A Combined Neural and Genetic Algorithm Model for Data Center Temperature Control

Weiping Yu [1], Zhaoguo Wang[1], Yibo Xue[1], Lingxu Guo[2] and Liyuan Xu[1]

[1] Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing, China
[2] State Grid Tianjin Electric Power Company, Hebei District, State Grid, Tianjin, China
yiboxue@tsinghua.edu.cn

Abstract. As many parameters of the cooling system in data center mainly depend on manual control, it is urgent to fix the problem of simulation and optimization at the same time and develop effective mitigation methods. In this paper, we proposes a combined method based on machine learning. Firstly, we have built the neural network with the temperature and humidity of the equipment in real data, then established evaluation model, and finally the optimal setting has been obtained by a genetic algorithm. Our results show that that the effect is better than artificial method and traditional greedy algorithms with 3% - 15% relative reduction in errors of temperature and humidity per set of parameters.

Keywords: Machine Learning, Combined Algorithm, Data Center, System Control.

1 Introduction

Data center is a facility that consumes a lot of energy and there is a strict standard of environmental factors. For example, if the temperature is too high, the life of the component will be reduced, and at the same time, the discharge capacity of the battery will be lowered if the temperature is too low. Early physicists tried to solve this problem by analyzing the structure of the cooling system and combining the formulas of fluid mechanics [1]. But in the complex and varied actual situation, it is easy to ignore some minor factors, and the modeling process requires a lot of professional knowledge and a huge amount of engineering work.

At present, the methods of controlling temperature and humidity in a data center are mainly focused on load balancing or changing the topology of equipment and air conditioning through Computational Fluid Dynamics(CFD) to simulate [2]. But the actual research found that many super-large computer room cooling parameters are not as simple as imagined. For example, indoor air conditioning system is usually controlled by cooling units directly, and cooling units and air conditioning system is not a one-to-one correspondence. This makes it difficult to simulate by CFD, because it unable to directly set the required parameters, so it is difficult to separate the corresponding relationship.

In recent years, deep neural networks have been widely applied in various fields. In particular, Google's approach to reducing Power Usage Effectiveness(PUE) through neural networks in 2014 [3] inspired us. Neural networks have an advantage in modeling complex systems, because neural networks do not require a user to preset the interaction characteristics of models, but look for patterns and interactions in features by itself, and then automatically generate the best matching model. Moreover, its performance will improve with the increase of data volume.

After getting an accurate simulation model, we need to get an optimal combination of parameters to make the temperature and humidity meet the standard. It is usually more convenient and accurate to use the machine learning algorithm to calculate the optimal solution. The advantage of the genetic algorithm is that it can deal with constraints very well, and jump out of local optimum easily, and finally get the global optimum solution. That is, the global search ability is strong. For the setting of the fitness function, we set up a hierarchical and segmented evaluation function on the basis of the above neural network model in order to take better care of the values with further distance with standards.

In this paper, we propose a combined machine learning model to overcome the shortcomings of a single algorithm or manual control of cooling parameters in data centers. The paper is mainly divided into the following parts: Section 3.1 describes how to use the neural network to simulate the actual production environment; Section 3.2 explains the method of increasing the weight of Abnormal values by layered function; Section 3.3 explains how the genetic algorithm is applied to the calculation of optimal solution; Section 4 describes some formulas and realizations in the implementation of the algorithm. Finally, the Section 5 will summarize the paper and explain the future work.

2 Related Work

2.1 Neural Network

To cope with these problems, the neural network is chosen as the framework of a data center model. The neural network is a machine learning algorithm which imitates the interaction between neurons in cognition. Like most learning systems, the accuracy of the model increases with the addition of new training data at each iteration. Google is a pioneer in intelligent data centers. They have used a fully-connected neural network to simulate data center energy consumption [3]: 19 input parameters were manually selected, and PUE was predicted by a 5-layer fully-connected neural network with an accuracy rate of 99.6%.

Our decision to optimize temperature and humidity comes from [13], in which they also use neural network for real-time calculations of the air properties required in drying of agricultural and food materials.

2.2 Genetic Algorithm

Google uses neural networks to get simulated data from data centers, and uses control variables to predict the label trend when one or two variables change simultaneously.

Although the method similar to the greedy algorithm can reduce PUE directly, it is easy to fall into local optimal solution and the analytical ability when sets of variables change simultaneously is weak.

In order to solve this problem, the genetic algorithm is adopted in this paper. Genetic algorithm is based on the evolution process of nature: selection, crossover and mutation, through the fitness function to evaluate the possibility of individual survival. The Genetic algorithm is widely used in finite element problems, such as in the green cloud memory allocation strategy, compared with the traditional SLC configuration, the maximum memory usage based on genetic algorithm can be reduced by 76.8% [4].

At the same time, due to the controllability of genetic algorithm, we adopt an adaptive method: by artificially restricting the survival probability of operators [7], we can quickly generate high-fitness operators and improve the efficiency of the algorithm.

2.3 Model Combination

In the application of actual scenes, a single model often cannot solve the problem of complex and multi-constraint conditions [5]. However, it is not a good idea to directly use the model combination of simulation & optimal solution as well. For example, in our dataset, assuming that there are two temperature data, one is closer to the standard and the other is farther. Finding the optimal solution may make the better data better, but the worse data changes less. So we have to change the weight of different data to fuse the two models. In fact, there are examples of using custom evaluation functions to achieve better optimization results in engineering: A custom approach to multi-target optimization of expensive-to-evaluate functions is explored that is based on a combined application of Gaussian processes, mutual information and a genetic algorithm[10].

3 the Proposed Approach

3.1 Neural Network Simulation of Data Center Refrigeration

In the experiment, our dataset contains more than 300 dimension numerical or non-numerical features. Referring to the method of [7], we generated some features ourselves, such as the backwater temperature of the cooler. Set it to - 1 when the temperature is higher than the previous moment, indicating that the cooler is in a relatively passive maintenance state; set it to 1 or 0 when the temperature is lower than or equal to the previous moment, indicating that the cooler is in a relatively positive cooling state. Considering the reason of saving computing resources and improving accuracy, we use principal component analysis (PCA) to get the first 100 dimension features of importance. Including chiller water supply temperature, chiller backwater temperature, external environment temperature, external environment wind speed, IT load, air conditioning power, etc. After that, 336 dimensional tags are selected as output including 168 temperature points and 168 humidity points. The proportion of training set and test set is 5:1. The experimental results of the neural network and decision tree (random forest) with different structure sizes are shown in Fig 1, from which we can see the reason why we the chose neural network.

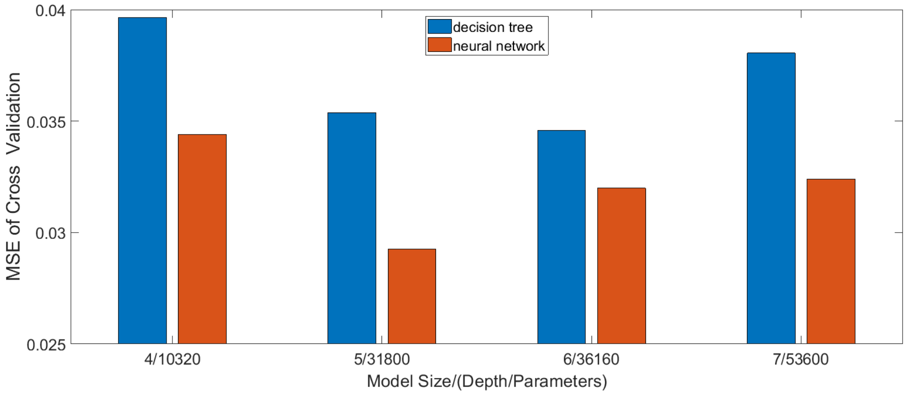

Fig. 1. Comparison of decision tree and neural network in different model sizes.

In addition, it should be noted that the standardization and normalization of data are also very important. In our case, the difference between the order of magnitude of the data items is very large. In this case, the standardization and normalization of data can speed up the convergence rate, reduce the calculation cost and improve the accuracy rate. In this article, we have mainly adopted the following methods:

The first step is to normalize the data to [-1,1]:

$$x_{norm} = \frac{x - MEAN(x)}{MAX(x) - MIN(x)} \tag{1}$$

Secondly, the batch normalization layer is added after the input layer, and the following formula is used[6]:

$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$$
$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i) \tag{2}$$

Where μ_B is the mean and σ_B^2 is variance. γ is a scaling variable and β is a translation variable. In the training process, the variance and mean are calculated by training input data, and γ and β are the parameters that need to be learned. This ensures that every time the data is normalized, it still retains the trained features.

At the same time, it can complete the normalization operation and accelerate the training. At each level, we also add the L2 regularization term. The formula is as follows:

$$c = c_0 + \frac{\lambda}{2n} \sum_{\omega} \omega^2 \tag{3}$$

Where c_0 is the original output term, ω is the product parameter of all neural networks, λ is the normal term coefficient, and n is the sample size of the training set. The addition of regular term will reduce the weight of, reduce the complexity of the network and prevent over-fitting.

Finally, in order to solve the problem of inadequate expression of the linear model, we add activation function to make neural network can deal with complex data better. In the middle hidden layer, we use the PReLU function[8]:

$$PReLU(x_i) = \begin{cases} x_i & if x_i > 0 \\ a_i x_i & if x_i \le 0 \end{cases} \tag{4}$$

The reason is that it has a faster convergence speed than tanh/sigmoid and does not appear the problem of gradient disappearance. At the same time, when x is less than 0, a smaller gradient (a_i) is used to replace 0, which can solve the problem that the excessive gradient stops activation after passing through the ordinary ReLU unit.
After the last level of output layer, we adopted *tanh*:

$$tanh(x) = \frac{1-e^{-2x}}{1+e^{2x}} \tag{5}$$

Because the output range is [-1,1], it is essentially 0-means, which corresponds to our previous normalized data method.

Based on the above theoretical analysis and experimental results, we select 100 parameters as input, 336 labels (168 temperature and 168 humidity measurement points) as output, including five hidden fully-connected layers, each containing 50 nodes. Our final network structure is as follows:

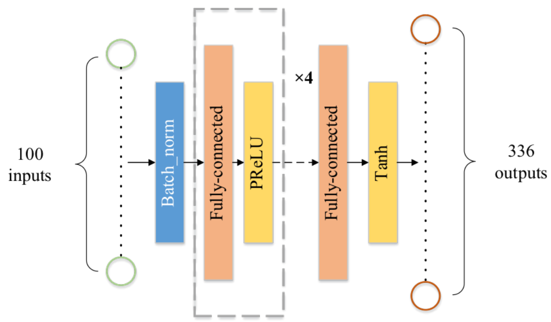

Fig. 2. A figure caption is always placed below the illustration. Short captions are centered, while long ones are justified. The macro button chooses the correct format automatically.

3.2 Hierarchical Evaluate Function

We obtained the standards of temperature and humidity of each monitoring point from long-term experience and expert markers, and found that if use mean-square error(MSE) as the evaluate function of the genetic network directly, it could not be very good to optimize the poor performance measurement points. And in the actual production situation, we pay more attention to the data far from the standard value, so we propose a hierarchical evaluate function to optimize the combination model.

The hierarchies mainly include two meanings, one is the numerical hierarchies of different distances from the standard value, the other is the weight distribution of different monitoring points. Firstly, the algorithm for different distance thresholds: There is a manually calibrated standard threshold set. Each standard of threshold has a pair of upper and lower limits. The *internal* is the length of the threshold internal in which the value (temperature or humidity) located. The *abs_value* is the absolute value of distance from value to the inner standard.; $base_i$ and $weight_i$ are the hierarchical base value and weight value, and the algorithm will eventually output a score. When the value is closer to the standard, the *score* is bigger.

$$score = base_i + weight_i(1 - abs\_value / internal) \tag{6}$$

Secondly, due to the multi-level and multi-quantity characteristics of the monitoring points, the weights of the measuring points for different scores should also be different: the input of the function is the set of multi-point scores obtained by formula (6); These values are divided into *low_score* and *high_score* according to a threshold. then *low_pro* and *high_pro* are calculated from normalized sets of the values as the proportion of low and high scores. The method of normalization is l2 normalization.

$$score = low\_score * low\_pro + high\_score * high\_pro \tag{7}$$

These two functions are very meaningful in actual production, because the temperature or humidity close to the standard value does not need to be processed, and when it exceeds a certain threshold may lead to safety accidents.

3.3 Genetic Algorithm For Optimal Solution

For genetic algorithms, the most important steps are selection, crossover and mutation. The step to produce the optimized offspring is the selection. According to individual fitness and certain rules, the selection algorithm selects some individuals with good traits from the nth generation population and inherits them into the next generation (n + 1) population. In this selection process, the greater the individual's fitness, the greater the chance of being selected to the next generation. The fitness ratio f_i of an individual *i* and NP of population size is the probability formula for *i* to be selected as follows:

$$P_i = \frac{f_i}{\sum_{i=1}^{NP} f_i} (i = 1,2,3....,NP) \tag{8}$$

It is obvious that using the fitness function optimized by us can greatly reduce the survival probability of operators with large differences in individual values, and allow all operators with small or moderate universal distances to survive.

In addition, in order to speed up the convergence speed and generate the operator with higher fitness, we adopt the idea of the adaptive genetic algorithm[9] to dynamically change the probability of mutation and crossover.

$$\hat{p}_c = p_c(1 - \frac{f' - f_{avg}}{f_{max} - f_{min}}) \tag{9}$$

$$\hat{p}_m = p_m(1 - \frac{f' - f_{avg}}{f_{max} - f_{min}}) \tag{10}$$

Where p_c and p_m are the standard probability of crossover and mutation, and f' is the fitness of the individual , f_{avg} , f_{max} , f_{min} are the average, maximum and minimum value of fitness of this batch of population. When f' is higher than f_{avg} , $\hat{p}_m$ and $\hat{p}_c$ will be lower to keep the excellent individual unchanged.

Finally, the whole process of optimization system is shown in Fig.3..

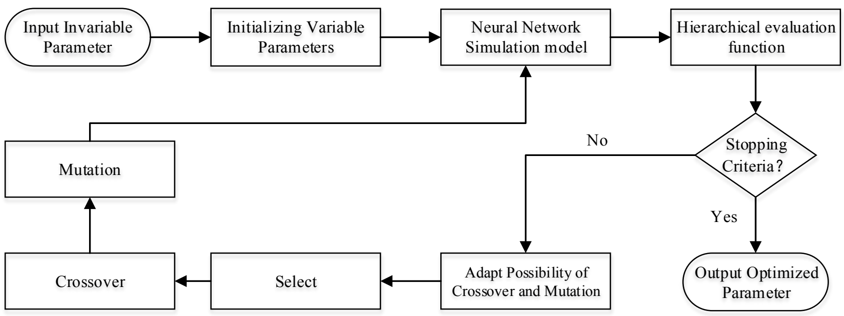

Fig. 3. Whole process of optimization system

4 Experimental Result

4.1 Neural Network Simulation Result

An accurate cooling system simulation is very useful for data center management. In order to ensure the safe operation of the equipment, the cooling parameters of the actual production environment are usually not directly adjusted. For example, after a system reboot, because of the great changes in the external environment, administrators can simulate according to the neural network to understand the general range of each parameter, and then adjust, it will not produce security risks, but also save a lot of manpower and material resources.

Another advantage of neural networks is that when the amount of data increases, the simulation results are more accurate than traditional machine learning methods such as decision trees. Because of this characteristic, we boldly select 100 parameters in many parameters, most of which are input from the network. At present, our experimental results are as follows:

The simulation results of a certain humidity monitoring point for a period of time.

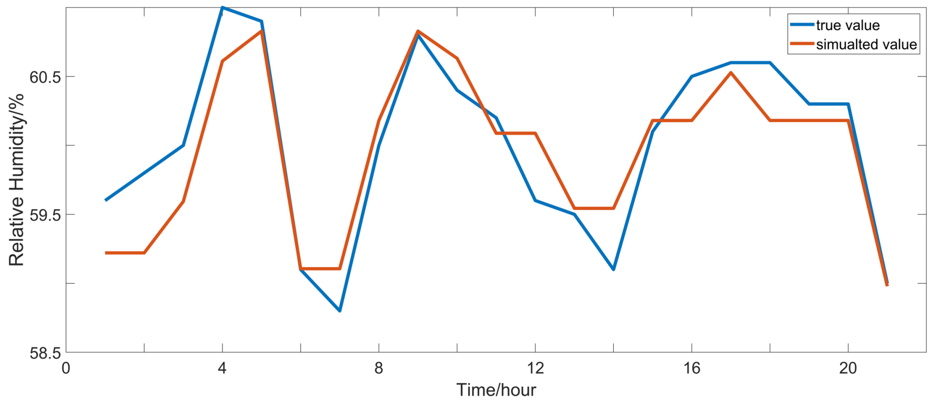

Fig. 4. The true value and simulated value of a humidity measuring point in a period of time.

Temperature of all measuring points at a certain point in time (some measuring points are not shown in the figure because the sensor is not activated or missing).

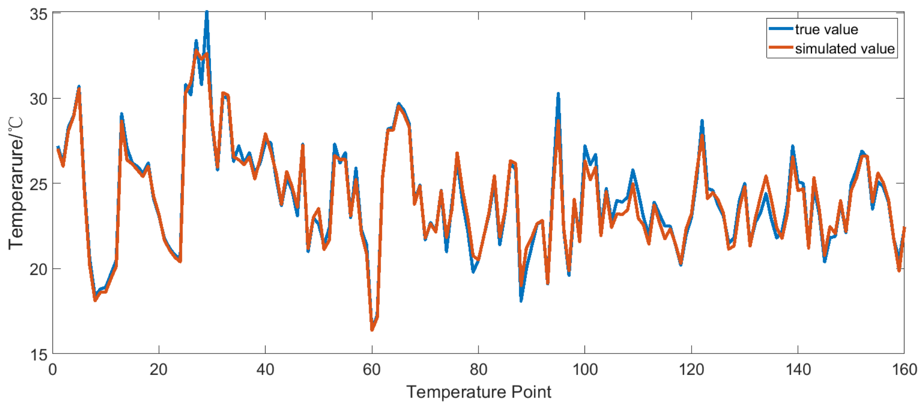

Fig. 5. Temperature of all measuring points at a certain point in time

Humidity of all measuring points at a certain point in time (some measuring points are not shown in the figure because the sensor is not activated or missing).

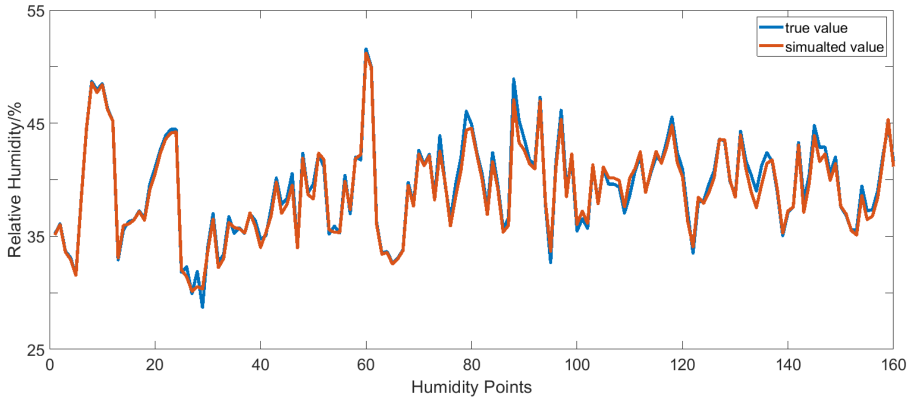

Fig. 6. Humidity of all measuring points at a certain point in time.

4.2 Hierarchical Evaluate Function

Using the hierarchical function does not reduce the overall error compared to ordinary MSE, but our goal is not to adjust the better points, but to pay more attention to the worse points. From Fig.7 (normalized error distribution) we can see that we did reduce the point farthest (normalized MSE>0.05)from the standard value. The remaining points that are not optimized are because the adjusted parameters are not sufficient for this purpose, such as the distance of the measurement points being too far from the topological distance of the air conditioner being adjusted.

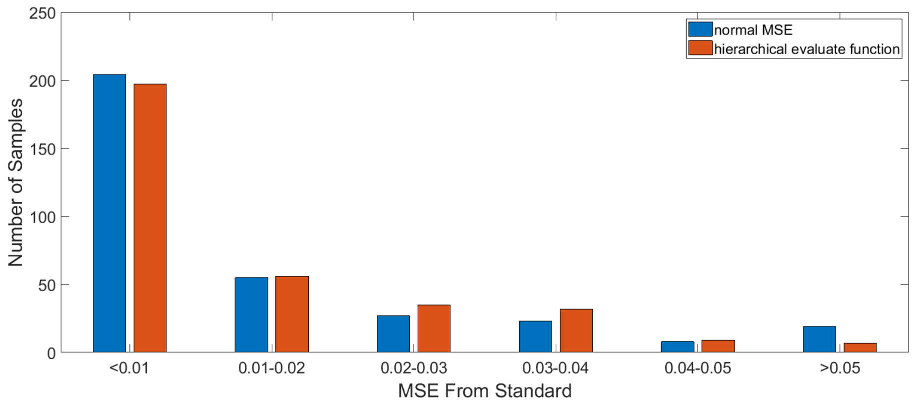

Fig. 7. MSE distribution caused by two different loss function.

4.3 Genetic Algorithm Optimization

As mentioned above, administrators can set the cooling parameters according to their own experience and the auxiliary simulation of the algorithm, but the disadvantages of doing so are obvious: time-consuming and laborious, and the accuracy is not high. This shows that a complete set of AI algorithms is necessary. Due to the limitation of time and the actual needs of the experiment, we only change 8 parameters, and fix the other 92 parameters for the experiment. These 8 parameters are the temperature and humidity control of 4 of the air conditioners. In this paper, we choose the artificial method and greedy algorithm to compare with our method. Setting parameters manually according to experience or based on physical models is a common practice in the industry, and Google's use of neural network sensitivity analysis to adjust the parameters of optimization parameters is basically greedy algorithm. In theory, the complete algorithm comparison should be in the simulation phase and the optimal solution phase using different algorithms, such as "neural Network + genetic algorithm" and "CFD + artificial." But because of the amount of calculation, we use the neural network in the simulation phase uniformly, but still can see that our method has a obvious effect. In addition, because only 8 parameters are changed, the difference between the algorithms is sometimes small, This because other parameters besides them need to be adjusted. And with the number of variable parameters increases, the greedy algorithm is easy to fall into the local optimal solution, while the effect of experience factors in the artificial method will be weakened as well, in the actual production environment, our method will be more excellent.

In the experimental details we refer to the method in [12], using the hill climbing algorithm as the optimized greedy algorithm. The step size is 0.01 and the number of iterations is 300. The basic crossover and mutation probabilities we used in the genetic algorithm were 0.5, 0.2, the number of initial population was 50, and the number of iterations was also 300.

In the experiment, our fitness function uses the hierarchical cost function, so the final result of MSE may not be the best, but in order to show the results more clearly, our final evaluation method is still MSE, we can see that the effect is still better than other methods which is shown in TABLE I (the Time is 5 points randomly selected).

To quantify this effect in the actual production situation, assuming that the temperature average error is 2 ℃when manually adjusting a certain set of parameters, and 10% error optimization is calculated according to the experimental result, then the temperature optimization for all devices is 2*10%*168= 67.2 (℃). Since we use a hierarchical evaluation function, most of this optimization come from poorly performing points which has significant benefits for safe operation of equipment. Moreover, this is calculated without taking into account the humidity error.

Table 1. MSE Between Optimized Results And Standard.

Time	Artificial Method	Greedy Algorithm	Our Approach
1	2.2894	2.1450	1.5504
2	0.6650	0.6293	0.5839

2	2.5906	2.4134	1.6903
4	2.6957	2.6593	1.4808
5	3.2031	3.0074	2.3037

5 Conclusion and Future Work

In this paper, we propose a combined algorithm based on machine learning to optimize the cooling system of data center. This approach can not only save human resources and time, but also make the control of cooling parameters more accurate. At the same time, it avoids the defect that the traditional greedy algorithm is easy to fall into the local optimal solution, and provides a better solution for multi-parameter simultaneous adjustment. However, due to the limitation of data set, we failed to further explore the relationship between setup and energy consumption. Besides, the lack of interpretability of neural networks is also a difficult problem. For this paper, the most important problem of the lack of interpretability is that it is impossible to explore the effect of adjusting cooling parameters on the measurement points at different topological locations. We have learned that in the work of [12], they used the decision tree to regularize the depth model, which help us move beyond sparsity toward models humans can easily simulate and thus trust. So our future work is to explore the of interpretability neural networks, making it possible for machine learning algorithms to replace CFD. At the same time, to research a more adaptable algorithms and improve computing efficiency are also the focus of work.

6 Acknowledgment

This work is supported by Science and Technology Program of State Grid Corporation of China under Grant No.5442DZ170019.

References

1. Li Li,Wenli Zheng,Xiaodong Wang,Xiaorui Wang. Data center power minimization with placement optimization of liquid-cooled servers and free air cooling[J]. Sustainable Computing: Informatics and Systems,2016,11.
2. Akhmad R.I. Mukaffi,Rizky S. Arief,Wisnu Hendradjit,Rahmat Romadhon. Optimization of Cooling System for Data Center Case Study: PAU ITB Data Center[J]. Procedia Engineering,2017,170.
3. Gao J, Jamidar R. Machine learning applications for data center optimization[J]. Google White Paper, 2014.
4. Qiu M, Ming Z, Li J, et al. Phase-change memory optimization for green cloud with genetic algorithm[J]. IEEE Transactions on Computers, 2015, 64(12): 3528-3540.
5. Abdallah F, Basurra S, Gaber M M. A hybrid agent-based and probabilistic model for fine-grained behavioural energy waste simulation[C]//Tools with Artificial Intelligence (ICTAI), 2017 IEEE 29th International Conference on. IEEE, 2017: 991-995.

6. Liu H, Song B. Stock Trends Forecasting by Multi-layer Stochastic ANN Bagging[C]//Tools with Artificial Intelligence (ICTAI), 2017 IEEE 29th International Conference on. IEEE, 2017: 322-329.
7. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. arXiv preprint arXiv:1502.03167, 2015.
8. Xu B, Wang N, Chen T, et al. Empirical evaluation of rectified activations in convolutional network[J]. arXiv preprint arXiv:1505.00853, 2015.
9. Chen R, Liang C Y, Hong W C, et al. Forecasting holiday daily tourist flow based on seasonal support vector regression with adaptive genetic algorithm[J]. Applied Soft Computing, 2015, 26: 435-443.
10. Peremezhney N, Hines E, Lapkin A, et al. Combining gaussian processes, mutual information and a genetic algorithm for multi-target optimization of expensive-to-evaluate functions[J]. Engineering Optimization, 2014, 46(11): 1593-1607.
11. Al-Betar M A, Awadallah M A, Bolaji A L, et al. β-hill climbing algorithm for Sudoku game[C]//Information and Communication Technology (PICICT), 2017 Palestinian International Conference on. IEEE, 2017: 84-88.
12. Wu M, Hughes M C, Parbhoo S, et al. Beyond sparsity: Tree regularization of deep models for interpretability[J]. arXiv preprint arXiv:1711.06178, 2017.
13. Mittal G S, Zhang J. Artificial neural network-based psychrometric predictor[J]. Biosystems Engineering, 2003, 85(3): 283-289.