Journal of Computers Vol. 35 No. 3, June 2024, pp. 13-26
doi: 10.53106/199115992024063503002

Load Forecasting Based on Optimized Random Forest Algorithm in Cloud Environment

Xin Sui[1], Hailong Zhao[2], Honghua Xu[2], Xiaolong Song[2], Dan Liu[2*]

[1] Jilin Provincial Institute of Education, Changchun 130022, Jilin, China
suixin_jledu@sina.com

[2] College of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, Jilin, China
{1019802235, 5739895, 18621820}@qq.com , ld@cust.edu.cn

Received 7 October 2022; Revised 25 October 2022; Accepted 26 October 2022

Abstract. To solve the problem of unbalanced resource load in cloud data center, a resource load forecasting method which is based on random forest model from the perspective of resource load forecasting is proposed in the paper. This method combines genetic algorithm with random forest algorithm to solve the problem that random forest algorithm can not determine the combination of parameter in order to obtain the optimum forecasting effect. The results of experiment show that compared with the super parametric method of random forest model, which is optimized by random search, the one optimized by genetic algorithm proposed in this paper has higher forecasting accuracy.

Keywords: random forest, load balancing, super parametric optimization

1 Introduction

At present, more and more enterprises deploy data information to the cloud data center, which provides users with great convenience. In recent years, the data center has achieved rapid growth. Promoting the green development of the data center is not only conducive to the sustainable development of the digital economy, but also the key to save energy and reduce consumption of new infrastructure. At the same time, the energy consumption of cloud data center is huge. Due to the massive data throughput and computing, the energy consumption face unprecedented challenges in the data center. How to accelerate the green development of cloud data center and reduce carbon emissions has become an important issue to be solved urgently.

According to statistics, the cost of energy consumption accounts for 50% of the total cost in the data center, including power, room occupation, bandwidth and other aspects. While the high energy consumption is generating, the data center also suffers from low energy efficiency, and the resource utilization of servers has been at a low level. Through virtualization and real-time migration technology, cloud computing technology can save energy, improve efficiency, and truly achieve energy conservation and emission reduction. From the perspective of energy conservation, by accurately predicting the resource load of the cloud data center, virtual machines that meet the energy-saving strategy are migrated to other servers, effectively integrate server resources, improve server resource utilization, and achieve the goal that reduces energy consumption of the cloud data center.

Therefore, in order to realize the effective integration of server resources, accurately predicting the server resource load is necessary. The prediction algorithm is mainly based on regression algorithm and neural network algorithm. Regression algorithms include linear regression, ridge regression, logistic regression, decision tree regression, etc. The advantage of regression algorithm lies in its simplicity, directness and fast training speed, while the main defect is that it requires strict assumptions and needs to deal with outliers. Neural network algorithms include BP neural network, Elman neural network, RBF neural network, GRNN neural network, wavelet neural network and various combined neural networks.

* Corresponding Author

2 Related Works

Calheiros et al proposed a cloud workload prediction module based on the autoregressive integrated moving average (ARIMA) model for SaaS providers, which used real traces of requests to Web servers evaluating its accuracy of future workload prediction [1]. Liu et al proposed an adaptive workload forecasting method, which classified workloads into different categories according to forecasting and then automatically assigned them to different forecasting models based on their characteristics [2]. Zharikov et al proposed an adaptive prediction model and developed a corresponding adaptive prediction method to predict the workload of the cloud data center, which guaranteed the quality of service and reduces power consumption to a large extent. Through experiments, the best combination of prediction methods and training set size could be determined to adapt to the current state of servers in the cloud data center. Therefore, the author considered six alternative prediction methods and 77 training data windows in each management step [3]. C. Lei et al used particle swarm optimization (PSO) to optimize random forests (RF) and support vector machines (SVM) in order to obtain optimal super parameters. To transform the original input data into irrelevant variable data set, they used Principal component analysis (PCA), which reduced the dimension of input variables. The results showed that no matter with PCA model or not, the RF model was the more robust of the two, and was less affected by its own parameters [4]. Motaki et al proposed an integrated learning strategy, which involved linear and non-parametric regression methods to predict the six key real-time migration indicators of each real-time migration algorithm. The model allowed to consider the best combination of algorithm metric pairs to migrate VM [5]. The way that a new combined model method of random forest based on pearson correlation coefficient on the basis of random forest was proposed by which included linear regression, ARIMA, and support vector regression for web applications. Further the model parameters were selected through a heuristic approach. The results of experiment described that not only the root-mean-squared error and mean absolute percentage error metrics were perceptibly improved but Yan et al, which was used to work out the problem that low prediction accuracy under a large number of characteristics and big data. Pearson correlation coefficient was used for correlation test to delete irrelevant features; The improved grid search method was used to optimize the parameters of the decision tree; The residual characteristics were modeled and regressed by using random forest, and the final conclusion was drawn [6]. An adaptive prediction model was proposed by Singh et al also the quality of service of web applications in a cloud environment was improved [7].

Kumar et al proposed a workload prediction model using long short term memory (LSTM) networks, which was tested on three benchmark datasets of web server logs. The results showed that the method reduced the mean squared error to $3.7*10^{-3}$, so its prediction accuracy of this method is high [8]. Bi et al designed a method to predict the time series, which integrated deep learning. For better results in forecasting workload and resource time series, the method combined a network model that includes bidirectional and grid long and short-term memory networks. Through experimental comparison, compared with several widely used methods in the Google cluster tracking datasets, the prediction accuracy of this method has better performance [9]. A cloud datacenter workload prediction model based on Evolutionary Quantum Neural-Network (EQNN) was presented by Singh et al, which is much novel. Harnessing the computational efficiency of quantum computing to actively estimate workload and resource requirements with greater accuracy by encoding workload information into qubits and propagating the information through the network, this method greatly improves the accuracy [10]. Wang et al proposed a mechanism, DBN with transfer learning (TL-GDBN), whose type is growth, and the size of structure could be automatically determined by itself, it could improve not only its learning rate but also model accuracy [11]. A state-of-the-art forecasting method was presented by Baldan et al, which analyzed the CWF problem from a time series perspective and was applied on real workload datasets in several different data centers. The results showed that no seasonal regularity was found in the analyzed series, and it is not linear in nature. In addition, the penalty cost can typically be reduced to 30% on average on the analyzed datasets [12]. In order that predict the workload which will be like at the next time in advance, an integrated prediction method was proposed by Bi et al, which combined the Savitzky-Golay filter and wavelet decomposition with stochastic configuration networks. This approach is divided into two steps, that is, used SG filter smoothing the task time series, and then decomposed the series into multiple components by wavelet decomposition. On this basis, a comprehensive model is established for the first time, which can well describe the statistical characteristics of trends and details [13]. Banerjee et al put forward a multi-step-ahead workload prediction method using machine learning technology, which could make more effective use of resources in the process of allocation, thus reducing the overall energy consumption of data center [14]. In order to predict the cloud workload for industry informatics, an efficient deep learning model using the canonical polyadic decomposition was proposed by Zhang et al. In this model, the parameters

needed to be compressed, which is accomplished by transforming the weight matrix into a standard multivariable format. The datasets from PlanetLab was used to test the model, and the prediction results showed that the model has better performance than other machine learning prediction methods [15]. Zhu et al put forward a method of encoder-decoder, which makes use of long-term and short-term memory (LSTM) with attention mechanism. The process was divided into three steps. First, the sequential and contextual features of the historical workload data were extracted, and the process needed to be completed by using an encoder network. Second, the batch workloads were predicted, and a decoder with integrated attention mechanism was needed in this process. Third, the method was applied to the workload traces datasets demonstrate of Alibaba and Dinda. The results of experiment showed that compared with the hybrid workload prediction of other algorithms in cloud computing environment, this method has better performance [16]. Karim et al put forward a mixed method including RNN and Bi LSTM to forecast the CPU load of virtual machines, which could improve the performance of using a single technology alone [17]. Sun et al proposed a Long-Short-Term-Memory (LSTM) based link quality confidence interval lower boundary prediction for the smart grid. In order to train two LSTM neural networks, it is necessary to decompose the SNR time series into two parts: deterministic and stochastic. This process needs to be completed by wavelet denoising algorithm. The results of experiment showed that the LQP method was more accurate and reliable than other link quality prediction methods [18].

To sum up, the regression-based method has proved to be effective in load forecasting. However, most of the methods are only applicable to the workload prediction of representation. For the complex prediction of server workload affected by multiple factors (CPU, memory, hard disk, network), the above regression algorithm is difficult to accurately predict; The methods based on machine learning and deep learning can neither make use of long-term memory dependence, nor solve the problem of gradient disappearance, nor accurately predict the complex and changeable workload in the cloud environment.

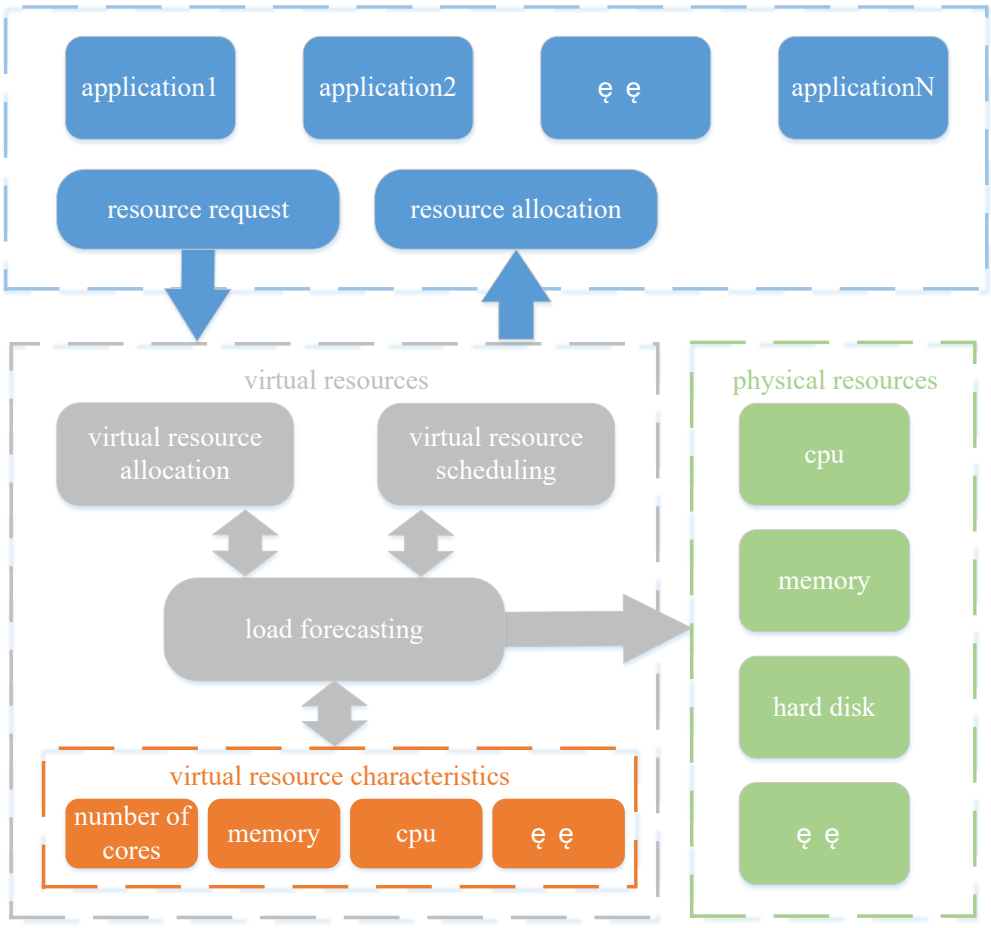


Fig. 1. Schematic diagram of load balancing based on virtual machine scheduling

In the cloud environment, the user task has the characteristics of strong complexity and high sparsity, while the random forest algorithm is insensitive to the sparsity of the data set, and has the advantages of strong generalization ability, fast training speed and stable classification results. Compared with other models, it is more suitable for parallel operation. At the same time, the random forest algorithm can predict according to multiple parameters and can consider the fluctuation of prediction results caused by multiple task characteristics. Therefore, compared with other prediction models, random forest algorithm is more suitable for task feature prediction in cloud environment. However, only distinguishing task characteristics can not deal with more complex situations. Task feature classification is discrete, and the factors that affect task features are diverse. For example, while receiving a large number of computing intensive tasks, you also need to perform IO intensive tasks. Discrete data is not conducive to accurate prediction. Therefore, we take the factors that determine the task characteristics as the characteristics of the resources required for prediction computing, which can effectively predict the changes in the demand for computing resources, realize the rational allocation of user tasks, and finally realize the load balancing of computing resources. Fig. 1 is a schematic diagram of load balancing based on virtual machine scheduling.

3 Super Parameters Optimization of Random Forest Model based on Genetic Algorithm

Random forest (RF) algorithm uses the idea of integrated learning, and it integrates multiple trees [19]. It takes the decision tree as the base learner, uses the bootstrap method to generate the training set, and generates the decision tree for each training set. When selecting feature attributes in the training process of traditional decision tree, the optimal attribute will be selected in the attribute set; Random forest is to randomly select an attribute subset from the attribute set, and then select an optimal attribute in the attribute subset. Random forest algorithm has the following characteristics: (1) accurate prediction; (2) ability to run large data sets; (3) capable of processing data with high-dimensional features; (4) be able to evaluate the importance that each feature for prediction results.

The parameters of random forest model have a great impact on the accuracy of model prediction, including the following parameters:

(1) n_estimators: number of decision trees, which is mean that the number of subsets generated by put back sampling of the original datasets. If n_estimators is too small and easy to under fitting; if it is too large, the improvement in the prediction accuracy of the model can not be significantly.

(2) max_features: the maximum number of features considered when building the optimal model of decision tree.

(3) min_samples_split: the minimum samples number that can be divided by a node. This value limits the conditions for continued division of the subtree, until the number of samples in the node is less than min_samples_split will stop selecting the optimal feature to devide.

(4) min_samples_leaf: the minimum samples number of leaf nodes. If the samples number of leaf nodes is less than min_ samples_ leaf, prune.

(5) max_leaf_nodes: maximum number of leaf nodes to prevent over-fitting. The default is none, meaning that the maximum number of leaf nodes is unlimited. If the constraints are added, the optimal decision tree will be built in the range of the maximum number of leaf nodes.

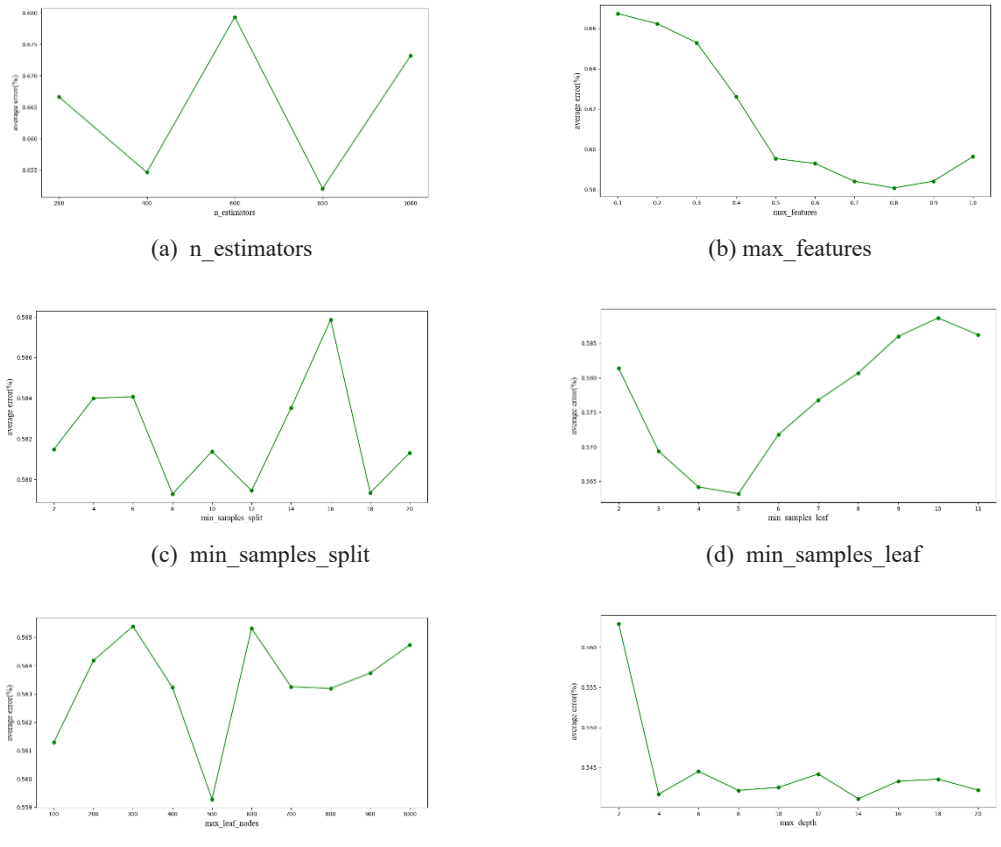
(6) max_depth: the maximum depth of the decision tree, if max_depth is equal to none, the depth of the subtree will not be limited when building the model; When the data sample size is large and there are many features, the maximum depth should be limited; When the situation is contrary to the above, the maximum depth shall not be limited.

This paper optimized the accuracy of the prediction error of the random forest model by adjusting the values of the above six parameters, and finally obtained the sensitivity of each parameter to the prediction error. In the process of testing the sensitivity of parameters to the prediction error of random forest model, the value of n_estimators, max_features, min_samples_split, min_samples_leaf, max_leaf_nodes and max_depth is shown in Table 1.

Table 1. Parameter sensitivity value of random forest model

Parameters	Values
n_estimators	200, 400, 600, 800, 1000
max_features	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0
min_samples_split	2, 4, 6, 8, 10, 12, 14, 16, 18, 20
min_samples_leaf	2, 3, 4, 5, 6, 7, 8, 9, 10, 11
max_leaf_nodes	100, 200, 300, 400, 500, 600, 700, 800, 900, 1000
max_depth	2, 4, 6, 8, 10, 12, 14, 16, 18, 20

In the process of parameter sensitivity testing of random forest model, when modifying the value of a parameter, ensured that the values of other parameters were unchanged. In this paper, the average value of 20 prediction errors was taken as the evaluation criterion of parameter sensitivity. The smaller the average error was, the more accurate the prediction was. Fig. 2 shows the sensitivity results of various parameters of the random forest algorithm, the abscissa represents the different values of parameters, and the ordinate represents the average prediction error. As can be seen from Fig. 2, average prediction error of n_estimators, min_samples_split, max_depth had multiple minimum values, which will affect the accuracy of the prediction error to a great extent, that is, these three parameters were sensitive to the prediction error of random forest algorithm; However, average prediction error of max_features, min_samples_leaf and max_leaf_nodes was only a minimum value, which had little impact on the prediction error. Therefore, this paper optimized the three parameters of the random forest model by genetic algorithm [20], including the number of random forest decision trees n_estimators, minimum number of samples required to split internal nodes min_samples_split and max_depth. The experimental results showed that, when max_features was 0.8, the prediction error was the lowest; when min_samples_leaf was 5, the prediction error was the lowest; when max_leaf_nodes was 500, the prediction error was the lowest. During the experiment, these three parameters were set to fixed values and were not used as parameter optimization options.



(a) n_estimators (b) max_features

(c) min_samples_split (d) min_samples_leaf

(e) max_leaf_nodes (f) max_depth

Fig. 2. Sensitivity testing results of parameters of random forest algorithm

In the process of load forecasting of resource nodes by random forest model, this paper proposed a load forecasting algorithm based on improved random forest, which was based on genetic algorithm n_estimators, max_depth and min_samples_split three parameters were optimized. Firstly, the three parameters of random forest model were coded to randomly generate population individuals; Then, the individual decoding of the population was used as the input parameter of the random forest model. The optimization of the three parameters of the random forest model was completed through model training, prediction and population iteration; Finally, the optimized n_estimators, max_depth and min_samples_split was used as the input parameter of the random forest model to predict and optimize the load of resource nodes in the cloud data center. The specific process is shown in Fig. 3.

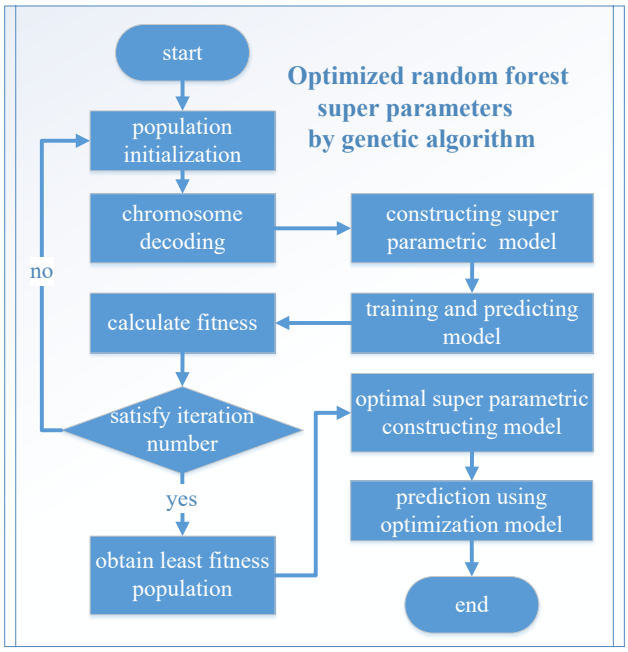


Fig. 3. Load forecasting algorithm based on improved random forest

In this paper, genetic algorithm was used to analyze n_estimators, max_depth and min_samples_split carried out chromosome coding with three parameters to generate 25 bit binary chromosome, the chromosome coding process is shown in Fig. 4. In the process of population initialization of genetic algorithm, the population size was assumed to be pop_size, chromosome length of each population was 25, the initial value of population was set to 0, and binary random assignment was made to each population chromosome, and finally the initial population was generated.

Fig. 4. Chromosome coding diagram

When the population initialization was completed, the individual population needed to be decoded before the three parameters of n_estimators, max_depth and min_samples_split was assignment. The chromosome decoding process was shown in formula (1):

$$\begin{aligned} &\text{n\_estimators} = \text{pop} \mid 0 \times 1FF8000 >> 15. \\ &\text{min\_samples\_split} = \text{pop} \mid 0 \times 7C00 >> 10. \\ &\text{max\_depth} = \text{pop} \mid 0 \times 3FF. \end{aligned} \tag{1}$$

After chromosome decoding, took the three parameters of n_estimators, max_depth and min_samples_split as the input parameters of the random forest model, trained and predicted the load data set of resource nodes, and calculated the prediction error of the random forest model according to the prediction results. The prediction error could be calculated by formula (2-4), as shown below:

$$\text{rmse} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(cpu_i^{pre} - cpu_i^{act})^2}. \tag{2}$$

$$\text{mae} = \frac{1}{N}\sum_{i=1}^{N}\left|cpu_i^{pre} - cpu_i^{act}\right|. \tag{3}$$

$$\text{errorRate} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|cpu_i^{pre} - cpu_i^{act}\right|}{cpu_i^{act}} * 100\%. \tag{4}$$

In formula (2-4), cpu_i^{pre} represented the predicted CPU utilization of the ith virtual machine, cpu_i^{act} represented the actual CPU utilization of the ith virtual machine, and N represented the number of virtual machines. Rmse represented the mean square error of prediction, mae represented the average absolute error of prediction, and errorRate represented the prediction error rate.

The algorithm in this paper was to realize the prediction and optimization of CPU utilization of resource nodes in cloud data center. The more obvious the error between the predicted value and the actual value, the more accurate the prediction was. Therefore, the fitness function of genetic algorithm used in this paper was shown in formula (5):

$$\text{fitness} = a * \text{rmse} + b * \text{mae} + c * \text{errorRate}. \tag{5}$$

In formula (5), rmse was calculated by formula (2), mae was calculated by formula (3), and errorRate was calculated by formula (4); a. b and c represented the proportion of each prediction error, and fitness represented the fitness of individual population.

The process of load forecasting algorithm based on improved random forest model proposed in this paper is as follows. Firstly, the three parameters of the random forest model were coded to generate population; Secondly, in the calculation of population fitness of each generation, decoded the population to generate three parameter lists of n_estimators, min_samples_split and max_depth, trained and predicted the random forest model; Calculate the prediction error according to formula (2-4), calculate the population fitness according to formula (5), and saved the population individual with the lowest fitness to the list of the best species; Then, crossover, mutation, selection and other operations were adopted for the population to save the newly generated population individuals to the next generation of population; When the maximum number of iterations was reached, the population with the least fitness was decoded to obtain three optimization parameters of the random forest model; Finally, the random forest model was trained based on the optimized parameters, and the load of resource nodes in the next time period was predicted.

```
Program LoadForecastingAlgorithm(population size S, iterations number N,
  crossover probability Pc, mutation probability Pm, max features Mf,
  min samples leaf Ms, max leaf nodes Ml)

Population Initialization:
for i in S:
  Randomly generate a 25 bit binary number and add it to the population list Pl;
for i in N:
  for j in Pl:
        Fitness Calculation Stage:
```

```
        Decode the ith chromosome of Pl and assign values to n_estimators Eo, min_
        samples_split So, max_depth Do;
        According to Eo,So,Do and Mf, Ms, Ml, create a random forest model and train;
        Calculation prediction error of random forest model based on formula (2-4);
        Calculate the individual fitness based on formula (5) and add it to the Fpl;
  Sort the Fpl in the order of fitness from small to large;
  Save Fpl[0] to the Bpl;

  Population Crossover, Mutation and Selection Stage:
  for i in Pl:
      Assign Pl [i] to the individual pop of the father population individual P^f;
      if random number is less than Pc:
         A population individual is randomly generated from the Pl and assigned to
         the mother population individual P^m;
         Randomly generate an integer int_random1 of [0,24] and a bool type number
         bool_number;
         if bool_number is true:
            P^f and P^m from 0 to int_random1 were crossed;
         else if bool_number is false:
            P^f and P^m from int_random1 to 24 were crossed;
         Assign a new population individual generated by crosso ver to P^c;
      if random number is less than Pm:
         Randomly generated integer int_random2 of [0,24],Pc's chromosome of int_
         random2 is reversed to generate a new population individual pop_new and
         added to the next generation population list;
      Add the population individuals in the top 80% of Fpl to the next generation
      population list;
      Randomly generated S * 20% population individuals are added to the next gener-
      ation population list;
  Sort the Bpl according to the order of fitness from small to large;
  Decode Bpl[0] and assign values to n_estimators_best Eb ,
  min_samples_split_best Sb, max_depth_best Db;

Random Forest Model Training and Predicting Stage:
According to Eb, Sb, Db and Mf, Ms, Ml, create a random forest model and train the
load data set of resource nodes;
The random forest model predicts the load in the next period of time;
End
```

4 Acknowledgement

4.1 Experimental Data

This experiment uses the datasets which is the tracking data version disclosed by Microsoft azure. It uses the virtual machine workload data from azure in 2017, which contains the relevant information about virtual machine and CPU utilization. This dataset specifically includes ID of virtual machine, ID of virtual machine description, ID of deployment, creation time of virtual machine, deletion time of virtual machine, category of virtual machine, number of virtual machine cores, memory of virtual machine, timestamp, minimum CPU value, maximum CPU value, average CPU value, etc.

4.2 Experimental Data Preprocessing

On the public datasets provided by azure (https://github.com/Azure/AzurePublic-Dataset), the virtual machine ID, virtual machine description ID and deployment ID are strings, which are independent of the requirements of computing resources. In the process of model training, the training data included: virtual machine creation time, virtual machine deletion time, virtual machine category, number of virtual machine cores, virtual machine memory, timestamp, CPU minimum value, CPU maximum value, and prediction data: average value of CPU.

4.3 Prediction Comparison Experiment

Fig. 5 to Fig. 7 shows the comparison of linear regression model [21], k-nearest neighbor regression model [22], support vector regression model [23] and random forest regression model in the prediction accuracy of average CPU utilization.

A comparison between the predicted average CPU utilization value of the four models and the actual average CPU utilization data is shown from Fig. 5. It highlights that the predicted values of k-nearest neighbor regression model and support vector regression model deviate greatly from the actual average CPU utilization. The predicted values of linear regression model and random forest model are not significantly different from the actual average CPU utilization rate.

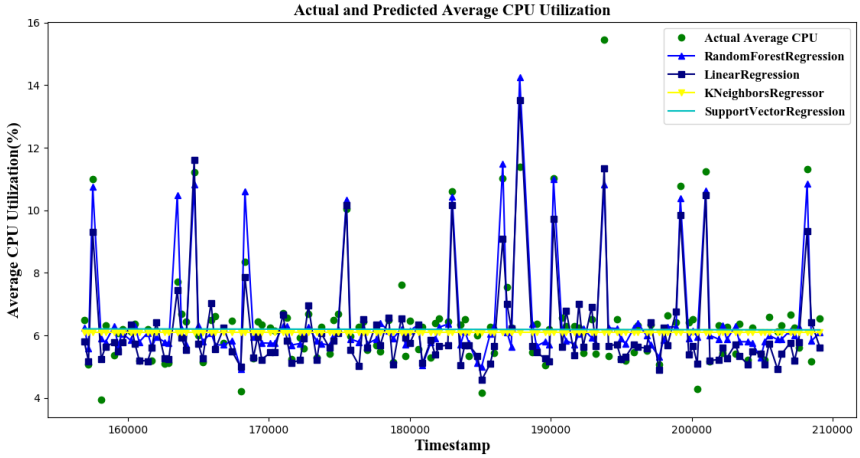


Fig. 5. Comparison of average CPU utilization prediction

Fig. 6 shows the error comparison between the predicted value and the actual value of the average CPU utilization. Fig. 7 is a box comparison diagram of the absolute error between the predicted value and the actual value. The experimental results showed that the average absolute error of k-nearest neighbor regression model is 0.98, the average absolute error of support vector regression model is 0.97, the average absolute error of linear regression model is 0.63, and the average absolute error of random forest model is 0.57. Among the four models, the random forest algorithm had the smallest prediction error for the average CPU utilization.

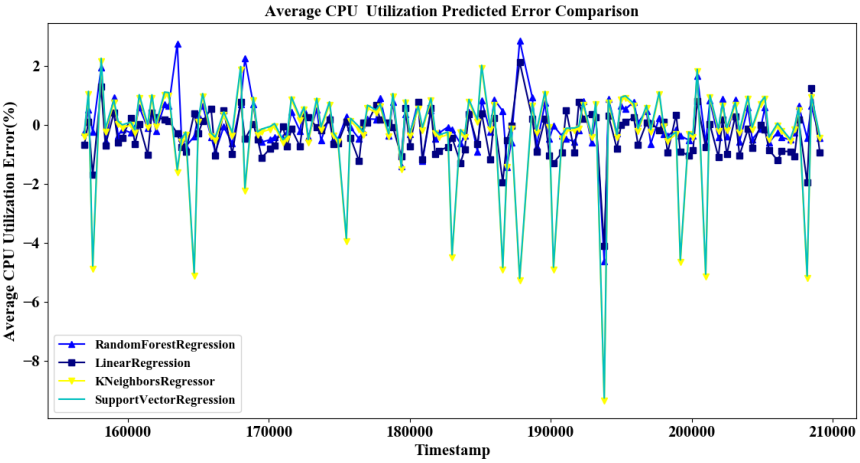


Fig. 6. Comparison of average CPU utilization prediction error

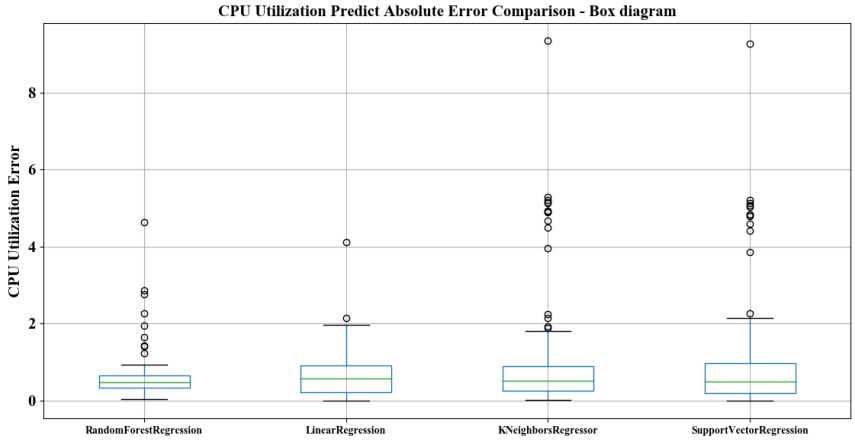


Fig. 7. Comparison of prediction error box diagram

4.4 Importance of Data Characteristics

Fig. 8 is a proportion diagram of data feature importance of random forest regression model. From the figure, what could be seen is the factors affecting the prediction accuracy of average CPU utilization mainly included: maximum CPU (max_cpu), minimum CPU (min_cpu), virtual machine category (vm_category) and time stamp (timestamp).

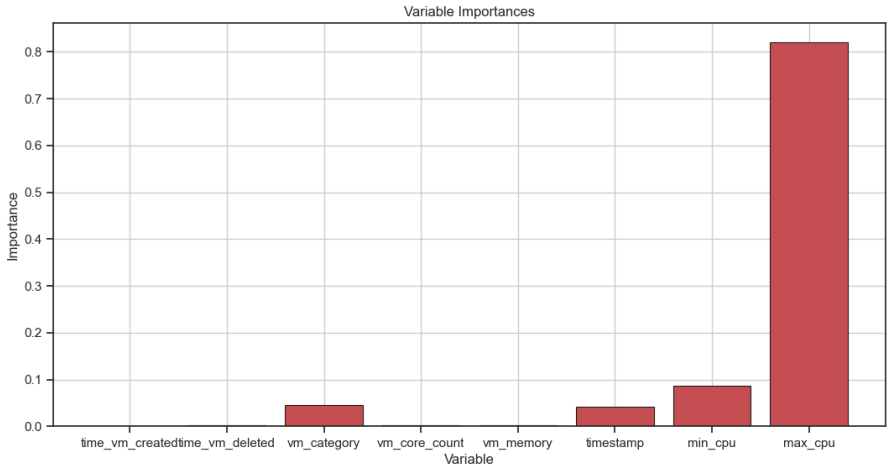


Fig. 8. Comparison of data feature importance based on random forest model

Fig. 9 is a diagram of the proportion accumulation of the importance of data characteristics of the random forest regression model. It could be concluded from the experimental results that the proportions of the data importance were maximum CPU (0.82), minimum CPU (0.09), virtual machine category (0.04), timestamp (0.04), virtual machine creation time (0.00), virtual machine deletion time (0.00), number of virtual machine cores (0.00), and virtual machine memory (0.00).

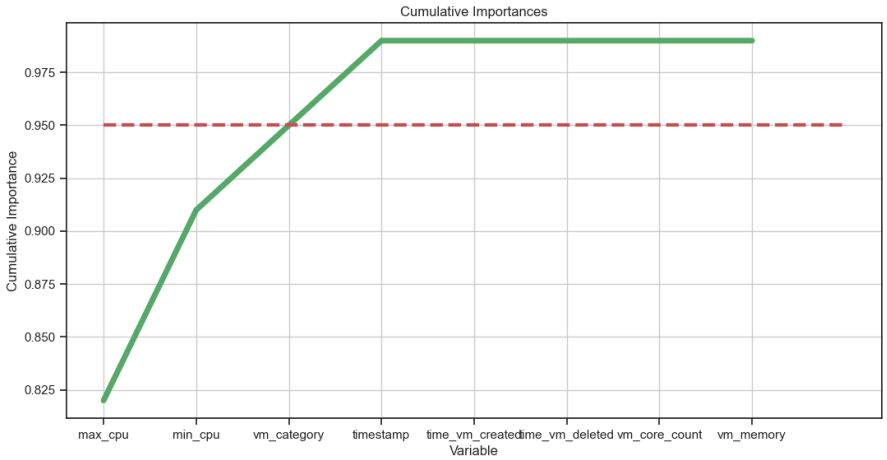


Fig. 9. Cumulative importance of data features based on random forest model

From the above two experiments, it could be seen that the factors affecting the prediction accuracy of average CPU mainly included: maximum CPU, minimum CPU, virtual machine category and timestamp. Therefore, this paper set the category of training data as the four factors that had the greatest impact on the prediction accuracy of average CPU, and reduced the training of irrelevant data features, which could effectively accelerate the convergence speed of the model. The result of the experiment shows that training and testing time of the model based on the above main data features is 5.79% shorter than that of all data features (eight factors), and the prediction accuracy based on the main data features is 0.66% lower than that of all data features.

4.5 Comparison Experiment of Super Parametric Prediction

Fig. 10 to Fig. 12 shows the comparative experiment on the prediction accuracy of average CPU utilization by optimizing the super parameter based on the random forest model. The super parameter optimization algorithm adopted is random search (RandomizedSearchCV) and genetic algorithm (GeneticAlgorithm_Optimization). A comparison between the predicted values and the actual average CPU utilization of the two optimized super parametric algorithms is shown in Fig. 10.

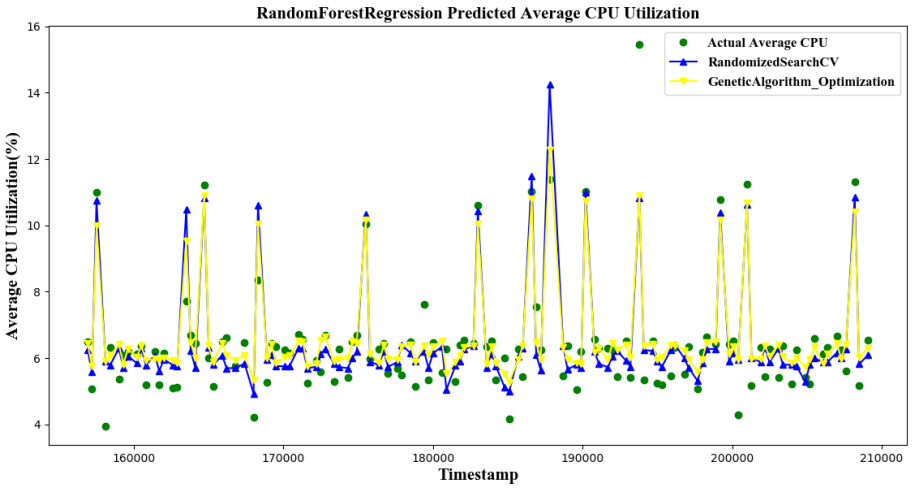


Fig. 10. Comparison of super parametric optimization predicted based on random forest model

Fig. 11 shows the comparison of average CPU utilization prediction errors of two optimized super parametric algorithms. As could be seen from the figure that the prediction error of genetic algorithm optimization super parameter proposed in this paper was less than that of random search optimization super parameter. The experimental results showed that the average absolute error (MAE) of super parameters optimized by genetic algorithm proposed in this paper is only 0.533, while the average absolute error (MAE) of random search super parameters is 0.6.

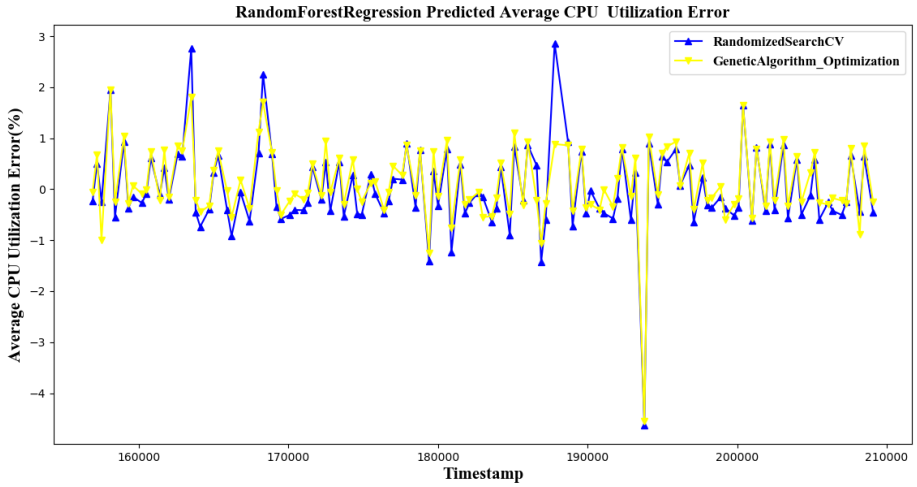


Fig. 11. Comparison of super parametric optimization prediction errors based on random forest model

Fig. 12 shows the box-plot comparison of average CPU utilization prediction errors of two optimized super parametric algorithms. It could be seen from the figure that there are four outliers in the prediction error of random search super parameters, while in the paper, using genetic algorithm optimizing super parameters reduce outlier to only one. The results of experimental showed that the mean square error (MSE) of super parameters optimized by genetic algorithm was 0.569 and the prediction accuracy was 91.26%, while the mean square error (MSE) of random search super parameters was 0.694 and the prediction accuracy was 90.49%.

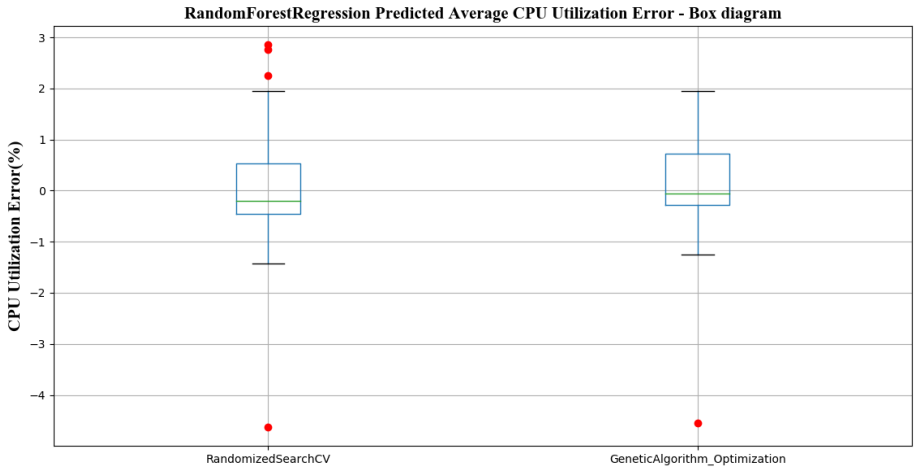


Fig. 12. Comparison of super parametric optimization errors based on random forest model-box diagram

5 Conclusion

From the perspective of resource load forecasting, this paper focused on the resource load balancing of cloud data center, proposed that using genetic algorithm optimizing the super parameters which is appeared in random forest model, and realized the accurate prediction of resource load. Firstly, this paper compared the random forest model with support vector regression model, linear regression model and k-nearest neighbor model, and proved the advantages of random forest model in resource load forecasting. Then, the importance of random forest model in data characteristics was compared to find out the main factors affecting resource load forecasting. Finally, the super parameters of genetic algorithm optimized random forest model and random search optimized random forest model proposed in this paper were analyzed separately. The data of experimental showed that the resource load forecasting of the model proposed in this paper has been improved in mean square error and prediction accuracy, and it is proved that the model is effective.

6 Acknowledgement

This topic has been partially supported by the “14th Five-Year” science and technology program of Jilin Provincial Department of Education (JJKH20220922KJ, JJKH20230842KJ); the reform of vocational and adult education of the Education Department of Jilin Province (2020ZCY348); and research topic of higher education teaching reform in Jilin Province (20213F28H04001O).

References

[1] R.N. Calheiros, E. Masoumi, R. Ranjan, R. Buyya, Workload prediction using arima model and its impact on cloud applications’ qos, IEEE Transactions on Cloud Computing 3(4)(2015) 449-458.
[2] C.H. Liu, C.C. Liu, Y.L. Shang, S.P. Chen, B. Cheng, J.L. Chen, An adaptive prediction approach based on workload pattern discrimination in the cloud, Journal of Network and Computer Applications 80(2017) 35-44.
[3] E. Zharikov, S. Telenyk, P. Bidyuk, Adaptive Workload Forecasting in Cloud Data Centers, Journal of Grid Computing 18(1)(2020) 149-168.
[4] C. Lei, J. Deng, K. Cao, Y. Xiao, L. Ma, W. Wang, T. Ma, C. Shu, A comparison of random forest and support vector machine approaches to predict coal spontaneous combustion in gob, Fuel 239(2019) 297-311.
[5] S.E. Motaki, A. Yahyaouy, H. Gualous, A prediction-based model for virtual machine live migration monitoring in a cloud datacenter, Computing 103(11)(2021) 2711-2735.
[6] Z.X. Yan, C. Qin, G. Song, Random Forest Model Stock Price Prediction Based on Pearson Feature Selection, Computer engineering and applications 57(15)(2021) 286-296.
[7] P. Singh, P. Gupta, K. Jyoti, Tasm: Technocrat arima and svr model for workload prediction of web applications in cloud, Cluster Computing 22(2)(2019) 619-633.
[8] J. Kumar, R. Goomer, A.K. Singh, Long short term memory recurrent neural network (lstm-rnn) based workload forecasting model for cloud datacenters, Procedia Computer Science 125(2018) 676-682.
[9] J. Bi, S. Li, H.T. Yuan, M.C. Zhou, Integrated deep learning method for workload and resource prediction in cloud systems, Neurocomputing 424(2021) 35-48.
[10] A.K. Singh, D. Saxena, J. Kumar, V. Gupta, A quantum approach towards the adaptive prediction of cloud workloads, IEEE Transactions on Parallel and Distributed Systems 32(12)(2021) 2893-2905.
[11] G.M. Wang, J.F. Qiao, J. Bi, W.J. Li, M.C. Zhou, TL-GDBN: growing deep belief network with transfer learning, IEEE Transactions on Automation Science and Engineering 16(2)(2019) 874-885.
[12] F.J. Baldan, S. Ramirez-Gallego, C. Bergmeir, F. Herrera, J.M. Benitez, A forecasting methodology for workload forecasting in cloud systems, IEEE Transactions on Cloud Computing 6(4)(2018) 929-941.
[13] J. Bi, H. Yuan, M. Zhou, Temporal prediction of multiapplication consolidated workloads in distributed clouds, IEEE Transactions on Automation Science and Engineering 16(4)(2019) 1763-1773.
[14] S. Banerjee, S. Roy, S. Khatua, Efficient resource utilization using multi-step-ahead workload prediction technique in cloud, The Journal of Supercomputing 77(9)(2021) 10636-10663.
[15] Q. Zhang, L.T. Yang, Z. Yan, Z. Chen, P. Li, An efficient deep learning model to predict cloud workload for industry informatics, IEEE Transactions on Industrial Informatics 14(7)(2018) 3170-3178.
[16] Y.H. Zhu, W.L. Zhang, Y.H. Chen, H.H. Gao, A novel approach to workload prediction using attention-based lstm encoder-decoder network in cloud environment, EURASIP Journal on Wireless Communications and Networking 2019(1)(2019) 274.

[17] M.E. Karim, M.M.S. Maswood, S. Das, A.G. Alharbi, Bhyprec: A novel bi-lstm based hybrid recurrent neural network model to predict the cpu workload of cloud virtual machine, IEEE Access 9(2021) 131476-131495.
[18] W. Sun, P.Y. Li, Z. Liu, X. Xue, Q.Y. Li, H.Y. Zhang, J.B. Wang, Lstm based link quality confidence interval boundary prediction for wireless communication in smart grid, Computing 103(2)(2021) 251-269.
[19] L. Breiman, Random Forests, Machine Learning 45(1)(2001) 5-32.
[20] C.F. Wang, K. Liu, P.P. Shen, A Novel Genetic Algorithm for Global Optimization, Acta Mathematicae Applicatae Sinica 36(2)(2020) 482-491.
[21] N. Pandis, Linear regression, American Journal of Orthodontics and Dentofacial Orthopedics 149(3)(2016) 431-434.
[22] S.C. Zhang, X.L. Li, M. Zong, X.F. Zhu, R.L. Wang, Efficient kNN Classification with Different Numbers of Nearest Neighbors, IEEE transactions on neural networks and learning systems 29(5)(2018) 1774-1785.
[23] A. Kavousi-Fard, H. Samet, F. Marzbani, A new hybrid Modified Firefly Algorithm and Support Vector Regression model for accurate Short Term Load Forecasting, Expert Systems with Applications 41(13)(2014) 6047-6056.