Cooling Capacity Prediction for Safety-Guaranteed Optimization in IoT-Enabled Data Center

Yu Sun*, Bo Cheng*, Gaoxiang Jiang*, Yanyi Wang*, and Haibo Zhou*
*School of Electronic Science and Engineering, Nanjing University, Nanjing, China, 210023.
Emails: {yusun,bocheng}@smail.nju.edu.cn, jgxwww@sina.cn, wyynju@foxmail.com, haibozhou@nju.edu.cn.

2023 IEEE 23rd International Conference on Communication Technology (ICCT) | 979-8-3503-2595-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCT59356.2023.10419397

***Abstract*—The increasing demand for computation and storage has led to a significant rise in energy consumption and carbon emissions in data centers. Optimizing cooling systems to reduce Power Usage Effectiveness (PUE) has become a key concern due to its significant impact on the energy consumption of data centers. Thanks to the development of Internet of Things (IoT) techniques, previous studies have employed machine learning methods, which are representative data-driven techniques that are superior to traditional methods, to predict the energy consumption of the data center for assisting optimization. However, these methods have not addressed the safety issues that may arise during the optimization process. In this paper, we investigate the problem of predicting cooling capacity to ensure safety-guaranteed PUE optimization in IoT-enabled data center by leveraging machine learning. Specifically, we develop a safety-guaranteed PUE optimization framework for IoT-enabled data center, and utilize a neural network to predict the cooling capacity, ensuring safety during the optimization process. To further reduce the complexity of prediction models and improve their performance, we propose a general supervised autoencoder for dimension reduction and feature extraction. Extensive experiments demonstrate the effectiveness and superiority of the proposed neural network and supervised autoencoder over other machine learning methods.**

***Index Terms*—Internet of Things, data center, cooling capacity prediction, safety-guaranteed optimization, machine learning.**

I. Introduction

The exponential growth in demand for computation and storage is expected to drive the rapid expansion of data centers in the foreseeable future. However, the resulting massive energy consumption and carbon emissions have garnered considerable attention from both academia and industry [1]. According to estimates in [2], global energy consumption by data centers is projected to increase from 800 terawatt hours (TWh) in 2020 to 2967 TWh in 2030, leading to significant carbon emissions. Data center operators have recently taken action to reduce the negative environmental impact of data centers. For example, Google and Microsoft have pledged to achieve carbon neutrality and carbon negativity, respectively, by 2030 [3]. Improving PUE has become an important way to reduce both energy consumption and carbon emissions in data centers. Cooling energy consumption accounts for a significant proportion (46%) of total energy consumption in data centers [4] and represents a potential area for energy savings that has been the subject of recent investigations.

To optimize cooling systems, existing research has followed a two-step approach: first, a theoretical system model is built to approximate the actual system, and then optimization algorithms are adopted based on the model. Traditional modeling methods incorporate expert knowledge and are based on thermodynamic principles, heat and mass transfer processes [5], sequential modular simulation [6], and mechanical principles [7]. However, these methods are inadequate for capturing the complexity of intricate cooling systems, leading to suboptimal, unstable, and insecure optimization results [8]. In recent years, with the rapid development of the Internet of Things (IoT), multiple sensors can be utilized to obtain various and massive amounts of data from real data center. The collected data can be used to precisely describe the model of data center. Thanks to IoT techniques in the data center [9], [10], the application of precise data-driven modeling has become possible. Machine learning approaches, representative methods in the data-driven area, are a powerful tool for modeling the complex nonlinear relationships between inputs and outputs [11]. They offer an attractive alternative for achieving more accurate modeling and better optimization results in data center PUE optimization. For example, Vu et al. [12] investigated data-driven chiller plant energy optimization by tailoring different deep learning algorithms for different types of modules in module-wise modeling. Similarly, Yang et al. [13] achieved accurate PUE predictions using machine learning methods by properly selecting features. However, these approaches often neglect the safety problem of maintaining a proper temperature in server rooms during the optimization process. Therefore, to ensure data center safety during optimization and better understand the relationship between cooling system parameters and cooling capacity, we propose a neural network to predict cooling capacity, which serves as the safety boundary for the optimization process.

In this paper, we investigate cooling capacity prediction to ensure the safety of data center cooling system optimization. We first develop a safety-guaranteed PUE optimization framework for IoT-enabled data center that considers safety boundaries, specifically the required cooling capacity, which is predicted by utilizing a neural network. Additionally, We propose a general supervised autoencoder that improves performance and reduces the complexity of prediction models. Finally, we evaluate the performance of the proposed neural network and supervised autoencoder through extensive experiments on real-world datasets. The main contributions of this paper are summarized as follows:

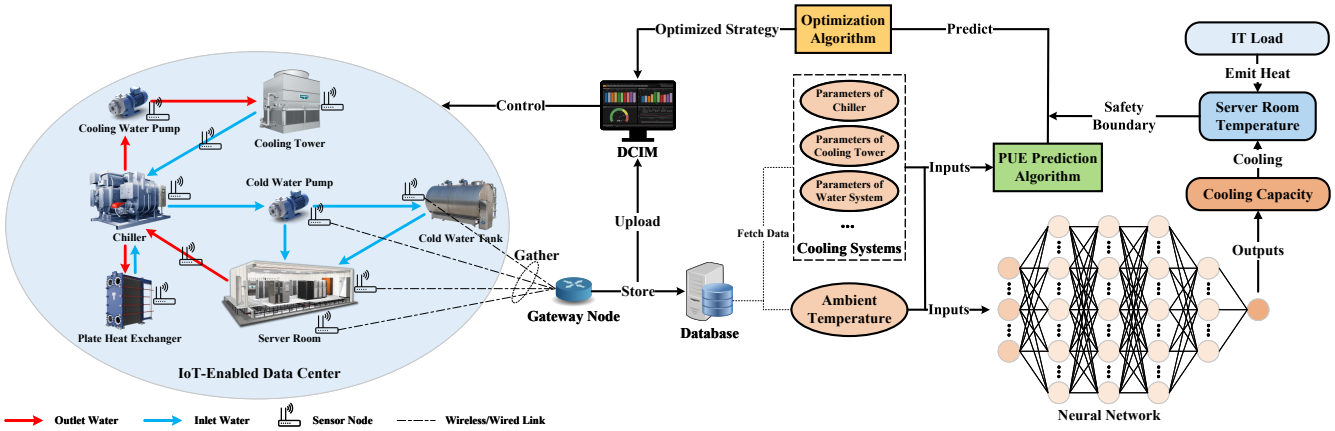

Fig. 1. A safety-guaranteed PUE optimization framework for IoT-enabled data center.

- We develop a safety-guaranteed PUE optimization framework for IoT-enabled data center that considers the required cooling capacity to ensure safe operation, which is predicted by utilizing a neural network.
- We propose a general supervised autoencoder that reduces the complexity of the prediction model and enhances prediction performance, which is applicable to a variety of prediction models.
- We demonstrate the superiority of the neural network in prediction performance and the effectiveness of the supervised autoencoder in performance improvement and general applicability through extensive simulations.

The rest of this paper is organized as follows. Section II introduces the safety-guaranteed PUE optimization framework. Sections III and IV describe the structure of the neural network and the general supervised autoencoder, respectively. The experimental setup and results are presented in Section V. Finally, Section VI concludes this paper.

II. System Model

The IoT-enabled data center and the safety-guaranteed PUE optimization framework are illustrated in Fig. 1. The left side of Fig. 1 shows the interaction between the cooling system and server rooms. The cooling system emits the heat generated by servers and provides cooling capacity with the cooperation of the chiller, plate heat exchanger, cold water pump, cold water tank, cooling water pump, and cooling tower. By deploying multiple and various sensor nodes at these equipment, such as temperature sensors, humidity sensors, and flowmeter sensors, the global state of the data center can be monitored. The sensed data is gathered and processed by the gateway, and then uploaded to the Data Center Infrastructure Management (DCIM) and stored in a database. DCIM receives the optimized strategies from the safety-guaranteed PUE optimization framework and then sends the control command to the cooling system. As shown in the left and right sides of Figure 1, IoT devices act as a bridge between the physical data center and the digital optimization framework.

The optimization of PUE in data center can be formulated as a general optimization problem, as shown below:

$$\min_{\mathbf{x}} \ \text{PUE} \tag{1a}$$

$$\text{s.t. } C_i(\mathbf{x}, \mathbf{y}) \le 0, \ \forall i \in \{1, 2, \cdots, I\}, \tag{1b}$$

$$S_j(\mathbf{x}, \mathbf{y}) \le 0, \ \forall j \in \{1, 2, \cdots, J\}, \tag{1c}$$

$$T_k^l \le T_k(\mathbf{x}, \mathbf{y}) \le T_k^u, \ \forall k \in \{1, 2, \cdots, K\}, \tag{1d}$$

where $\mathbf{x} \in \mathbb{R}^m$ is the vector of controllable parameters of cooling system devices, while $\mathbf{y} \in \mathbb{R}^n$ is the vector of uncontrollable parameters, including the environmental parameters and uncontrollable parameters of devices. The function C_i represents the physical constraints on the devices, and S_j represents the safety constraints on the devices. The temperature of the k-th server room is denoted as T_k, and T_k^l and T_k^u are the lower and upper bounds of the temperature, respectively.

To ensure the safe operation of data center, it is necessary to satisfy the constraints (1b)-(1d) while optimizing the PUE. However, previous works have neglected the constraints (1d). Therefore, we propose a safety-guaranteed PUE optimization framework that optimizes the PUE while satisfying all the constraints (1b)-(1d) as depicted in the right side of Figure 1. PUE can be predicted by collecting parameters of the cooling system, and then optimized by adjusting the parameters based on the built model. However, decreasing the PUE may result in a higher temperature that could exceed the thermal redlining. To guarantee the safe running of data center, a safety boundary, i.e., the temperature redlining of the server room, is introduced. As the IT load increases, the temperature of the server room rises, requiring more cooling capacity to cool it down. In the following sections, we investigate the prediction of cooling capacity by a neural network to determine how to meet the demand for cooling capacity by adjusting the parameters of the cooler at a given ambient temperature.

III. Cooling Capacity Prediction

Neural networks are capable of modeling complex nonlinear systems and have been widely applied in many fields. In this section, we propose a neural network model for predicting cooling capacity.

The input of the neural network is composed of the parameters of the cooling system, denoted as $\mathbf{x} \in \mathbb{R}^{d_x}$, while the cooling capacity is considered as the output, denoted as y. The neural network is trained on the training set $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$, where $\mathbf{x}_n$ and y_n are the input and output of the n-th sample, respectively. The structure of the proposed neural network is shown in Fig. 1, where the input layer collects the parameters from various devices in the cooling system, the output layer predicts the cooling capacity, and several hidden layers are between the input and output layers. The l-th layer of the neural network can be expressed as:

$$\mathbf{h}^{(l+1)} = f_\sigma^{(l)}\left(\mathrm{BN}^{(l)}_{\gamma^{(l)},\beta^{(l)}}\left(\mathbf{W}^{(l)}\mathbf{h}^{(l)} + \mathbf{b}^{(l)}\right)\right), \tag{2}$$

where $\mathbf{h}^{(l)} \in \mathbb{R}^{d^{(l)}}$ and $\mathbf{h}^{(l+1)} \in \mathbb{R}^{d^{(l+1)}}$ are the input and output of the l-th layer, respectively. $\mathbf{W}^{(l)} \in \mathbb{R}^{d^{(l+1)}\times d^{(l)}}$ is the weight matrix of the l-th layer, and $\mathbf{b}^{(l)} \in \mathbb{R}^{d^{(l+1)}}$ is the bias of the l-th layer.

The batch normalization operation $\mathrm{BN}^{(l)}_{\gamma^{(l)},\beta^{(l)}}(x)$ is used to accelerate the training process and improve the network performance by normalizing the data over a mini-batch, which is defined as:

$$\mathrm{BN}^{(l)}_{\gamma^{(l)},\beta^{(l)}}(x_i) = \gamma^{(l)}\frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \beta^{(l)}, \tag{3}$$

where x_i is the i-th sample in a mini-batch, ϵ is a constant for numerical stability, and $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}^2$ are the mean and variance of the mini-batch, respectively. $\gamma^{(l)}$ and $\beta^{(l)}$ are the scale factor and shift factor, respectively, which are learnable parameters of the batch normalization operation.

$f_\sigma^{(l)}(x)$ is an element-wise operation for vector, and in this neural network, we adopt the ReLU function as the activation function, which is defined as:

$$f_\sigma^{(l)}(x) = \mathrm{ReLU}(x) = \max(0, x). \tag{4}$$

The mean squared error (MSE) function is used as the loss function in the training stage, which can be expressed as:

$$\mathcal{L}(\Theta) = \frac{1}{N}\sum_{n=1}^{N}\|y_n - \hat{y}_n\|_2^2, \tag{5}$$

where $\Theta = \left\{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}, \gamma^{(l)}, \beta^{(l)}, l = 0, 1, \cdots, L\right\}$ is the set of all parameters of the neural network, which are updated by the backpropagation algorithm during the training stage.

IV. General Supervised Autoencoder

The neural network introduced in Section III may become outdated due to the continuous accumulation of data in data center. However, training a new neural network can be both computation-intensive and time-consuming. Furthermore, multiple machine learning models are employed in data center for the purpose of providing redundant backup and accommodating various scenarios, despite the fact that some models may not perform satisfactorily. Motivated by the aforementioned concerns, we propose a general supervised autoencoder to learn the feature representation of the original data. Our approach not only reduces the complexity of the prediction model by compressing the input dimension, but also enhances prediction performance by extracting key features and eliminating unwarranted factors. Furthermore, our method is applicable to various prediction models.

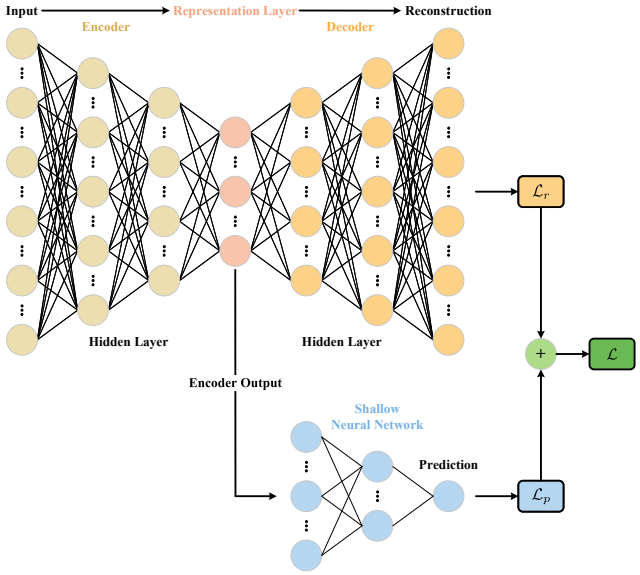

Fig. 2. Network structure of the supervised autoencoder.

The network structure of the proposed supervised autoencoder is shown in Fig. 2, which includes an encoder, a decoder, and a shallow neural network. The input of the supervised autoencoder is $\mathbf{x} \in \mathbb{R}^{d_x}$, which comprises the parameters of cooling systems. The output is the predicted cooling capacity y and the reconstructed input $\hat{\mathbf{x}}$. The encoder encodes the input as the feature representation $\mathbf{z} \in \mathbb{R}^{d_z}$, and the decoder reconstructs the original input $\hat{\mathbf{x}}$ from the feature representation $\mathbf{z}$. The shallow neural network predicts the cooling capacity y from the feature representation $\mathbf{z}$.

*1) **Encoder**:* The encoder is a neural network that performs feature extraction and dimension reduction. The input is original data $\mathbf{x}$, and the output $\mathbf{z}$ is the learned representation of the original data. Therefore, the output layer is also named the representation layer. $\mathbf{z}$ is a lower-dimensional representation of the original features. The encoder can be described as:

$$\mathbf{h}_e^{(0)} = \mathbf{x}, \tag{6a}$$

$$\mathbf{h}_e^{(l+1)} = f_\sigma^{(l)}\left(\mathbf{W}^{(l)}\mathbf{h}_e^{(l)} + \mathbf{b}^{(l)}\right),\ l = 0, 1, \cdots, L_e, \tag{6b}$$

$$\mathbf{z} = \mathbf{h}_e^{(L_e+1)}, \tag{6c}$$

where $\mathbf{h}_e^{(l)}$ represents the output of the l-th layer, while $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ denote the weight matrix and bias vector of the l-th layer, respectively. L_e indicates the number of layers in the encoder. The ReLU function is used as the activation function $f_\sigma^{(l)}(x)$ in the encoder. Due to the compression characteristic of the encoder, the dimension of the representation is typically lower than that of the input, that is, $d_z < d_x$.

*2) **Decoder**:* The decoder is a neural network that performs the reconstruction from the representation. The input layer is the representation layer $\mathbf{z}$ of the encoder, and the output $\hat{\mathbf{x}}$ is

the reconstructed original data. The structure of the decoder is symmetric to that of the encoder, with mirror symmetry about the representation layer, which can be expressed as:

$$\mathbf{h}_d^{(0)} = \mathbf{z}, \tag{7a}$$

$$\mathbf{h}_d^{(l+1)} = g_\sigma^{(l)}\left(\mathbf{W}^{(l)}\mathbf{h}_d^{(l)} + \mathbf{b}^{(l)}\right),\ l = 0, 1, \cdots, L_d, \tag{7b}$$

$$\hat{\mathbf{x}} = \mathbf{h}_d^{(L_d+1)}, \tag{7c}$$

where the symbols used in the decoder are the same as those in the encoder, except that L_d denotes the number of layers in the decoder and $d_z < d_{\hat{x}}$. Additionally, the activation function $g_\sigma^{(l)}(x)$ in the last layer is a Sigmoid function:

$$g_\sigma^{(l)}(x) = \begin{cases} \max\{0, x\}, & l = 0, 1, \cdots, L_d - 1, \\ 1/(1 + e^{-x}), & l = L_d. \end{cases} \tag{8}$$

*3) **Shallow Neural Network**:* The shallow neural network predicts the cooling capacity using the features extracted by the encoder. The input layer of the shallow neural network is the representation layer $\mathbf{z}$ of the encoder, while the output layer y represents the predicted cooling capacity. The structure of the shallow neural network is similar to the model described in Section III. However, due to the extracted features and reduced dimensionality of the input layer, the shallow neural network is much simpler and has lower complexity compared to the original neural network.

*4) **Supervised Learning**:* A self-supervised autoencoder evaluates the effectiveness of the encoder and decoder using the reconstruction loss criterion, which is based on the MSE function and can be expressed as:

$$\mathcal{L}_r = \sum_{n=1}^{N} \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|_2^2. \tag{9}$$

The supervised autoencoder, as shown in Fig. 2, is an autoencoder with an additional supervised loss. The supervised loss $\mathcal{L}_s$ evaluates the prediction performance of the shallow neural network from the representation layer. By combining both the reconstruction loss and the supervised loss, the supervised autoencoder's loss function is defined as:

$$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_r = \frac{1}{N}\sum_{n=1}^{N}\left(\|y_n - \hat{y}_n\|_2^2 + \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|_2^2\right). \tag{10}$$

Simultaneously carrying out representation learning and supervised learning, the supervised autoencoder can guide the autoencoder towards a more effective representation for the desired task, leading to better performance compared to pure representation learning. Conversely, relying solely on supervised learning, like the neural network in Section III, can only fit the data well, but it does not generalize well and cannot find the underlying structure of the data [14].

V. Experiments

In this section, we compare the proposed approaches with other methods on a real-world dataset to evaluate their effectiveness. We begin by introducing the dataset and describing the preprocessing steps. Then, we provide details on the experimental setup and the baseline methods used. Finally, we present and analyze the experimental results.

A. Data Preprocessing

The data are collected from a data center located in Jinan, Shandong province, China. Data are derived from various IoT devices installed in the data center, such as temperature and humidity sensors and flowmeter sensors. A portion of the data is presented in Table I, where cooling capacity is the label of the data, and the other 319 parameters are the input features. The data are collected from November 5, 2022, at 08:39 to December 8, 2022, at 04:43, and the sampling frequency is either one or two minutes, resulting in 40,786 samples.

TABLE I
Collected Raw Data from Real-World Data Center

Devices	Parameters	Unit
Chillers	Inlet Temperature of Chilled Water	°C
	Outlet Pressure of Chilled Water	kPa
	.....	...
	Inlet Temperature of Cooling Water	°C
Cooling Towers	Water Tray Temperature	°C
	Fan Frequency Feedback	Hz
	.....	...
Chilled Water Pump	Inlet Pressure of the Water Pump	kPa
.....	.....	...
Water System	Cooling Capacity	kJ

We perform pre-processing on the raw data to improve the quality and usability of the data, to reduce errors and biases in data analysis and modeling. The input feature and output label are normalized using Min-Max normalization to the range of $[0, 1]$:

$$x = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \tag{11}$$

We also remove single-valued features, also known as zero variance features, from the dataset to reduce calculation in the training process. This is done because uninformative features may have negligible effects on the prediction. To mitigate the negative effects of inconsistent data on prediction, we employed an outlier mining method, the box-whisker plot [15], to detect outliers. This method is a robust method that does not require data to follow a specific statistical distribution.

After data preprocessing, the number of input features is reduced to 222. The data are randomly split into training and test sets with a ratio of 8:2.

B. Experiment Setups

We design a neural network for our experiments with 5 hidden layers, and each of these layers contains 256, 128, 64, 32, and 1 neurons, respectively. The batch size is set as 512, and the learning rate is fixed at 0.0003. To train the neural network, we use the Adam optimizer. We evaluate

our proposed model against the following machine learning algorithms[1]:

- Bayesian Regression.
- Decision Tree: The maximum depth of the tree is set as 7, the minimum samples split is set as 2, and the minimum samples leaf is set as 1.
- Random Forest: The number of trees set as 100, and the set of trees is identical to the Decision Tree algorithm.
- Support Vector Regression (SVR): The kernel function is set as a radial basis function (RBF).
- K-Nearest Neighbors (KNN): The number of neighbors is set as 15, and the metric is set as Manhattan distance.

For the supervised autoencoder, we set the number of hidden layers of the encoder as 4, and the number of neurons in each of these layers as 256, 128, 64, and 32. The decoder part is symmetric to the encoder part. The shallow neural network is a single-layer neural network with 32 neurons. We use the Adam optimizer to train the neural network and set the learning rate and batch size as 0.0003 and 512, respectively. To compare our supervised autoencoder with other methods, we use the following algorithms[2]:

- Principle Component Analysis (PCA): We adopt the RBF kernel to enable the nonlinear reduction of PCA.
- Uniform Manifold Approximation and Projection (UMAP) [16].
- Autoencoder (AE): The same as the proposed supervised autoencoder, but without the supervised part.
- Deep Belief Network (DBN) [17]: The structure of the network is the same as that of the AE, with each layer consisting of restricted Boltzmann machines.

We use the following three metrics to evaluate the prediction performance of these algorithms:

1) Root mean square error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}. \tag{12}$$

2) Mean absolute error (MAE):

$$\mathrm{MAE} = \frac{1}{T}\sum_{t=1}^{T}|\hat{y}_t - y_t|. \tag{13}$$

3) Mean absolute percentage error (MAPE):

$$\mathrm{MAPE} = \frac{1}{T}\sum_{t=1}^{T}\frac{|\hat{y}_t - y_t|}{y_t} \times 100\%. \tag{14}$$

C. Experiment Results

*1) **Prediction Performance**:* Figure 3 illustrates the training and validation loss of the proposed neural network during the training stage, where the neural network converges quickly in only 50 epochs. To test the prediction performance of the proposed neural network, we selected 200 samples from the validation set, as shown in Figure 4. The results demonstrate that the predicted cooling capacity is close to the actual value, and the error is acceptable.

We compared the proposed neural network with other machine learning algorithms in terms of RMSE, MAE, and MAPE, and present the results in Table II (No Compression terms listed at the bottom of the table). The comparison indicates that the proposed neural network outperforms all other algorithms. Specifically, our neural network achieves a 3.6% lower RMSE, a 3.9% lower MAE, and a 3.7% lower MAPE than the next best algorithm. Moreover, the margin can be up to 17.9%, 16.2%, and 17.6%, respectively, compared with the worst algorithm, indicating that the proposed neural network has superior prediction performance compared to other algorithms.

TABLE II
PERFORMANCE COMPARISON OF FEATURE EXTRACTION ALGORITHMS

Prediction Model	**Bayesian Regression**			**Decision Tree**		
Algorithm	**RMSE**	**MAE**	**MAPE**	**RMSE**	**MAE**	**MAPE**
PCA	9.60%	7.47%	18.00%	9.46%	7.34%	17.68%
UMAP	10.72%	8.42%	20.63%	9.71%	7.52%	18.15%
DBN	9.87%	7.66%	18.44%	9.56%	7.42%	17.89%
AE	9.78%	7.62%	18.42%	9.53%	7.39%	17.86%
Supervised AE	**6.69**%	**5.23**%	**12.44**%	**7.19**%	**5.63**%	**13.51**%
No Compression	6.88%	5.38%	12.81%	7.36%	5.71%	13.54%
Prediction Model	**Random Forest**			**SVR**		
Algorithm	**RMSE**	**MAE**	**MAPE**	**RMSE**	**MAE**	**MAPE**
PCA	9.24%	7.16%	17.29%	8.85%	6.85%	16.59%
UMAP	9.42%	7.33%	17.71%	9.49%	7.37%	17.80%
DBN	9.50%	7.38%	17.86%	9.88%	7.68%	18.45%
AE	9.40%	7.28%	17.63%	8.91%	6.94%	16.85%
Supervised AE	**6.93**%	**5.42**%	**13.06**%	**6.71**%	**5.24**%	**12.54**%
No Compression	7.03%	5.46%	13.01%	7.03%	5.50%	13.28%
Prediction Model	**KNN**			**Neural Network**		
Algorithm	**RMSE**	**MAE**	**MAPE**	**RMSE**	**MAE**	**MAPE**
PCA	8.98%	6.87%	16.62%	9.60%	7.47%	18.02%
UMAP	9.55%	7.33%	17.70%	10.63%	8.32%	20.33%
DBN	9.61%	7.42%	17.91%	9.88%	7.69%	18.62%
AE	9.19%	7.08%	17.15%	9.77%	7.61%	18.43%
Supervised AE	**7.00**%	**5.46**%	**13.12**%	6.69%	5.23%	12.46%
No Compression	8.08%	6.17%	14.97%	**6.63**%	**5.17**%	**12.33**%

*2) **Prediction Performance with Feature Extraction**:* Multiple feature extraction algorithms are used to compress the input into 32-dimensional features, which are then utilized by multiple prediction algorithms to predict the cooling capacity. A comprehensive comparison of all combinations of multiple feature extraction algorithms and multiple prediction algorithms is presented in Table II.

PCA, UMAP, DBN, and AE can reduce the complexity of

[1]Due to the high-dimensional input, polynomial regression is unsuitable for our experiments.

[2]Although there are some supervised dimensionality reduction methods, such as Linear Discriminant Analysis (LDA) and Locally Linear Embedding (LLE), they are only suitable for classification problems. In this paper, we choose the DBN as our supervised baseline.

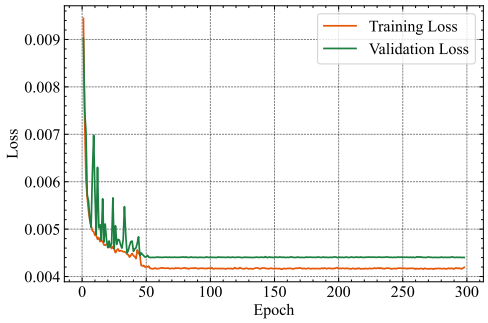

Fig. 3. Training and validation loss curve of proposed neural network.

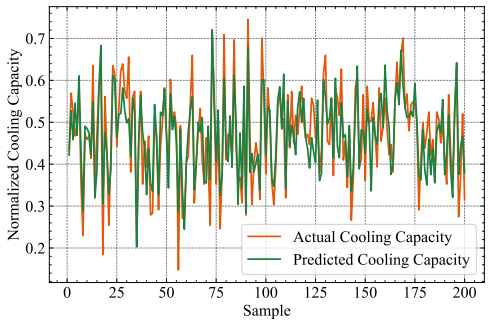

Fig. 4. Prediction results of proposed neural network.

the prediction model by reducing the input dimension, albeit with a slight performance loss. However, our proposed supervised autoencoder improves the performance of prediction models in addition to reducing complexity by extracting key features and eliminating unwarranted factors. For example, compared to KNN with no compression, supervised autoencoder achieves a 13.4%, 11.5%, and 12.4% improvement on RMSE, MAE, and MAPE, respectively, for KNN. The results indicate that the general supervised autoencoder can extract useful features and remove noise from the original data, resulting in an improvement in the performance of prediction models, while other feature extraction algorithms cannot achieve this and may lead to decreased performance. Furthermore, we observe that the neural network outperforms other prediction models in all cases where they are combined with the supervised autoencoder, demonstrating its superiority in prediction. We observed that supervised autoencoder cannot enhance the performance of the neural network, which already has the great ability to extract key features. However, the performance of the neural network with supervised autoencoder is similar to that of the neural network with no compression, with a performance loss of less than 1.2%, which is negligible. Therefore, supervised autoencoder is still a good choice for the neural network, as it reduces complexity. These results also demonstrate the general applicability of supervised autoencoder across different prediction algorithms.

VI. Conclusion

In this paper, we have investigated the problem of predicting cooling capacity in IoT-enabled data center while ensuring safe operation. We have developed a safety-guaranteed power usage effectiveness optimization framework for IoT-enabled data center and utilized a neural network to predict cooling capacity, which is crucial for ensuring safe operation. To reduce model complexity and improve performance, we have proposed a general supervised autoencoder for dimension reduction and feature extraction. Extensive experiments have demonstrated the superiority of the neural network in capturing underlying patterns and reducing prediction errors, as well as the benefits of our proposed supervised autoencoder in performance improvement and general applicability. In the future, we plan to validate the effectiveness of our proposed methods from multiple perspectives and further improve them.

References

[1] J. Zhao, Q. Yu, B. Qian, K. Yu, Y. Xu, H. Zhou, and X. Shen, "Fully-Decoupled Radio Access Networks: A Resilient Uplink Base Stations Cooperative Reception Framework," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023.

[2] A. S. G. Andrae and T. Edler, "On Global Electricity Usage of Communication Technology: Trends to 2030," *Challenges*, vol. 6, no. 1, pp. 117–157, Jun. 2015.

[3] Z. Cao, X. Zhou, H. Hu, Z. Wang, and Y. Wen, "Toward a Systematic Survey for Carbon Neutral Data Centers," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 895–936, 2022.

[4] J. Ni and X. Bai, "A review of air conditioning energy performance in data centers," *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 625–640, Jan. 2017.

[5] Z. Ma and S. Wang, "An optimal control strategy for complex building central chilled water systems for practical and real-time applications," *Building and Environment*, vol. 44, no. 6, pp. 1188–1198, Jun. 2009.

[6] J. Sun and A. Reddy, "Optimal control of building HVAC&R systems using complete simulation-based sequential quadratic programming (CSB-SQP)," *Building and Environment*, vol. 40, no. 5, pp. 657–669, May 2005.

[7] L. Lu, W. Cai, L. Xie, S. Li, and Y. C. Soh, "HVAC system optimization—in-building section," *Energy and Buildings*, vol. 37, no. 1, pp. 11–22, Jan. 2005.

[8] C. A. Balaras, J. Lelekis, E. G. Dascalaki, and D. Atsidaftis, "High Performance Data Centers and Energy Efficiency Potential in Greece," *Procedia Environmental Sciences*, vol. 38, pp. 107–114, Jan. 2017.

[9] M. K. Jackson Ramphela, P. A. Owolawi, T. Mapayi, and G. Aiyetoro, "Internet of Things (IoT) Integrated Data Center Infrastructure Monitoring System," in *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, Aug. 2020, pp. 1–6.

[10] A. Medina-Santiago, A. D. P. Azucena, J. M. Gómez-Zea, J. A. Jesús-Magaña, M. de la Luz Valdez-Ramos, E. Sosa-Silva, and F. Falcón-Pérez, "Adaptive Model IoT for Monitoring in Data Centers," *IEEE Access*, vol. 8, pp. 5622–5634, 2020.

[11] J. Xue, T. Zhang, W. Wu, H. Zhou, and X. Shen, "Sparse Big Data for Vehicular Network Traffic Flow Estimation: A Machine Learning Approach," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, Dec. 2022, pp. 4959–4963.

[12] H. D. Vu, K. S. Chai, B. Keating, N. Tursynbek, B. Xu, K. Yang, X. Yang, and Z. Zhang, "Data Driven Chiller Plant Energy Optimization with Domain Knowledge," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM '17. New York, NY, USA: Association for Computing Machinery, Nov. 2017, pp. 1309–1317.

[13] Z. Yang, J. Du, Y. Lin, Z. Du, L. Xia, Q. Zhao, and X. Guan, "Increasing the energy efficiency of a data center based on machine learning," *Journal of Industrial Ecology*, vol. 26, no. 1, pp. 323–335, 2022.

[14] L. Le, A. Patterson, and M. White, "Supervised autoencoders: Improving generalization performance with unsupervised regularizers," in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018.

[15] J. W. Tukey, *Exploratory Data Analysis*. Reading, MA, 1977, vol. 2.

[16] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," Sep. 2020.

[17] G. E. Hinton, "Deep belief networks," *Scholarpedia*, vol. 4, no. 5, p. 5947, May 2009.