PowerEdge 2950 III (Intel Xeon E5440)
d) Hewlett Packard Enterprise ProLiant ML350 Gen11

$$U(T) = \frac{Real\ Traffic}{Max(T)} \tag{3}$$

U: CPU usage, Real Traffic: real traffic, Max: maximum traffic that can be handled by servers, T: traffic

$$Max(T) = \begin{cases} Floor\left(\frac{c*Pred(T)}{a}\right) \times a, \&c \times Pred\,(T)\,\%a = 0 \\ Floor\left(\frac{c*Pred(T)}{a}\right) \times a + a, \&c \times Pred\,(T)\,\%a > 0 \end{cases} \tag{4}$$

Max: maximum traffic that can be handled by servers, c: adjustment value, T: traffic, α: traffic throughput per server

TABLE 5. Relationship between CPU usage and power usage.

Class	CPU Utilization	Average Active Power(W)	Class	CPU Utilization	Average Active Power(W)
a	99.6%	289	b	99.6%	658
	90.0%	261		89.9%	589
	80.2%	242		79.8%	524
	70.0%	212		70.0%	483
	60.0%	198		59.9%	450
	50.0%	184		50.0%	413
	40.0%	175		39.9%	386
	30.0%	161		30.0%	355
	20.0%	150		20.0%	306
	10.0%	138		10.0%	257
	0%	112		0%	126
c	99.7%	276	d	99.8%	611
	90.3%	270		90.0%	573
	80.1%	262		80.1%	529
	70.3%	253		70.0%	486
	60.1%	243		60.0%	445
	49.7%	230		50.0%	409
	40.1%	217		40.1%	376
	30.2%	204		30.0%	342
	20.0%	189		20.0%	308
	10.2%	173		10.1%	274
	0%	157		0%	154

E. CDSM EVALUATION METRICS

This section describes the evaluation metrics for validating the CDSM. As for the proposed traffic prediction model's performance, R2 score is used as an evaluation metric to compare the difference between the actual and predicted values. We calculated the server power consumption for the proposed method based on the actual power consumption according to the CPU usage of different servers provided by SPECpower. As CPU usage is highly correlated with server power consumption, we calculated the server power consumption based on CPU usage [41], [42], [43]. We also calculated the amount of computation for the program and deep learning model by quantifying them in FLOPs, and based on this, we validated the CDSM by considering the trade-off between server power reduction and computational performance.

1) R2 SCORE

We used the R2 score to measure the performance of the traffic prediction model. The R2 score is the sum of the squared errors divided by the sum of the squared deviations minus one. It is a metric that measures the goodness of fit of a linear regression model. It has a value between 0 and 1, with higher values indicating higher prediction performance [44], [45].

2) MINIMUM VALUE OF SERVER FREE RESOURCE

The higher the minimum value of the server free resource, the more sufficient the capacity to handle traffic reliably, indicating that the server will not go down and is managed stably. Conversely, when the value is low, it is determined that the server is not managed stably. These values are real numbers, with higher values indicating higher server management performance.

3) MAXIMUM AND AVERAGE VALUES OF SERVER FREE RESOURCE

The lower the maximum and average values of server free resources, the lesser the wastage of capacity, indicating that the server is managed efficiently. The higher the value, as opposed to the case of the minimum value, the more inefficiently the server is managed. The maximum and average values of server free resources are real numbers such as the minimum value, and the smaller the value, the higher the server management performance.

4) FLOPS

FLOPs refer to the number of floating-point operations. FLOPs are a suitable metric for actual waiting time and energy usage and a better metric for evaluating energy usage and waiting time than the parameters [46], [47], [48]. We used it to express the amount of computation for the program and deep learning model. It has an integer value, and a smaller value means a smaller amount of computation.

V. EXPERIMENTAL RESULTS

A. EVALUATION OF SERVER MANAGEMENT PERFORMANCE BASED ON ANALYSIS OF SERVER FREE RESOURCE EVALUATION METRICS

Table 6 shows the results of the test set in terms of CDSM evaluation metrics. The rows of Table 6 show the evaluation metrics of server free resources introduced in Section IV-E, and the columns consist of static server management method, CDMS method proposed in this study, and ratio of the difference between their results. Fig. 6 is a graph for the period from August 1 to 2, 2022 in the test set, and Fig. 7 is a graph of server free resources and average server free resources for the same period as Fig. 6. The X-axis in Figs. 6 and 7 represents time in 4-min increments, and the Y-axis represents traffic (byte/s). Here, Static Server Management refers to a method of managing servers by fixing the adjustment value to 1.5 [18]. In terms of R2 score, the two results in Table 6 are from the same model. The R2 score of 0.991 means that the proposed model has a 99.1% goodness-of-fit for the actual values. This implies that the proposed model predicts the actual traffic with high accuracy, indicating that it is suitable for server management.

Server free resources are calculated as shown in Eq. (1). It is an indicator to assess how stably and efficiently the server is managed. The minimum value of the server free resource is used to determine whether the server was managed stably, and the average and maximum values of the server free resource are used to determine whether the server was managed efficiently. As a result of comparing CDSM to Static Server Management for the average and maximum values of server free resources, the CDSM reduces the average and maximum values of server free resources by approximately 12%. By contrast, as a result of comparing CDSM to Static Server Management for the minimum value of server free resources, the CDSM reduces the minimum value of server free resources by approximately 11.8%. This decrease is because the average server free resource is calculated as shown in Eq. (2), which is not an accurate average value of server free resources initially. As shown in Fig. 7, this can be confirmed by the fact that the average server free resource initially starts at 0 and gradually approaches the correct average server free resource. When the correct average server free resource is approached after a certain time, the minimum value of the server free resource is 0.3425, which is an increase of approximately 70%. Therefore, it is safe to say that the server is managed reliably. That is, the CDMS proposed in this study shows excellent experimental results overall in terms of efficiency and reliability.

Fig. 8 shows a graph of the adjustment value for the same period as Fig. 6. The X-axis represents time in 4-min increments, and the Y-axis represents the adjustment value. As shown in Fig. 6, the CDSM facilitates more free server resources than Static Server Management when traffic is low and when traffic surges. When traffic is high and when it decreases sharply, the use of CDSM leads to less free server resources. The reason is that the adjustment value is reduced to 1.4 in high traffic situations to manage the servers efficiently because the server free resource is higher than the average server free resource, as shown in Figs. 7 and 8. In low traffic situations, on the other hand, the server free resource is lower than the average server free resource, so the adjustment value is increased to 1.8 for reliable management. This means that if the adjustment value is changed dynamically when managing servers, more reliable and efficient server management will be feasible.

B. EVALUATION OF SERVER POWER CONSUMPTION

Table 7 and Fig. 9 show the results of the experiments described in Section IV-D. The rows in Table 7 show the experimental results for the servers introduced in Section IV-D, and the columns consist of the number of servers used in the experiment, the traffic throughput per server, the server management method type, and the power consumption. Fig. 9 shows a graph for Table 7, where the X-axis represents the traffic throughput per server and Y-axis represents the power consumption.

Metrics such as variance and standard deviation of power consumption are also available, but they are not suitable for use as evaluation metrics because the trend of these metrics changes significantly depending on the power consumption of the server's CPU usage.

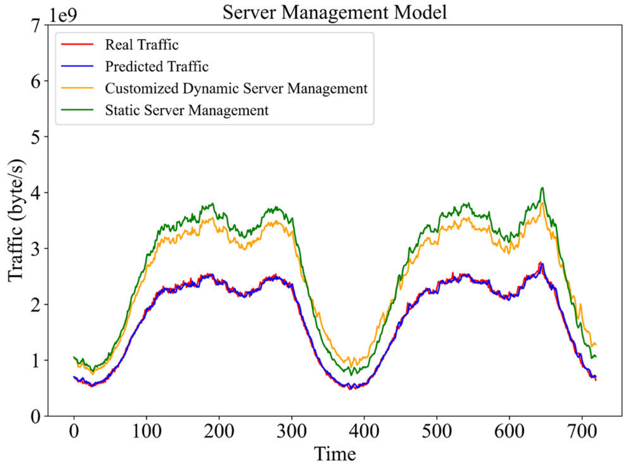


FIGURE 6. Traffic graphs of experimental results.

TABLE 6. Summary of experimental results.

	Static Server Management	CDMS	Rate of Change
R2 score	0.991	0.991	-
Server Free Resources Average	0.8681GB	0.7827GB	-9.8%
Server Free Resources Maximum	2.3785GB	2.0407GB	-14.2%
Server Free Resources Minimum	0.2017GB	0.1778GB	-11.8%

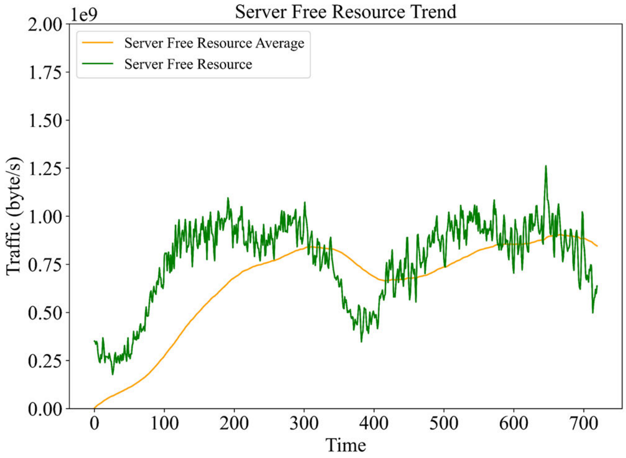


FIGURE 7. Graph of experimental results server free resource trend.

TABLE 7. Summary of experimental results.

Class	Number Of Servers	Traffic Throughput per Server	Server Management Type	Power Consumption (W)	Class	Number Of Servers	Traffic Throughput per Server	Server Management Type	Power Consumption (W)
a	250	20MB/s	Dynamic	26725.4572	b	250	20MB/s	Dynamic	60420.9829
			Static	26976.1840				Static	61399.6224
			Non	41786.996				Non	90001.1721
		50MB/s	Dynamic	10709.7348			50MB/s	Dynamic	24231.8542
			Static	10822.7640				Static	24632.8206
			Non	35237.8221				Non	66537.4037
		100MB/s	Dynamic	5373.3750			100MB/s	Dynamic	12173.2963
			Static	5440.9591				Static	12382.4320
			Non	32489.3971				Non	54114.8384
		150MB/s	Dynamic	3597.5342			150MB/s	Dynamic	8155.5468
			Static	3645.1801				Static	8294.5186
			Non	30996.8436				Non	46599.4736
		200MB/s	Dynamic	2708.8243			200MB/s	Dynamic	6143.6489
			Static	2748.6186				Static	6253.0543
			Non	30247.6400				Non	42824.6479
c	250	20MB/s	Dynamic	31651.4393	d	250	20MB/s	Dynamic	60607.7758
			Static	32468.1264				Static	61387.1016
			Non	52219.4687				Non	89278.1615
		50MB/s	Dynamic	12712.8015			50MB/s	Dynamic	24301.2435
			Static	13040.5845				Static	24613.0869
			Non	44750.6590				Non	69233.1973
		100MB/s	Dynamic	6402.8682			100MB/s	Dynamic	12202.4051
			Static	6568.3875				Static	12359.5182
			Non	41962.2100				Non	59211.3083
		150MB/s	Dynamic	4299.1741			150MB/s	Dynamic	8169.8785
			Static	4407.6826				Static	8271.7383
			Non	41058.0550				Non	52331.5713
		200MB/s	Dynamic	3244.5899			200MB/s	Dynamic	6150.3906
			Static	3329.1955				Static	6230.4494
			Non	40606.0422				Non	48873.7232

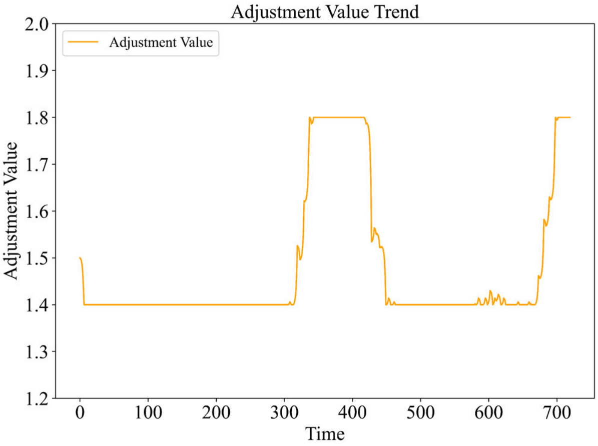
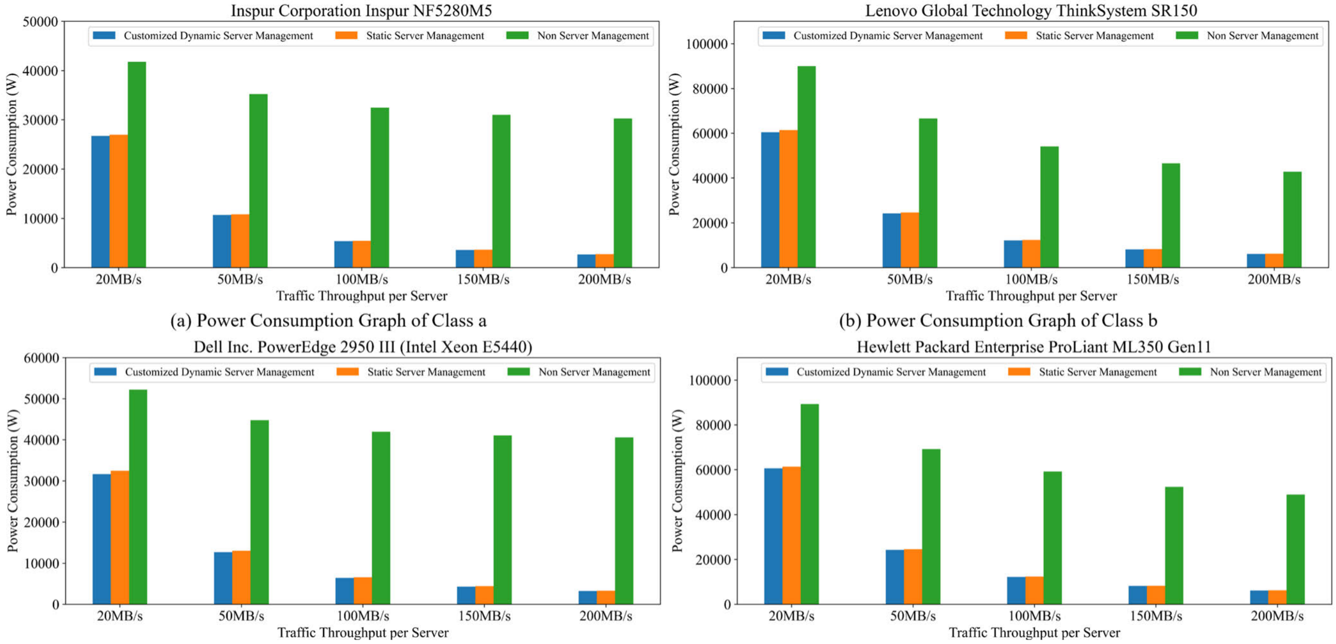


FIGURE 8. Graph of experimental results adjustment value trend.

As shown by the results in Table 7 and Fig. 9, when comparing the cases of not managing servers overall, managing with CDSM, and managing with Static Server Management in terms of traffic throughput per server and number of servers, we have found that server power consumption is lower when managed with CDSM and Static Server Management in all experiments. This shows that the power consumption of servers can be significantly reduced if the proposed deep learning-based traffic prediction model is used to manage the servers. Furthermore, the CDSM consumes less server power than the Static Server Management in all cases. In particular, the experimental results show that the difference is the highest for the case of Class b, 250 servers, and 20 MB/s traffic throughput, and the lowest for the case of Class a, 250 servers, and 200 MB/s traffic throughput. The reason is as follows. In the former case, the increase in power consumption of the servers is large according to the CPU usage of Class b, and simultaneously, the traffic throughput per server is low, resulting in many servers operating. In the latter case, the increase in power consumption of the servers is small according to the CPU usage of Class a, and simultaneously, the traffic throughput per server is high, thus resulting in a small number of servers operating. This shows



(a) Power Consumption Graph of Class a

(b) Power Consumption Graph of Class b

(c) Power Consumption Graph of Class c

(d) Power Consumption Graph of Class d

FIGURE 9. Power consumption graphs of experimental results.

that CDSM facilitates more reliable and, at the same time, efficient operation of servers than Static Server Management.

As shown in Table 7, the difference in power consumption between CDSM and Static Server Management decreases as the traffic throughput per server increases. This is because as the traffic throughput per server increases, the number of servers reduced by CDSM decreases. For example, suppose the same traffic is handled with 5 servers in Data Center A and with 30 servers in Data Center B. Further assuming that both data centers use CDSM to control servers and have an adjustment value of 1.4 (1/1.4=71.8%). Data Center A that has 5 servers will operate with 4 servers (5∗0.718=3.59), which is 1 server less, and Data Center B that has 30 servers will operate with 22 servers (30∗0.718=21.54), which is 8 servers less. Therefore, the power consumption reduction will be greater in Data Center B where 8 servers are reduced.

Based on this, we can see that CDSM is more effective in reducing power consumption for data centers that have many servers due to large traffic throughput. This means that CDSM facilitates reliable server management and, at the same time, efficient energy operation.

C. ANALYSIS OF TRADE-OFF BETWEEN PROGRAM AND DEEP LEARNING COMPUTATIONS AND SERVER POWER REDUCTION

If CDSM is used to control servers, servers can be managed reliably and efficiently. However, CDSM requires deep learning and computation to change the adjustment value dynamically. If the amount of computation is large, the reduction in power consumption achieved by CDSM is negated to an extent, and rather, controlling the servers with CDSM may increase power consumption. Therefore, we analyzed the trade-off between program and deep learning computations and server power reduction.

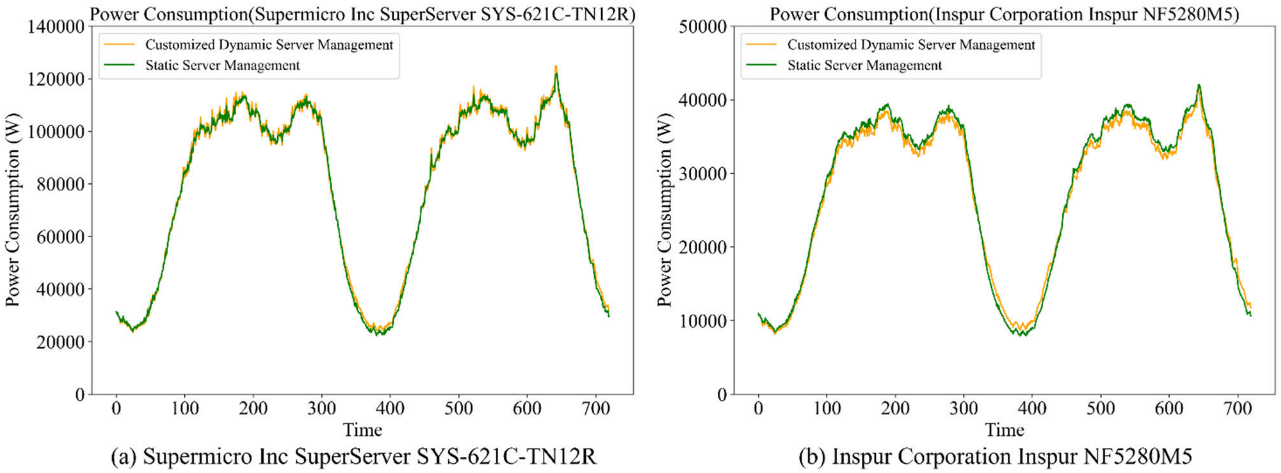
We used PyTorch to calculate the amount of computation for the deep learning model. The same model as the traffic prediction model was created and calculated using PyTorch, and the calculated program and deep learning computations are as follows.

- 174,000FLOPs per prediction
- 50.112 GFLOPs per train
- 10FLOPs per Adjustment Value Calculation

If CDSM is used for a day, the amount of computation will be approximately 62.64 MFLOPs. Recently, the Floating Operation per Second (FLOPS) per watt of a Graphic Processing Unit (GPU) has reached approximately 70 GFLOPS per watt [49], [50]. By comparison, the amount of computation performed by CDSM is very small, assuming that the deep learning-based traffic prediction model is retrained every three months. In addition, the time complexity of the operations used when managing servers with CDSM is only O(1) [19]. Based on this, the amount of computation performed by CDSM does not affect the amount of power reduction on the server.

VI. DISCUSSION

In this section, we discuss the optimal range of CDSM adjustment value, the measurement of server power consumption, and the reliability of CDSM in the face of rapid fluctuations in traffic.



(a) Supermicro Inc SuperServer SYS-621C-TN12R

(b) Inspur Corporation Inspur NF5280M5

FIGURE 10. Power consumption graphs of discussion experimental result.

A. OPTIMAL RANGE ANALYSIS OF CDSM ADJUSTMENT VALUE

The adjustment value range is a range that showed the optimal performance in the experiments. Table 8 shows the experimental results for different CDSM adjustment value ranges. The rows in Table 8 show the adjustment value ranges used in the experiment, and the columns consist of the results for the evaluation metrics of server free resources introduced in Section IV-E. As shown in Table 8, when the minimum value of adjustment value is set to 1.5, the minimum value of server free resource increases, but the average and maximum values of server free resource also increase, resulting in inefficient server management. When set to 1.3, the minimum value of the server free resource is very low, indicating that the servers are not managed reliably. Therefore, we set the minimum adjustment value to be used in the experiment to 1.4, which is the average of 1.3 and 1.5.

When the maximum value of adjustment value is reduced from 2 to 1.8, the average value of server free resource decreases, and there is no difference in the minimum value of server free resource. Therefore, we set the maximum value of adjustment value to be used in the experiment to 1.4 to manage the servers efficiently.

B. ANALYSIS OF SERVER POWER CONSUMPTION FOR CDSM BASED ON SERVER CHARACTERISTICS

As in the experiments conducted in Section V-B, CDSM does not cause less server power consumption than Static Server Management in all cases. Table 9 and Fig. 10 (a) show experimental results using Supermicro Inc SuperServer SYS-621C-TN12R as servers, assuming a total of 250 servers with 20 MB/s of traffic throughput per server. Under the same assumption, Fig. 10 (b) shows the results of an experiment using servers of Inspur Corporation Inspur NF5280M5. CDSM reduces the number of servers turned on by reducing the adjustment value when traffic is high, resulting in lower power consumption than Static Server Management as shown in Fig. 10 (b). However, in Fig. 10 (a), the power consumption is higher than that of Static Server Management when traffic is high. This seems to occur on servers where power consumption increases exponentially with CPU usage. In short, the performance of CDSM may vary depending on the server characteristics.

TABLE 8. Results of dynamic adjustment value.

		Server Free Resources		
		Average	Maximum	Minimum
Adjustment Value	1.3~2	0.6315GB	1.7029GB	0.0489GB
	1.4~2	0.8023GB	2.0407GB	0.2662GB
	1.5~2	0.9665GB	2.3785GB	0.2667GB
	1.3~1.9	0.6229GB	1.7029GB	0.0489GB
	1.4~1.9	0.7915GB	2.0407GB	0.2662GB
	1.5~1.9	0.9508GB	2.3785GB	0.2667GB
	1.3~1.8	0.6146GB	1.7029GB	0.0489GB
	1.4~1.8	0.7768GB	2.0407GB	0.2662GB
	1.5~1.8	0.9352GB	2.3785GB	0.2667GB

C. RELIABILITY ANALYSIS OF CDSM UNDER RAPID CHANGES OF TRAFFIC

When managing servers with CDSM, responding quickly to sharp increase in traffic is very important. To analyze the CDSM reliability in this case, we determined time periods when traffic changes rapidly and collected data for those periods as a test set. Fig. 11 shows the results from August 3 to August 4, 2022, a period when traffic changes rapidly

TABLE 9. Summary of discussion experimental results.

Number Of Servers	Traffic Throughput per Server	Server Management Type	Power Consumption (W)
250	20MB/s	Dynamic	77900.9605
		Static	77406.6673
		Non	105992.2027

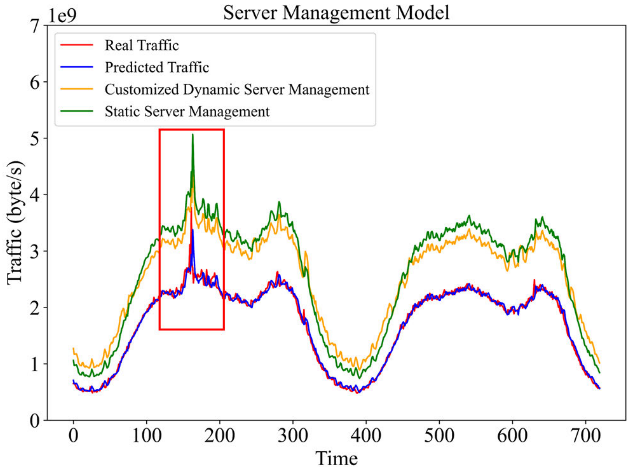


FIGURE 11. Traffic graphs of discussion experimental results.

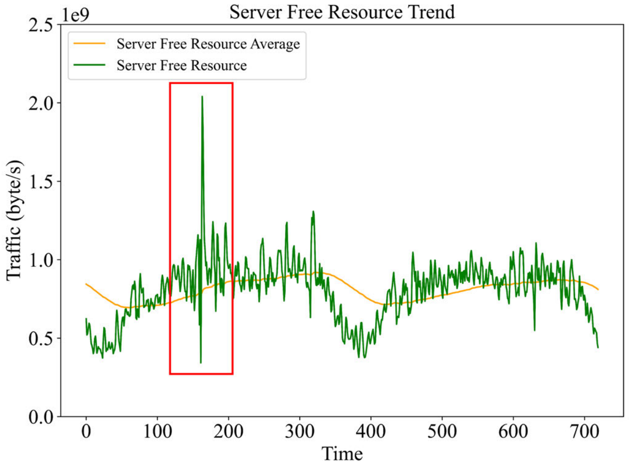


FIGURE 12. Graph of discussion experimental server free resource trend.

in the test set. In Fig. 11, traffic occurs at regular intervals and then increases sharply. Despite the surge in traffic, the deep learning-based traffic prediction model predicted the traffic accurately. Simultaneously, CDSM responded to it by quickly increasing the adjustment value. In addition, Fig. 12 shows the graph of server free resource and average server free resource from August 3 to August 4, 2022, a period when traffic changes rapidly in the test set. It shows that, under the situation where traffic suddenly spikes, the server free resource drops sharply below the average server free resource and then rises above the average server free resource. Based on this, it is confirmed that CDSM is suitable as a server management method because it can stably manage servers even when traffic changes rapidly.

VII. CONCLUSION

In this study, we proposed CDSM for efficient server management. To validate the proposed method, we collected actual transmission and reception traffic from Wikipedia's six data centers in Wikitech and used the evaluation metrics of power consumption, FLOPs, R2 score, and server free resources for objective experiments and validation. The deep learning-based traffic prediction model showed an R2 score of 0.991, and it was found that the servers were managed smoothly through Static Server Management.

However, the average and maximum values of server free resources were reduced by approximately 12% when managing servers with CDSM compared to that when managing with Static Server Management. Although the minimum value of server free resources decreased by approximately 11.8%, it increased by approximately 70% to 0.3425 when the correct average server free resource was approached after a certain period of time.

Furthermore, the power consumption was much lower when the deep learning-based traffic prediction model was used to manage the servers, and CDSM consumed less server power than that of Static Server Management. CDSM requires a deep learning model and additional program computation; however, the amount of reduction in server power consumption remains unaffected.

This shows that servers are managed more efficiently when CDSM is used. Moreover, using CDSM improves server management performance than when using Static Server Management. This can be seen through the evaluation metrics of server free resources. The evaluation metric of R2 score shows that the model has a high level of traffic prediction performance, making it an appropriate model to use for server management. Furthermore, CDSM facilitates reliable management of servers and, at the same time, efficient management of energy. As the amount of computation used in CDSM is very small, it does not affect the server power consumption. In other words, the proposed method is suitable for server management.

The proposed method showed significant performance on various evaluation metrics, demonstrating that servers can be managed reliably and efficiently. Through this study, it is expected that servers can be managed more efficiently than before in the current operating environment, while ultimately contributing to carbon neutrality by reducing energy consumption.

In future, we will use more features (e.g., external server traffic) as well as the traffic data analyzed in this study, and propose a new server management deep learning model that prevents inaccurate predictions programmatically. Moreover, we will try to analyze QoS based on latency metrics. Finally, we will use previous studies on carbon fingerprints in cloud data centers to advance our work.

REFERENCES

[1] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, I. Azevedo, and W. Lintner, ''United States data center energy usage report,'' Lawrence Berkeley Nat. Lab., Berkeley, CA, USA, Tech. Rep. LBNL-1005775, 2016.

[2] J. G. Koomey, ''Worldwide electricity used in data centers,'' *Environ. Res. Lett.*, vol. 3, no. 3, Jul. 2008, Art. no. 034008.

[3] M. A. B. Siddik, A. Shehabi, and L. Marston, ''The environmental footprint of data centers in the United States,'' *Environ. Res. Lett.*, vol. 16, no. 6, Jun. 2021, Art. no. 064017.

[4] D. Bouley, ''Estimating a data center's electrical carbon footprint,'' Schneider Electr. Library, White Paper 66, pp. 14–22, 2011.

[5] K. Le, O. Bilgir, R. Bianchini, M. Martonosi, and T. D. Nguyen, ''Managing the cost, energy consumption, and carbon footprint of internet services,'' *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 38, no. 1, pp. 357–358, Jun. 2010.

[6] M. Uddin and A. A. Rahman, ''Energy efficiency and low carbon enabler green IT framework for data centers considering green metrics,'' *Renew. Sustain. Energy Rev.*, vol. 16, no. 6, pp. 4078–4094, Aug. 2012.

[7] S. J. Davis et al., ''Net-zero emissions energy systems,'' *Science*, vol. 360, no. 6396, 2018, Art. no. eaas9793.

[8] Q. Li, ''The view of technological innovation in coal industry under the vision of carbon neutralization,'' *Int. J. Coal Sci. Technol.*, vol. 8, no. 6, pp. 1197–1207, Dec. 2021.

[9] Y. Wang, C.-H. Guo, X.-J. Chen, L.-Q. Jia, X.-N. Guo, R.-S. Chen, M.-S. Zhang, Z.-Y. Chen, and H.-D. Wang, ''Carbon peak and carbon neutrality in China: Goals, implementation path and prospects,'' *China Geol.*, vol. 4, no. 4, pp. 720–746, 2021.

[10] S. L. Gillan, A. Koch, and L. T. Starks, ''Firms and social responsibility: A review of ESG and CSR research in corporate finance,'' *J. Corporate Finance*, vol. 66, Feb. 2021, Art. no. 101889.

[11] G. Halbritter and G. Dorfleitner, ''The wages of social responsibility—Where are they? A critical review of ESG investing,'' *Rev. Financial Econ.*, vol. 26, no. 1, pp. 25–35, Sep. 2015.

[12] S. Du, C. B. Bhattacharya, and S. Sen, ''Maximizing business returns to corporate social responsibility (CSR): The role of CSR communication,'' *Int. J. Manage. Rev.*, vol. 12, no. 1, pp. 8–19, Mar. 2010.

[13] J. Judge, J. Pouchet, A. Ekbote, and S. Dixit, ''Reducing data center energy consumption,'' *Ashrae J.*, vol. 50, no. 11, pp. 14–26, 2008.

[14] H. R. Lee, H, R. Lee, and Y. J. Seo, ''A suitability analysis of IEEE 802.15.3c for data center monitoring and traffic distributing networks,'' in *Proc. Conf. Korean Inst. Commun. Inf. Sci.*, no. 1, 2014, pp. 641–642.

[15] J. S. Lee. *A Study on the Detection of Network Traffic Precursor Symptom by Modeling Change-Points [Internet]*. Accessed: Sep. 7, 2023. [Online]. Available: https://repository.hanyang.ac.kr/handle/20.500.11754/164010

[16] K. J. Lee and E. S. Kim, ''Data placement and power management for energy saving in multimedia servers,'' *J. Digital Contents Soc.*, vol. 19, no. 1, pp. 43–49, 2018.

[17] H. Y. Kim, C. H. Ham, H. K. Kwak, H. U. Kwon, Y. J. Kim, and K. S. Chung, ''A dynamic server power mode control for saving energy in a server cluster environment,'' *KIPS Trans., C*, vol. 19C, no. 2, pp. 135–144, 2012.

[18] S. G. Ma, J. H. Park, and Y. S. Seo, ''Towards carbon-neutralization: deep learning-based server management method for efficient energy operation in data centers,'' *KIPS Trans. Softw. Data Eng.*, vol. 12, no. 4, pp. 149–158, 2023.

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

[20] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, ''A comparison of ARIMA and LSTM in forecasting time series,'' in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 1394–1401.

[21] F. A. Gers, J. Schmidhuber, and F. Cummins, ''Learning to forget: Continual prediction with LSTM,'' *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

[22] X. Song, Y. Liu, L. Xue, J. Wang, J. Zhang, J. Wang, L. Jiang, and Z. Cheng, ''Time-series well performance prediction based on long short-term memory (LSTM) neural network model,'' *J. Petroleum Sci. Eng.*, vol. 186, no. 8, 2020, Art. no. 103382.

[23] A. Bharathi, R. S. Mohana, and A. Ushapriya, ''Profit and energy aware scheduling in cloud computing using task consolidation,'' in *Proc. Int. Conf. Inf. Commun. Embedded Syst. (ICICES)*, Feb. 2014, pp. 1–6.

[24] M. K. Gourisaria, S. S. Patra, and P. M. Khilar, ''Energy saving task consolidation technique in cloud centers with resource utilization threshold,'' in *Progress in Advanced Computing and Intelligent Engineering*, vol. 1. Singapore: Springer, 2018, pp. 655–666.

[25] T.-Y. Kim and S.-B. Cho, ''Traffic anomaly detection using C-LSTM neural networks,'' *Expert Syst. Appl.*, vol. 106, pp. 66–76, Sep. 2018.

[26] Y. J. Jang. *Network Prediction of Traffic Generation Amount using Time Series Prediction Model*. Accessed: Sep. 7, 2023. [Online]. Available: https://repository.hanyang.ac.kr/handle/20.500.11754/168385

[27] B. R. Park, Y. J. Choi, J. Y. Hyun, Y. R. Tae, and J. W. Moon, ''Development of optimal chilled water mass flow rate prediction and control algorithm for data center cooling energy saving,'' *J. Korea Inst. Ecol. Archit. Environ.*, vol. 21, no. 3, pp. 47–53, 2021.

[28] J. Y. Kim, H. J. Chang, and Y. H. Jung, ''Study on the development of energy conservation system for data center by utilizing multi-staged outdoor air cooling,'' in *Proc. Conf. Architectural Inst. Korea*, 2014, vol. 34, no. 2, pp. 267–268.

[29] Z. Xiao, W. Song, and Q. Chen, ''Dynamic resource allocation using virtual machines for cloud computing environment,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1107–1117, Jun. 2013.

[30] J. Praveenchandar and A. Tamilarasi, ''RETRACTED ARTICLE: Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing,'' *J. Ambient Intell. Hum. Comput.*, vol. 12, no. 3, pp. 4147–4159, Mar. 2021.

[31] *Wikitech*. Accessed: Sep. 7, 2023. [Online]. Available: https://wikitech.wikimedia.org/

[32] J. M. Johnson and T. M. Khoshgoftaar, ''Survey on deep learning with class imbalance,'' *J. Big Data*, vol. 6, no. 1, pp. 1–54, Dec. 2019.

[33] N. Japkowicz and S. Stephen, ''The class imbalance problem: A systematic study,'' *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, Jan. 2002.

[34] X. Guo, Y. Yin, C. Dong, G. Yang, and G. Zhou, ''On the class imbalance problem,'' in *Proc. 4th Int. Conf. Natural Comput.*, vol. 4, 2008, pp. 192–201.

[35] H. Cao, H. Sun, M. Sheng, Y. Shi, and J. Li, ''A QoS-guaranteed energy-efficient VM dynamic migration strategy in cloud data centers,'' in *Proc. 10th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2018, pp. 1–6.

[36] A. Beloglazov and R. Buyya, ''Energy efficient allocation of virtual machines in cloud data centers,'' in *Proc. 10th IEEE/ACM Int. Conf. Cluster, Cloud Grid Comput.*, May 2010, pp. 577–578.

[37] R. Buyya, A. Beloglazov, and J. Abawajy, ''Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges,'' 2010, *arXiv:1006.0308*.

[38] J. Benesty, J. Chen, Y. Huang, and I. Cohen, ''Pearson correlation coefficient,'' in *Noise Reduction in Speech Processing*. Berlin, Germany: Springer, 2009, pp. 1–4.

[39] T. A. Craney and J. G. Surles, ''Model-dependent variance inflation factor cutoff values,'' *Qual. Eng.*, vol. 14, no. 3, pp. 391–403, Mar. 2002.

[40] *SPECpower*. Accessed: Sep. 7, 2023. [Online]. Available: https://www.spec.org/

[41] P. Bohrer, E. N. Elnozahy, T. Keller, M. Kistler, C. Lefurgy, C. McDowell, and R. Rajamony, ''The case for power management in web servers,'' in *Power Aware Computing*, R. Graybill and R. Melhem, Eds. 2002.

[42] R. Yadav and W. Zhang, ''*MeReg*: Managing energy-SLA tradeoff for green mobile cloud computing,'' *Wireless Commun. Mobile Comput.*, vol. 2017, pp. 1–11, Dec. 2017.

[43] Q. Huang, F. Gao, R. Wang, and Z. Qi, ''Power consumption of virtual machine live migration in clouds,'' in *Proc. 3rd Int. Conf. Commun. Mobile Comput.*, Apr. 2011, pp. 122–125.

[44] L. Jin and S. Myers, ''R^2 around the world: New theory and new tests,'' *J. Financial Econ.*, vol. 79, no. 2, pp. 257–292, Feb. 2006.

[45] O. Israeli, ''A shapley-based decomposition of the R-square of a linear regression,'' *J. Econ. Inequality*, vol. 5, no. 2, pp. 199–212, Mar. 2007.

[46] R. Tang, A. Adhikari, and J. Lin, ''FLOPs as a direct optimization objective for learning sparse neural networks,'' 2018, *arXiv:1811.03060*.

[47] T.-J. Yang, Y.-H. Chen, and V. Sze, ''Designing energy-efficient convolutional neural networks using energy-aware pruning,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5687–5695.

[48] B. Paria, C.-K. Yeh, I. E. H. Yen, N. Xu, P. Ravikumar, and B. Póczos, ''Minimizing FLOPs to learn efficient sparse representations,'' 2020, *arXiv:2004.05665*.

[49] Y. Sun, N. B. Agostini, S. Dong, and D. Kaeli, ''Summarizing CPU and GPU design trends with product data,'' 2019, *arXiv:1911.11313*.

[50] R. Desislavov, F. Martínez-Plumed, and J. Hernández-Orallo, ''Compute and energy consumption trends in deep learning inference,'' 2021, *arXiv:2109.05472*.

SANG-GYUN MA is currently pursuing the B.S. degree in computer engineering with Yeungnam University, South Korea. His research interests include deep learning, machine learning, time series data analysis, and data mining.

DONG-GUN LEE received the B.S. and M.S. degrees in computer engineering from Yeungnam University, South Korea, in 2019 and 2021, respectively, where he is currently pursuing the Ph.D. degree with the Department of Computer Engineering. His research interests include software engineering, open-source software, software defect prediction, and edge computing.

YEONG-SEOK SEO (Member, IEEE) received the B.S. degree in computer science from Soongsil University, Seoul, Republic of Korea, in 2006, and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 2008 and 2012, respectively.

From September 2012 to December 2013, he was a Postdoctoral Researcher with the KAIST Institute for Information and Electronics, Daejeon. From January 2014 to August 2016, he was a Senior Researcher with the Korea Testing Laboratory (KTL), Seoul. From September 2016 to August 2022, he was an Assistant Professor (Tenure Track) with the Department of Computer Engineering, Yeungnam University, Gyeongsan, Gyeongbuk, Republic of Korea, where he has been an Associate Professor (Tenure Track), since September 2022. His research interests include software engineering, artificial intelligence, the Internet of Things, and big data analysis. Furthermore, he is involved in international standardization activities and is a member of the Korean National Body Mirror Committee to ISO on IT Service Management and IT Governance (ISO/IEC JTC1/SC40). He served as the chair for international conferences and workshops; the Proceedings Co-Chair for APSEC 2018, the Publicity Chair for WITC 2019, and the Program Chair for HCIS 2019. He also served as a technical committee member for some international conferences and workshops; ICSE 2020, ICSE 2020 Demonstrations Track, ICSE 2020 Software Engineering in Practice, MITA 2019, QRS 2020, CSA 2020, WITC 2021, FutureTech 2021, CUTE 2021, CSA 2021, WITC 2022, FutureTech 2022, CUTE 2022, CSA 2022, MITA 2021, WITC 2023, ACIIDS 2023, and ICCCI 2023.

Prof. Seo is a member of the Board of Directors of Software Engineering Society, South Korea. He was a recipient of the Undang Academic Paper Award (grand prize), in 2022, the 2nd JIPS Survey Paper Award, in 2019, and the Best Paper Award at the ASK 2022, ASK 2021, and MITA 2021 and 2019. He is an Associate Editor of *Human-Centric Computing and Information Sciences* (HCIS) (SCI indexed), *Processes* (SCI indexed), *Electronics* (SCI indexed), and *Journal of Information Processing Systems* (JIPS) (SCOPUS/ESCI indexed). He was a Guest Editor of *Journal of Systems and Software* (JSS) (SCI indexed).