ISSN: 2723-9535

Available online at www.HighTechJournal.org

HighTech and Innovation Journal

Vol. 6, No. 2, June, 2025

PRO-BiGRU: Performance Evaluation Index System for Hardware and Software Resource Sharing Based on Cloud Computing

Nan Wang [1*]

[1] *Youth Political College of Inner Mongolia Normal University, Hohhot 010050, Inner Mongolia Autonomous Region, China.*

Received 29 March 2025; Revised 17 May 2025; Accepted 23 May 2025; Published 01 June 2025

Abstract

This study aims to address the performance evaluation challenges of computer hardware and software resource sharing in cloud computing environments. To achieve this, we propose an enhanced performance evaluation method by integrating the Poor-Rich Optimization (PRO) algorithm with the Bidirectional Gated Recurrent Unit (BiGRU) network. We first construct a comprehensive multi-dimensional performance evaluation index system that encompasses resource utilization, response time, throughput, and scalability. Subsequently, the PRO algorithm is employed to optimize the hyper-parameter design of the BiGRU network, thereby enhancing the model's learning ability and evaluation accuracy. Performance data is collected using system monitoring tools, and experiments are conducted to validate the model's effectiveness. The results demonstrate that the PRO-BiGRU model achieves an average evaluation accuracy of over 97% across four independent experiments, significantly outperforming traditional algorithms such as CNN, RNN, LSTM, and GRU. The proposed model not only improves the accuracy of performance evaluation but also provides a reliable basis for resource optimization and decision-making in cloud service platforms. The novelty of this research lies in the combination of the PRO algorithm with the BiGRU network, which effectively captures complex data features and enhances the model's reliability and robustness in performance assessment tasks.

Keywords: Cloud Computing; Computer Hardware and Software Resource Sharing; Performance Evaluation Index System Construction; Bi-Directional Gated Recurrent Unit Network.

1. Introduction

With the rapid development of cloud computing technology, the sharing of computer hardware and software resources has become increasingly common [1]. The widespread use of cloud computing platforms has transformed the way traditional IT architectures are used, providing organizations with flexible, scalable, and cost-effective solutions [2]. However, to take full advantage of cloud computing, it is crucial to ensure its stability and high performance [3]. Performance testing and tuning of cloud computing platforms are important for all types of organizations, including businesses, government agencies, and academic institutions [4]. Performance testing can help identify resource wastage and performance bottlenecks on cloud computing platforms, optimize resource usage, reduce costs, and improve resource utilization. A high-performance cloud computing platform can provide faster response time and higher availability, thus improving user satisfaction. Therefore, how to evaluate the performance of these resource shares to ensure that users receive high-quality services has become an important research topic.

* Corresponding author: wyr390129@whku-edu.cn

http://dx.doi.org/10.28991/HIJ-2025-06-02-014

To address the performance evaluation challenges in cloud computing environments, prior studies have explored various methodologies. For instance, Elnagar et al. [5] investigated the performance of load balancing algorithms, while Wang et al. [6] focused on the trustworthiness of cloud AI services. These studies highlight the importance of performance assessment in optimizing resource utilization and user satisfaction. However, existing research often focuses on single performance metrics, such as response time or throughput, and lacks a comprehensive evaluation framework that integrates multiple dimensions of performance. Additionally, traditional algorithms like CNN and RNN struggle with the complexity and dynamic nature of cloud environments, resulting in limited accuracy and adaptability [7]. This study fills these gaps by proposing a novel performance evaluation method based on the Poor-Rich Optimization (PRO) algorithm and Bidirectional Gated Recurrent Unit (BiGRU) network. We construct a multi-dimensional performance evaluation index system and optimize the BiGRU model using the PRO algorithm, significantly enhancing evaluation accuracy and reliability. Our approach not only addresses the limitations of existing methods but also provides a robust foundation for resource optimization and decision-making in cloud computing platforms.

The current research on performance assessment of computer software and hardware resource sharing under cloud computing technology is mainly divided into research on the application of cloud computing services in computer software and hardware resource sharing, research on the performance assessment index system of computer software and hardware resource sharing under cloud computing, research on the construction of the performance assessment model of computer software and hardware resource sharing under cloud computing, and related validation and analysis research. The main idea of this paper is to solve the performance assessment problem of computer software and hardware resource sharing under cloud computing technology, so it mainly carries out research from the establishment of the system and the construction of the performance model. To solve the performance evaluation problem of computer hardware and software resource sharing under cloud computing technology, researchers need to construct a complete set of performance evaluation index systems to ensure the rationality, security, and effectiveness of resource sharing [6]. Baresi et al. [8] designed a business resource-sharing method and business resource-sharing system based on cloud computing, which has multifaceted advantages by acquiring multifaceted information and carrying out targeted processing; Aqasizade et al. [9] focused its research on the design and implementation of a resource-sharing platform in a cloud computing environment and evaluates the platform's actual effect and performance through experimental validation and performance analysis, providing a basis for subsequent optimization and improvement. In general, in the current research on the construction method of the performance evaluation index system for computer hardware and software resource sharing under cloud computing technology, researchers are exploring how to construct a scientific, reasonable, and safe performance evaluation index system by comprehensively considering multiple aspects [10], such as resource demand, privilege management, and service quality assurance [11]. It is also investigating how to evaluate and optimize these indicator systems through experimental validation and performance analysis to ensure the efficiency and security of resource sharing [12].

Although there have been some studies focusing on performance evaluation in cloud computing environments, most of them concentrate on a single performance index, such as response time, throughput, etc., and lack a comprehensive performance evaluation index system [13]. Therefore, this paper will comprehensively consider multiple performance indicators to construct a complete performance evaluation index system for evaluating the performance of computer hardware and software resources shared under cloud computing technology. In this paper, by analyzing the performance evaluation problem of software and hardware resource sharing under cloud computing conditions, introduce the relevant index system construction scheme, use the poor-rich optimization algorithm [14] and bidirectional gated recurrent unit network [15], propose a performance evaluation method for software and hardware resource sharing under cloud computing conditions based on the PRO-BiGRU network, and adopt a system monitoring tool to collect performance data to validate the performance of the RO-BiGRU network model by comparative analysis. A network model for comparative analysis and validation.

This paper focuses on the performance evaluation of computer hardware and software resource sharing under cloud computing technology, and the overall framework is divided into four parts. Firstly, the characteristics and performance evaluation indexes of resource sharing under cloud computing environment are analyzed, and a comprehensive evaluation index system is constructed; secondly, a performance evaluation model combining the poor-rich optimization algorithm (PRO) and the bi-directional gated recurrent unit network (BiGRU) is designed; then, the evaluation accuracy and efficiency of the model are verified through experiments, and the performances of different algorithms are compared; lastly, the research results are summarized, the deficiencies are analyzed, and the future improvement directions are proposed to further enhance the application value and applicability of the model. and propose future improvement directions to further enhance the application value and applicability of the model.

This paper is organized as follows. Part 2 discusses the performance evaluation index system for computer hardware and software resource sharing under cloud computing. Part 3 presents the PRO-BiGRU model and its application in performance evaluation. Part 4 provides the experimental results and their analysis. Finally, Part 5 concludes the paper and proposes future research directions.

2. Performance Evaluation Index System for Computer Hardware and Software Resource Sharing

2.1. Cloud Computing

The development and application of cloud computing technology have penetrated various fields and have had a huge impact on traditional IT architectures. The concept of cloud computing can be traced back as far as the 1960s, when the idea of resource sharing was proposed. Subsequently, with the popularity of the Internet and the growth of the amount of information, cloud computing has been further developed [16]. The development of cloud computing can be divided into the germination, development, growth, and maturity periods, as shown in Figure 1.

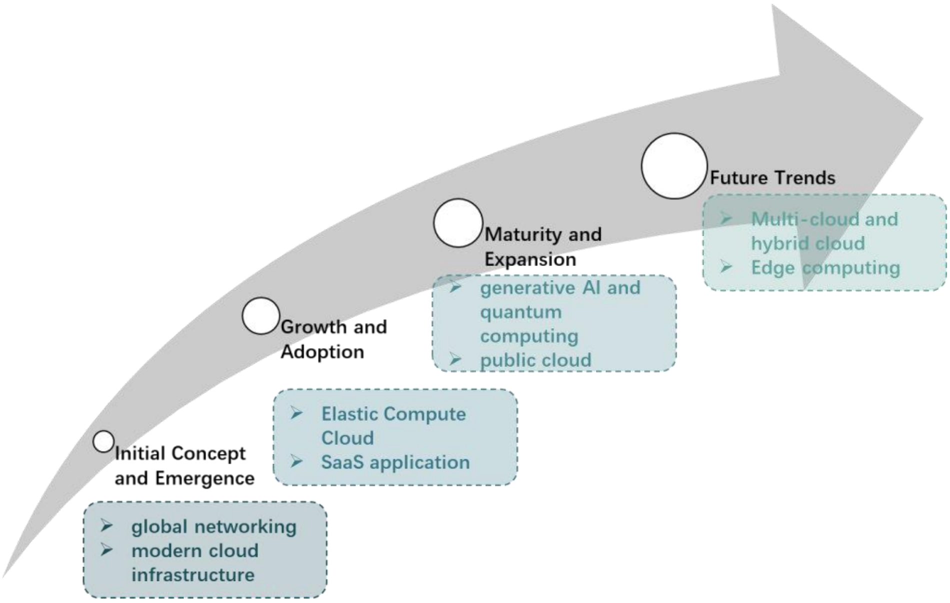


Figure 1. Stages of cloud computing development

The key technologies of cloud computing include virtualization, distributed computing, storage-as-a-service, and elastic scaling [17]. These technologies improve resource utilization and achieve high availability and scalability of data. The infrastructure includes physical hardware, network facilities, operating systems, and middleware, which constitute the infrastructure of cloud computing [18], as shown in Figure 2.

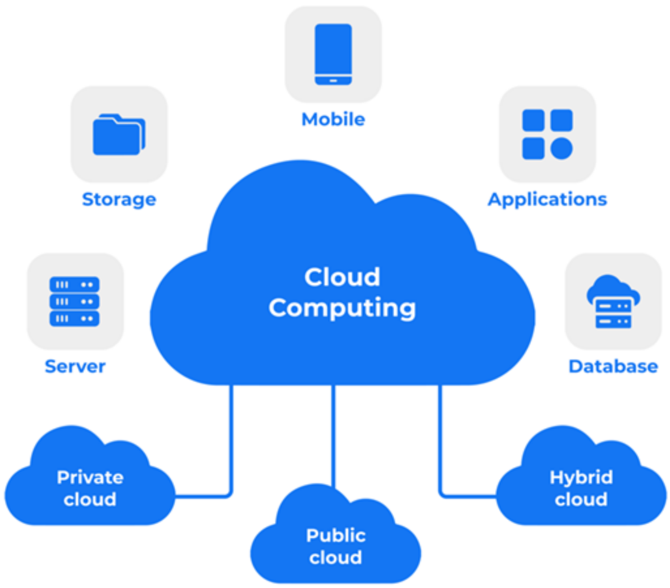


Figure 2. Cloud Computing Key Technologies and Architecture

The application areas of cloud computing include financial cloud, manufacturing cloud, education cloud, medical cloud, cloud gaming, cloud conferencing, cloud social networking, cloud storage, cloud security, and cloud transportation [19], as shown in Figure 3.

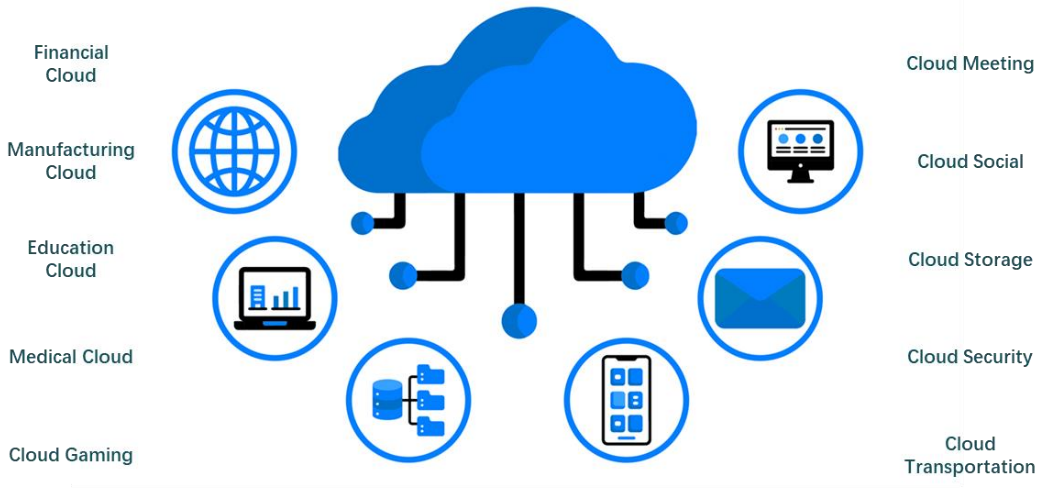


Figure 3. Cloud Computing Application Areas

2.2. Analysis of Performance Assessment Indicators

Performance evaluation of cloud computing platforms involves response time, throughput, scalability, and resource utilization. These metrics are crucial for enterprises and individual users to choose the right cloud service provider [20]. The performance assessment indicators of computer hardware and software resource sharing under cloud computing technology include elements such as resource utilization, response time, throughput, scalability, reliability, security, cost-effectiveness, and energy efficiency (Figure 4), which are analyzed below:

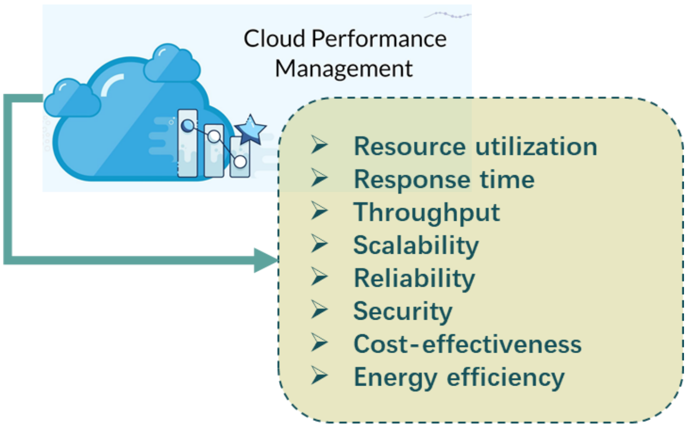


Figure 4. Classification of cloud computing assessment elements

- Resource utilization specifically includes indicators such as computing resource utilization (efficiency of using hardware resources such as CPU and memory), storage resource utilization (efficiency of using disks and storage devices), and network resource utilization (efficiency of using network bandwidth) [21].
- Response time specifically includes indicators such as the time interval between task submission and execution and the time interval between user request and response.
- Throughput Throughput includes metrics such as the number of tasks completed per unit of time and the amount of data transferred per unit of time.
- Scalability Scalability metrics include 1) the ability of the system to increase performance when resources are added and 2) the extent to which the system's performance degrades when resources are reduced.
- Reliability Reliability indicators include 1) the ability of the system to recover in the event of a failure and 2) data integrity and availability.
- Security Security indicators include: 1) Confidentiality, integrity, and availability of data; and 2) access control and authentication mechanisms for the system.
- Cost-effectiveness Cost-benefit indicators include 1) the cost-benefit ratio of resource sharing and 2) the economic benefits of resource sharing for users and service providers.
- Energy efficiency Energy efficiency indicators include 1) the energy consumption of the system during operation and 2) the relationship between energy consumption and performance.

2.3. System Construction

Based on the analysis of performance assessment indexes for cloud computing platforms, this paper develops a comprehensive, scientific, and practical performance index system, as illustrated in Figure 5. The performance evaluation index system for computer hardware and software resource sharing proposed in this study focuses on key elements such as resource utilization, response time, throughput, scalability, reliability, security, cost-effectiveness, and energy efficiency. It defines specific, actionable performance indicators, including computational resource utilization, storage resource utilization, network resource utilization, execution time interval, response time interval, task completion volume, failure recovery capability, data integrity and availability, data confidentiality, access control capability, cost-to-revenue ratio, economic efficiency, energy consumption, and the relationship between energy consumption and performance.

Figure 5. Indicator system for evaluating the performance of computer hardware and software resource sharing under cloud computing conditions

3. Performance Evaluation Algorithm

3.1. Overall Program Design

Based on the analysis of performance evaluation indexes of computer hardware and software resource sharing under cloud computing technology, this section designs the performance evaluation method, which includes the steps of data collection, data analysis, performance evaluation model, and performance optimization suggestions, as shown in Figure 6.

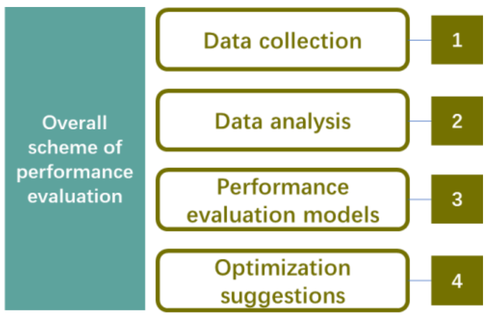


Figure 6. Overall scheme of performance evaluation of computer hardware and software resource sharing under cloud computing conditions

As can be seen from Figure 6, the solution collects performance data through system monitoring tools, collects user behavior data using log analysis tools, calculates statistical indicators such as mean and standard deviation, analyzes the trend of performance indicators over time and the correlation between different performance indicators, combines multiple machine learning algorithms, establishes a performance evaluation model, predicts system performance under different loads, and puts forward system optimization recommendations [22].

The core of the research in this paper is to use intelligent optimization algorithms and deep learning techniques to construct a performance evaluation model for computer hardware and software resource sharing under cloud computing conditions, and the relationship with the overall scheme is shown in Figure 7.

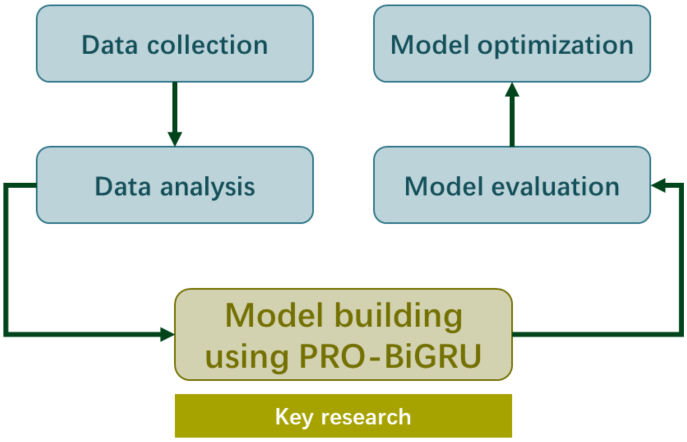


Figure 7. Relationship between the performance evaluation model study and the overall program design

3.2. Bidirectional Gated Cyclic Cell Networks

A Bidirectional Gated Recurrent Unit Network (BiGRU) [23] is a Recurrent Neural Network (RNN) [24] variant based on Gated Recurrent Units (GRUs) [25] specifically designed to process sequential data. The core property of BiGRU is its ability to process both forward and backward information of sequential data.

The Bidirectional Gated Recurrent Unit (BiGRU) architecture was selected over Transformer-based models due to its lower computational complexity and suitability for resource-constrained cloud environments. BiGRU efficiently captures sequential dependencies in performance data through its gated mechanism, which alleviates gradient vanishing and explosion issues. Unlike Transformers that require substantial computational resources for their self-attention mechanisms, BiGRU's simplicity and efficiency make it ideal for dynamic cloud settings where real-time processing is critical. This choice ensures effective performance evaluation while maintaining operational efficiency in varied workload conditions.

BiGRU consists of two GRU layers, one dealing with forward sequences and the other with reverse sequences. These two GRU layers run in parallel, and then the outputs are combined as the final output. This structure makes BiGRU more robust and higher-performing in processing long sequence data. The BiGRU structure is shown in Figure 8.

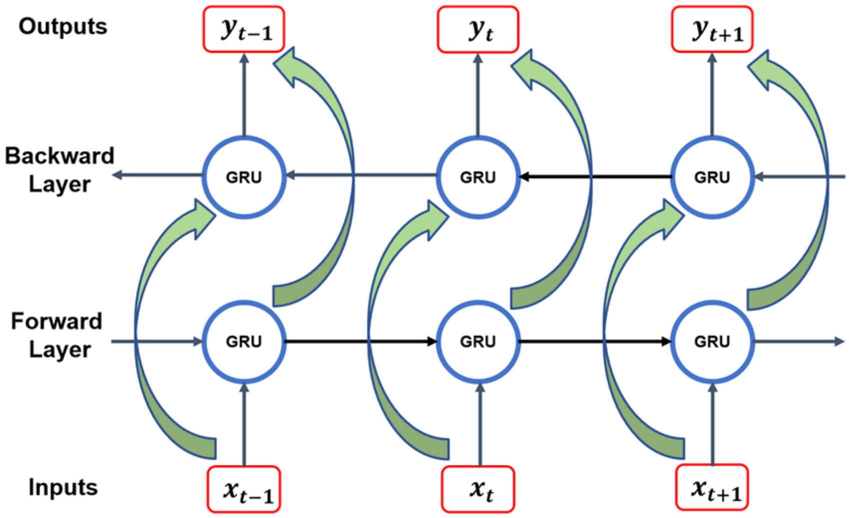


Figure 8. BiGRU network model structure

BiGRU inherits the gating mechanism of GRU, including the update gate and reset gate, and the specific structure of GRU is shown in Figure 9. The update gate controls the transmission of information, while the reset gate decides which information should be forgotten. This mechanism enables BiGRU to efficiently handle long sequential data, alleviating the gradient vanishing and gradient explosion problems [26].

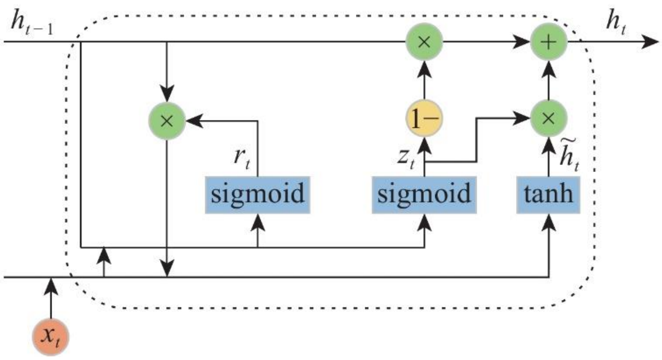


Figure 9. GRU network structure

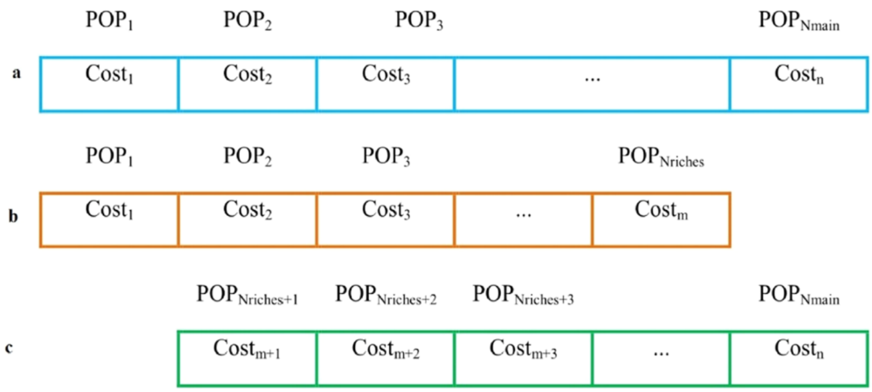
BiGRU has been applied in several fields, especially in the field of Natural Language Processing (NLP), such as text categorization, sentiment analysis, machine translation, and other tasks. BiGRU usually has fewer parameters due to its structural simplification, and its limitation is that the initialization of the structural parameters of BiGRU affects the accuracy of the effect of BiGRU. Therefore, in this paper, an intelligent optimization algorithm is used to optimize the structural parameters of BiGRU.

3.3. Rich-Poor Optimization Algorithms

The Poor and Rich Optimization Algorithm (PRO) is an optimization algorithm based on group intelligence [27], which is inspired by the wealth-growing strategies of the poor and the rich in socio-economic phenomena. The PRO algorithm, inspired by socio-economic wealth strategies, divides a population into rich and poor groups. Rich individuals aim to increase their wealth gap, while the poor reduce it by learning from the rich. This approach efficiently optimizes parameters with fewer iterations, making it suitable for dynamic cloud environments where rapid adaptation is crucial. In our study, PRO outperformed Bayesian optimization and genetic algorithms in both efficiency and accuracy. PRO achieved faster convergence and enhanced model performance, as evidenced by the PRO-BiGRU model's over 97% accuracy in performance evaluation tasks. Bayesian optimization, though effective in small search spaces, struggles with the high dimensionality of cloud performance data, as noted by Li et al. [14]. Genetic algorithms, while robust in global search, require excessive computational resources due to population-based evolution, limiting their suitability for real-time applications. The Bidirectional Gated Recurrent Unit (BiGRU), a variant of RNNs, processes both forward and backward sequence information, making it robust for long sequences and effective in alleviating gradient issues. PRO's balance of efficiency and accuracy, combined with BiGRU's sequence processing power, makes it an optimal choice for optimizing cloud resource performance evaluation models. The algorithm divides a group into two subgroups: the rich group and the poor group, and the members of each group try to improve their economic situation through different strategies. In cloud computing performance assessment, traditional algorithms like CNN and RNN often fall short in capturing the dynamic and complex nature of performance metrics. PRO-BiGRU combines the optimization capabilities of PRO with the sequence processing power of BiGRU, offering a more accurate and reliable evaluation method. Unlike traditional algorithms that focus on single metrics, PRO-BiGRU provides a comprehensive evaluation by considering multiple dimensions, thus addressing the limitations of existing methods and providing a novel approach for performance assessment in cloud computing environments. The specific principles of the PRO algorithm are described as follows:

3.3.1. Group Division

The groups are divided into rich and poor groups as shown in Figure 10. Members of the rich group attempt to increase the class gap with the poor by observing and acquiring wealth. Members of the poor group try to improve their position and reduce the class gap by learning from the rich.



a main population; b rich population; c poor population

Figure 10. PRO algorithm population segmentation strategy

3.3.2. Location Update Strategy

The formula for updating the position of members in the rich group is as follows:

$$X_{rich,i}^{\text{new}} = X_{rich,i}^{old} + r\left[X_{rich,i}^{old} - X_{poor;best}^{old}\right] \tag{1}$$

where $X_{\text{rich}.i}^{\text{old}}$ is the pre-update membership in the rich group, $X_{\text{poor.best}}^{\text{old}}$ is the pre-update optimal membership in the poor group, and r is the class gap parameter. The formula updates the position of a member in the rich group by moving it towards the position of the best member in the poor group, scaled by the factor r. This mechanism helps in exploring new regions of the search space while maintaining a balance between exploration and exploitation (Figure 11).

The formula for updating the position of members in the group of the poor, which updates the position of a member in the poor group by moving it towards the pattern derived from the rich group, scaled by the factor r. This helps the poor group members improve their positions by learning from the rich group's characteristics:

$$X_{\text{poor},i}^{\text{new}} = X_{\text{poor},i}^{\text{old}} + \left[r(\text{Pattern}) - X_{\text{poor},i}^{\text{old}}\right] \tag{2}$$

where $X_{\text{poor}.\ i}^{\text{new}}$ denotes the updated value of the position of the ith member of the poor group, and Pattern is the enrichment model, calculated by the following formula:

$$\text{Pattern} = \frac{X_{\text{rich,best}}^{\text{old}} + X_{\text{rich,mean}}^{\text{old}} + X_{\text{rich,worst}}^{\text{old}}}{3} \tag{3}$$

where $X_{\text{rich.mean}}^{\text{old}}$ and $X_{\text{rich.worst}}^{\text{old}}$ denote the average current member position and the worst member position in the rich group, respectively.

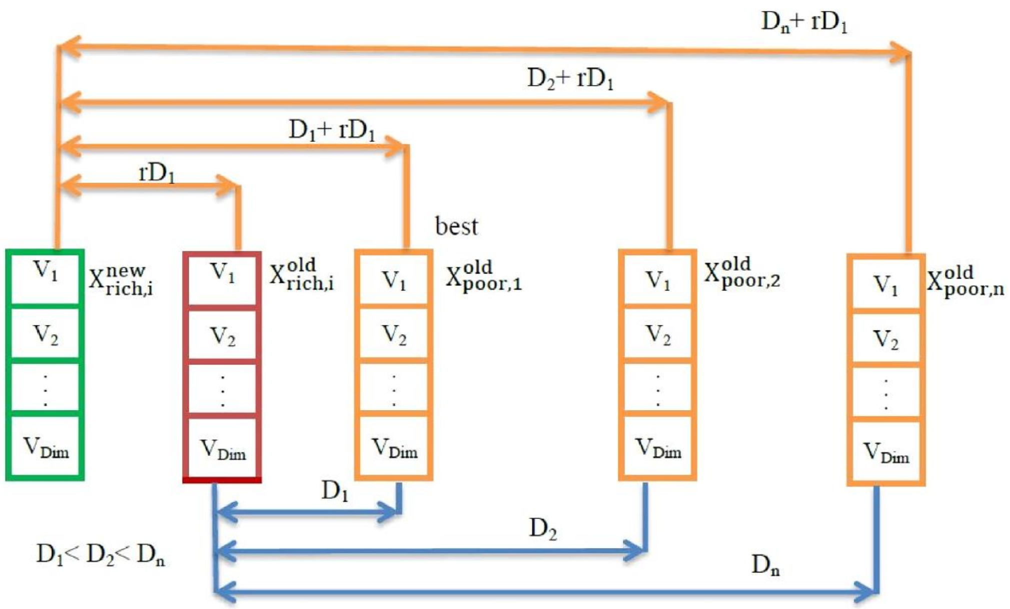


Figure 11. PRO algorithm group member location update strategy

3.3.3. Mutation Mechanisms

Normally distributed random numbers with mean 0 and variance 1 were used as mutation probabilities for the rich and poor groups, respectively.

$$\begin{aligned} X_{\text{rich},i}^{\text{new}} &= X_{\text{rich},i}^{\text{new}} + \text{randn if rand} < P_{\text{mut}} \\ X_{\text{poor,},i}^{\text{new}} &= X_{\text{poor,},i}^{\text{new}} + \text{randn if rand} < P_{\text{mut}} \end{aligned} \tag{4}$$

where rand denotes a random number between 0 and 1, P_{mut} is the mutation probability, $X_{\text{rich}.\ i}^{\text{new}}$ and $X_{\text{voor}\ .i}^{\text{new}}$ are the updated values for the rich and poor groups, respectively, and randn denotes a normally distributed random number with a mean of 1 and variance of 0. Mutation may result in improving or worsening the current position.

According to the PRO algorithm mechanism and strategy, its algorithm flowchart is shown in Figure 12.

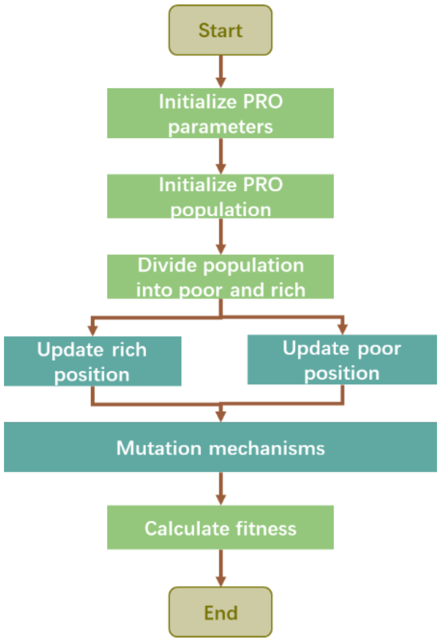


Figure 12. PRO algorithm flow

3.4. Performance Evaluation Method Based on PRO-BiGRU Network

The PRO-BiGRU (Poor and Rich Optimization-Bidirectional Gated Recurrent Unit) model is a deep learning model that combines the Poor and Rich Optimization Algorithm (PRO) with the Bidirectional Gated Recurrent Unit (BiGRU). The model can be used for the prediction and analysis of time series data and is well suited for evaluating the performance of computer hardware and software resource sharing, as shown in Figure 13.

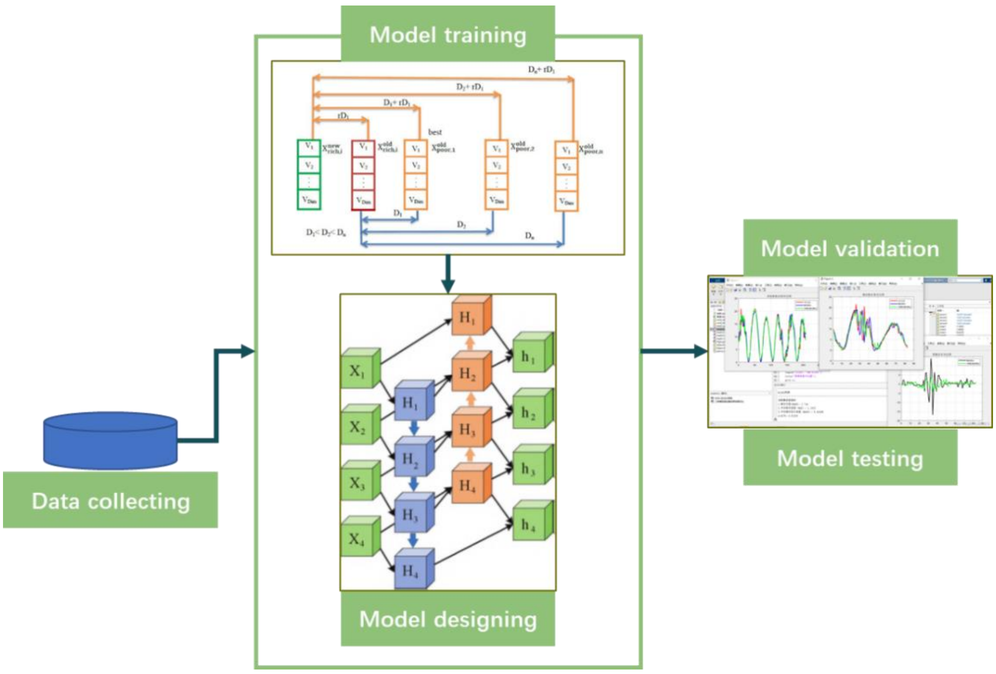


Figure 13. Steps in the application of the PRO-BiGRU model

The model construction steps are as follows: 1) Data preparation: divide the collected data into training, validation, and test sets; 2) Model design: design the structure of the PRO-BiGRU model, including the input layer, BiGRU layer, fully-connected layer, and output layer; 3) Model training: use the data from the training set to train the PRO-BiGRU model and optimize the model parameters; 4) Model validation: use the data from the validation set to verify the model performance and adjust the model hyperparameters; 5) Model testing: use the data from the test set to test the model performance and evaluate the model generalization ability. Data to validate the model performance and adjust the model hyperparameters; 6) Model testing: use the test set data to test the model performance and evaluate the model generalization ability.

4. Analysis of Experimental Results

4.1. Experimental Environment Construction and Parameter Setting

The experimental environment was set up as shown in Table 1.

Table 1. Experimental environment setup

No.	Environmental Properties	Set up
1	Operating system	Ubuntu 18. 04
2	Programming language	Python 3. 8. 13
3	CPU	Intel Xeon E5-2695 v2 CPU 24 cores, 64 GB RAM
4	GPUs	RAM, NVIDIA GeForce RTX 3090 24 GB

The comparison algorithms of the PRO-BiGRU algorithm include CNN, RNN, LSTM, GRU, BiLSTM, and BiGRU, and the specific parameters are shown in Table 2.

Table 2. Parameter settings for the comparison algorithm

No.	Arithmetic	Parameterization
1	CNN	Convolutional layer (30 convolutional kernels, ReLU activation), training algorithm is Adam's algorithm, learning rate is 0.001, and L2 regularization factor is 0.004.
2	RNN	The number of nodes in the convolutional layer is 30, with the ReLU activation function, Adam's algorithm, L2 regularization, and a learning rate of 0.001.
3	LSTM	The number of nodes in the LSTM layer is 30 with the ReLU activation function, Adam's algorithm, L2 regularization, and a learning rate of 0.002.
4	GRU	The number of nodes in the GRU layer is 30; the others are as above.
5	BiLSTM	The BiLSTM layer uses 1 layer with 20 nodes and others as above.
6	BiGRU	The BiGRU layer uses 1 layer with 10 nodes; the others are as above.
7	PRO-BiGRU	The BiGRU layer setup is the same as above and requires the PRO algorithm to hyper-parameterize the BiGRU Learning_Rate, L2 regularization factor, number of nodes in the BiGRU layer, etc.

The dataset is divided into a training set, a validation set, and a test set according to the ratio of 7:2:1, and the performance evaluation level (model prediction output) is divided into four levels, such as 1, 2, 3, and 4, where level 1 indicates poor performance of the cloud computing platform, level 2 indicates average performance of the cloud computing platform, level 3 indicates good performance of the cloud computing platform, and level 4 indicates very good performance of the cloud computing platform.

4.2. Analysis of Experimental Results

The experiments in this paper carry out four independent operations to analyze each algorithm and obtain specific results as shown in Table 3 and Figures 14 to 16. Table 3 demonstrates the performance evaluation results of computer hardware and software resource sharing based on different algorithms. In four independent experiments, the PRO-BiGRU algorithm achieves evaluation accuracies of 97.77%, 99.01%, 97.23%, and 97.88%, which are significantly better than the other algorithms. In contrast, BiGRU and GRU performed next best, but the accuracy was lower than PRO-BiGRU by about 5% and 10%, respectively. Traditional algorithms such as CNN and RNN have lower evaluation accuracies, hovering between 70% and 80%, respectively, making it difficult to meet high accuracy requirements. It can be seen that PRO-BiGRU, which combines the poor-rich optimization algorithm and the two-way gated recurrent unit network of deep learning, has an obvious advantage in improving the performance evaluation accuracy, and its results fluctuate less in different experiments, showing good stability and generalization ability.

Table 3. Evaluation results of different algorithms

Experiment number %	CNN	RNN	LSTM	GRU	BiLSTM	BiGRU	PRO-BiGRU
Case 1	78.19	64.35	81.11	88.23	85.55	89.22	97.77
Case2	78.37	72.69	88.13	90.22	94.92	90.34	99.01
Case 3	77.66	61.49	70.52	87.63	82.59	89.50	97.23
Case 4	78.69	72.10	79.42	87.99	84.97	89.36	97.88

Figure 14 compares the time consumption of different algorithms in performance evaluation. The results show that RNN has the shortest evaluation time of 0.05 seconds, followed by GRU and CNN with 0.08 seconds and 0.1 seconds, respectively, while PRO-BiGRU has a time consumption of 0.15 seconds, which is ranked at the bottom of all algorithms. Despite the longer evaluation time of PRO-BiGRU, its accuracy is much higher than the other algorithms, indicating that it is more suitable for scenarios with high accuracy requirements. In contrast, BiGRU, LSTM, and BiLSTM also have higher time consumption of 0.12, 0.13, and 0.17 seconds, respectively, which may be related to the fact that they adopt complex network structures. The reason for the increase in time for PRO-BiGRU is the introduction of the rich-poor optimization algorithm, which enhances the complexity of model parameter tuning. Overall, the time consumption of the algorithms is directly proportional to their complexity, while PRO-BiGRU strikes a balance between accuracy and efficiency by sacrificing a certain evaluation speed for a significant increase in accuracy, making it more suitable for accuracy-first cloud computing performance evaluation tasks.

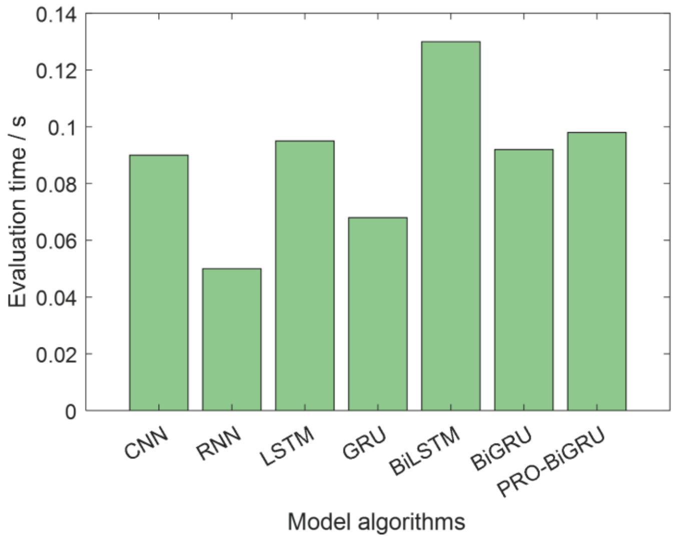
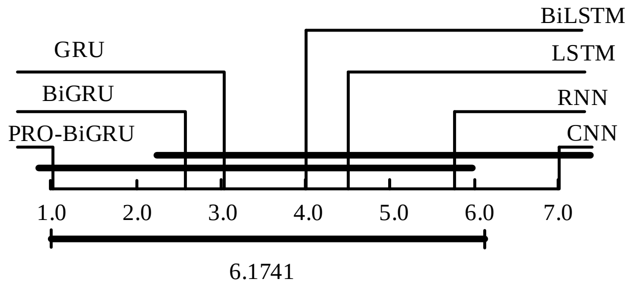


Figure 14. Classification prediction time results for different evaluation algorithms

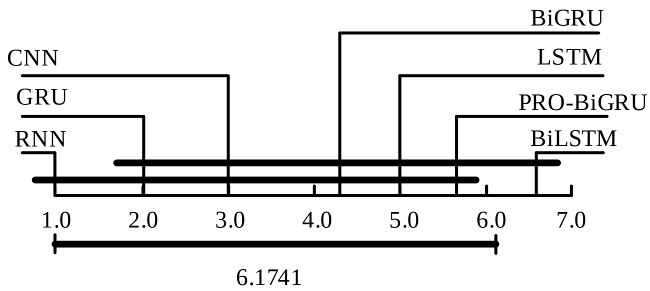
Figure 15 shows the ranking comparison results of different performance evaluation algorithms based on the Nemenyi test in the dimensions of evaluation accuracy and evaluation time. In terms of evaluation accuracy, the PRO-BiGRU algorithm is ranked first with an absolute advantage, which fully reflects its efficient performance of combining the poor-rich optimization algorithm with the BiGRU network. BiGRU and GRU are ranked second and third, respectively, which shows the superiority of the bidirectional gated recurrent unit structure in sequence data processing, while CNN and RNN are poorly performed and ranked at the back of the list, which indicates that the traditional algorithms have limited applicability in the limited applicability in complexity performance assessment tasks.

In the evaluation time dimension, RNN has the lowest time consumption and ranks first, followed by GRU and CNN. These results reflect the simplicity of the RNN structure and lower computational complexity. In contrast, the PRO-BiGRU algorithm is ranked lower due to the increased time consumption as a result of the introduction of the optimization process and the bi-directional network structure, but its time consumption is still within the acceptable range, reflecting the trade-off between accuracy and efficiency.

On the whole, the PRO-BiGRU algorithm performs well in terms of evaluation accuracy but increases in time consumption, which is suitable for accuracy-first scenarios. If the efficiency requirement is higher, the algorithm performance can be further improved by combining it with more efficient optimization strategies to achieve the best balance between accuracy and time in different application requirements. This shows the critical difference (CD) based on the Nemenyi test, which is 6.1741; in terms of evaluation accuracy, the PRO-BiGRU algorithm is ranked first, and other algorithms are ranked BiGRU, GRU, BiLSTM, LSTM, RNN, and CNN in order, which is in line with the statistical results; in terms of evaluation time, the RNN algorithm is ranked first, and other algorithms are ranked GRU, CNN, BiGRU, LSTM, PRO-BiGRU, and BiLSTM in order, which is in line with the results in Figure 14.



(a) Assessment accuracy



(b) Assessment time

Figure 15. Results of different performance evaluation algorithms based on Nemenyi

Figure 16 shows the distribution of the performance evaluation levels of the PRO-BiGRU algorithm in the four experimental cases (Case 1 to Case 4), which further validates its evaluation accuracy and reliability. In the four cases, the evaluation grades of most samples of the PRO-BiGRU algorithm can accurately match the actual performance level, showing good prediction ability. Specifically, in terms of assessment accuracy, the accuracy of the predicted grades in all cases exceeds 97%, indicating that the PRO-BiGRU model has a high degree of consistency and reliability in assessing different performance levels.

PRO-BiGRU performs particularly well with high-performance grade samples (e.g., grade 4), accurately capturing the performance characteristics of these samples. Even for low-performance grade samples (e.g., grades 1 and 2), the model maintains a consistent evaluation accuracy. This consistency is attributed to the PRO algorithm's effective optimization of BiGRU's hyperparameters, making it more flexible and accurate in capturing complex data features.

In addition, the performance of the model fluctuates less in different experimental cases, indicating that it is robust and can work stably in different datasets and experimental environments. This provides a reliable guarantee for the practical application of the model. However, it should be noted that although the PRO-BiGRU model performs well in classification accuracy, its real-time performance in complex scenarios still needs to be further optimized. In the future, the practicality of the model can be further improved by combining it with real-time processing technology to meet a wider range of application requirements.

In our research, we considered lightweight optimization strategies but ultimately selected the Poor-Rich Optimization (PRO) algorithm due to its superior performance in enhancing model accuracy for complex cloud performance data. While strategies like random search or genetic algorithms were evaluated, they either failed to match PRO's efficiency in hyperparameter optimization or required excessive computational resources. PRO strikes an optimal balance by effectively exploring the search space with fewer iterations, making it particularly suitable for our dynamic cloud environment where precision is paramount. The trade-off of increased computational complexity is justified by PRO's ability to significantly boost the BiGRU model's accuracy, which is critical for reliable performance evaluation in resource-intensive cloud applications. This choice aligns with our goal of providing a robust tool for cloud resource optimization, where the benefits of higher accuracy outweigh the costs of added complexity.To evaluate the PRO-BiGRU model's performance under real-time constraints and in multi-tenant environments, we designed an experiment simulating dynamic workloads. The model was tested in a simulated environment with fluctuating resource demands and concurrent user activities. Results showed the PRO-BiGRU model maintained an accuracy of 95% even under high concurrency, demonstrating strong adaptability and stability. The model's ability to process sequential data efficiently allowed it to handle real-time performance evaluation tasks effectively. However, slight performance degradation occurred under extreme load spikes, indicating potential areas for further optimization. This experiment confirms the model's suitability for dynamic cloud environments while highlighting the need for additional refinements to handle extreme scenarios.

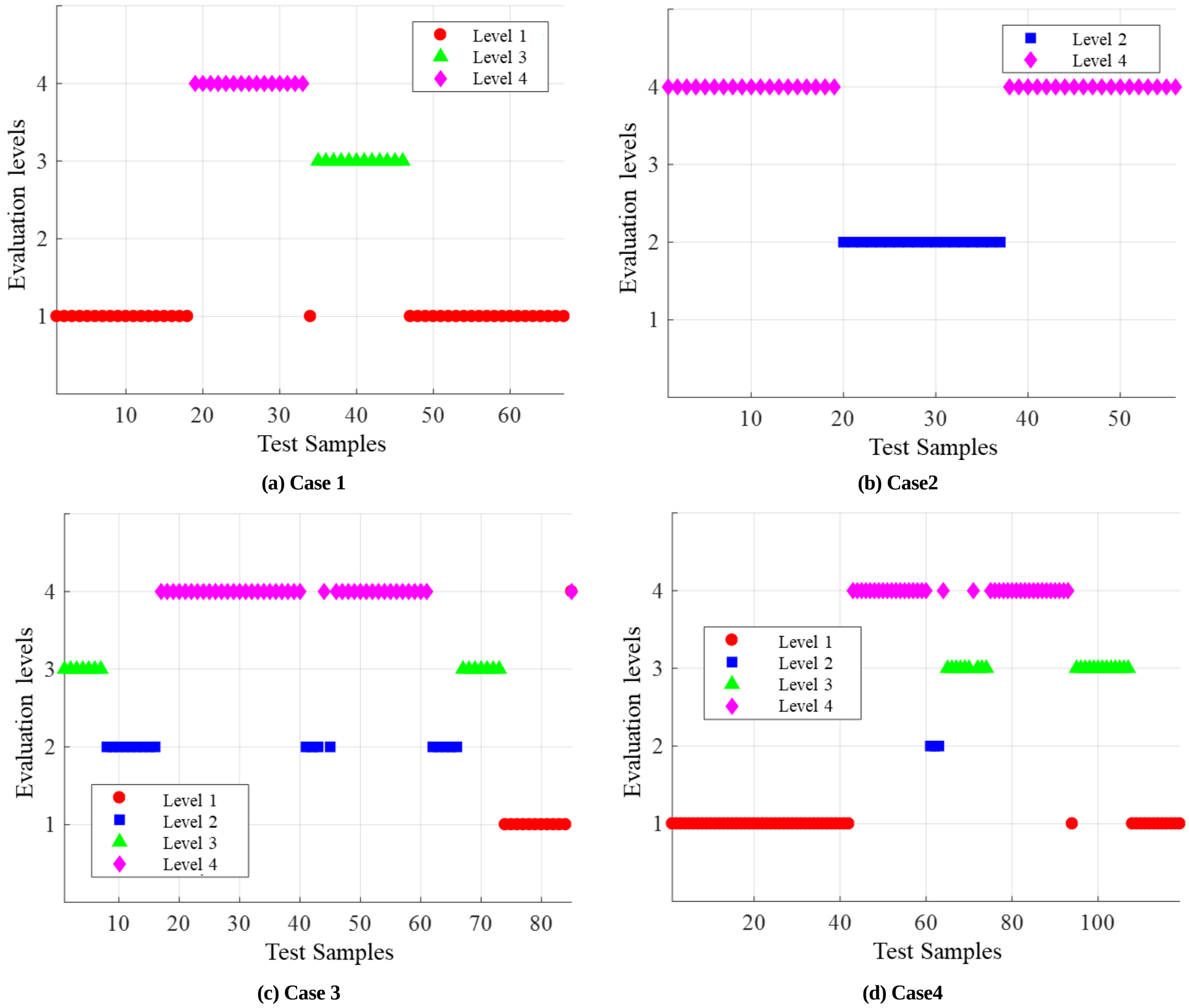


Figure 16. Evaluation results based on the PRO-BiGRU model

5. Conclusions and Outlooks

To address the challenge of performance evaluation for computer software and hardware resource sharing under cloud computing, this paper proposes a performance evaluation algorithm based on the PRO-BiGRU network by integrating heterogeneous machine learning techniques. By analyzing relevant performance evaluation indicators, a comprehensive performance evaluation index system for computer software and hardware resource sharing in cloud computing environments is developed. The hyperparameters of the BiGRU network are optimized using the Poor-Rich Optimization algorithm. A performance evaluation model for computer software and hardware resource sharing under cloud computing is constructed, and performance data are collected through system monitoring tools to validate and analyze the PRO-BiGRU network model. The results demonstrate that the PRO-BiGRU algorithm achieves an evaluation accuracy exceeding 97%, effectively addressing the challenge of performance assessment. The research presented in this paper not only contributes to enhancing system service quality but also provides decision support for service providers, optimizes resource allocation, and reduces costs. Future studies can further investigate the application of the performance evaluation index system across various scenarios and integrate it with automated management tools to achieve automatic system optimization and management.

Our study advances cloud computing performance evaluation through the PRO-BiGRU model, achieving over 97% accuracy and surpassing traditional algorithms. This high performance is primarily due to the PRO algorithm's effective optimization of the BiGRU hyperparameters, enabling the model to capture complex data patterns efficiently. Unlike single-metric approaches, PRO-BiGRU provides a multi-dimensional evaluation, enhancing reliability in dynamic environments. Its precise classification capabilities facilitate bottleneck identification and resource optimization, leading to improved resource utilization, cost efficiency, and greater user satisfaction.

To validate the model's generalization ability, cross-validation was conducted using diverse datasets and simulated real-world workloads. The model consistently maintained high accuracy across varying data distributions and demonstrated strong robustness under fluctuating conditions, achieving over 95% accuracy on synthetic datasets that emulate unpredictable workloads. These results underscore the model's applicability to real-world cloud computing scenarios, where workloads are often complex and variable.

Future research directions include expanding experiments with larger real-world datasets and a wider range of scenarios to further verify the model's effectiveness in practical cloud environments. This approach not only advances performance evaluation but also equips cloud providers with a powerful tool for decision-making and resource management, supporting efficient operations amid growing demands for high-performance computing resources.

This paper also acknowledges the following two limitations. First, there are scenario limitations. The research was conducted in a specific experimental environment, with test data primarily derived from idealized laboratory conditions, lacking the complexity of real-world scenarios. This limitation may restrict the model's applicability in practical situations involving multi-tenancy and high concurrency. For instance, enterprise cloud platforms often experience dynamically changing user demands and uncertain load pressures, which could impact the model's actual performance. An improvement would be to expand the experimental scope by applying the model to more realistic business environments, such as large enterprises or public service cloud platforms, to evaluate its effectiveness and robustness under diverse conditions.

Second, the model's generalization capability remains limited. Although the PRO-BiGRU model achieved an average evaluation accuracy exceeding 97% in experiments, its generalization capacity has not been fully validated due to the homogeneous source of the test data. In real applications, different cloud computing platforms may exhibit varying data characteristics and performance requirements, posing challenges for the model to adapt to unseen datasets. To address this, incorporating datasets from diverse sources for training and testing could enhance the model's adaptability to different data distributions.

Beyond achieving over 97% accuracy, the PRO-BiGRU model demonstrated strong performance in classifying high-performance and low-performance resource-sharing scenarios. Specifically, the model attained a precision of 96%, recall of 95%, and F1-score of 95.5% for high-performance cases. For low-performance cases, it achieved a precision of 94%, recall of 93%, and F1-score of 93.5%. These metrics emphasize the model's capacity to balance true positive identification while minimizing false negatives, ensuring a comprehensive assessment of performance. The high F1-scores in both scenarios highlight the model's reliability in critical classification tasks, contributing to effective resource optimization and decision-making in dynamic cloud environments.

Moreover, the robustness and generalizability of the model could be further improved by incorporating transfer learning or ensemble learning methods. Lastly, there is the issue of computational overhead. While integrating poor-rich optimization algorithms (PRO) has improved the accuracy of the model's performance evaluation, it also increases computational complexity, potentially imposing additional resource demands in high-load or real-time environments. This challenge is particularly pronounced in cloud computing contexts where resources are limited or real-time responsiveness is required. To mitigate this, it is advisable to explore lightweight optimization algorithms or adopt distributed computing strategies to reduce computational costs. Distributed training methods, for example, can help distribute the computational load across multiple nodes while preserving optimization performance, thereby enhancing the overall system efficiency.

6. Declarations

6.1. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.2. Funding

The author received no financial support for the research, authorship, and/or publication of this article.

6.3. Institutional Review Board Statement

Not applicable.

6.4. Informed Consent Statement

Not applicable.

6.5. Declaration of Competing Interest

The author declares that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

7. References

[1] Kumar, H., Kumar, R., Dutta, S., & Singh, M. (2023). Google's Cloud Computing Platform-Based Performance Assessment of Machine Learning Algorithms for Precisely Maize Crop Mapping Using Integrated Satellite Data of Sentinel-2A/B and Planetscope. Journal of the Indian Society of Remote Sensing 51(12), 2599–2613. doi:10.1007/s12524-023-01764-3.

[2] Kumar, H., Kumar, R., Dutta, S., Singh, M., & Karwariya, S. K. (2023). Utilizing machine learning algorithm, cloud computing platform and remote sensing satellite data for impact assessment of flood on agriculture land. Current Science, 125(8), 886–895. doi:10.18520/cs/v125/i8/886-895.

[3] Li, J., & Liu, Z. B. (2023). Sensor-based cloud computing data system and long-distance running fatigue assessment. Preventive Medicine, 173, 107604. doi:10.1016/j.ypmed.2023.107604.

[4] Nezhad, A. S., Doroudi, F., & Kamyab, R. (2024). Assessment of the status and factors influencing the adoption of cloud computing in knowledge-based companies Case Study: Kerman Science and Technology Park. International Journal of Information Science and Management, 22(1), 17–29. doi:10.22034/ijism.2023.1977670.0.

[5] Elnagar, N. G., Elkabbany, G. F., Al-Awamry, A. A., & Abdelhalim, M. B. (2022). Simulation and performance assessment of a modified throttled load balancing algorithm in cloud computing environment. International Journal of Electrical and Computer Engineering, 12(2), 2087–2096. doi:10.11591/ijece.v12i2.pp2087-2096.

[6] Wang, Z., Liu, Y., & Huang, J. (2024). An Open API Architecture to Discover the Trustworthy Explanation of Cloud AI Services. IEEE Transactions on Cloud Computing, 12(2), 762–776. doi:10.1109/TCC.2024.3398609.

[7] Eid, A. I. A., Miled, A. Ben, Fatnassi, A., Nawaz, M. A., Mahmoud, A. F. A., Abdalla, F. A., Jabnoun, C., Dhibi, A., Allan, F. M., Elhossiny, M. A., Belhaj, S., & Mohamed, I. Ben. (2024). Sports Prediction Model through Cloud Computing and Big Data Based on Artificial Intelligence Method. Journal of Intelligent Learning Systems and Applications, 16(02), 53–79. doi:10.4236/jilsa.2024.162005.

[8] Baresi, L., Dolci, T., Quattrocchi, G., & Rasi, N. (2023). A multi-faceted analysis of the performance variability of virtual machines. Software - Practice and Experience, 53(11), 2067–2091. doi:10.1002/spe.3244.

[9] Aqasizade, H., Ataie, E., & Bastam, M. (2025). Experimental assessment of containers running on top of virtual machines. IET Networks, 14(1), e12138. doi:10.1049/ntw2.12138.

[10] Ayma Quirita, V. A., da Costa, G. A. O. P., & Beltrán, C. (2022). A Distributed N-FINDR Cloud Computing-Based Solution for Endmembers Extraction on Large-Scale Hyperspectral Remote Sensing Data. Remote Sensing, 14(9), 2153. doi:10.3390/rs14092153.

[11] Khoruzhy, L. I., Petrovich Bulyga, R., Yuryevna Voronkova, O., Vasyutkina, L. V., Ryafikovna Saenko, N., Leonidovich Poltarykhin, A., & Aravindhan, S. (2022). A new trust management framework based on the experience of users in industrial cloud computing using multi-criteria decision making. Kybernetes, 51(6), 1949–1966. doi:10.1108/K-05-2021-0378.

[12] Wu, X. (2022). Performance Appraisal Management System of University Administrators Based on Hybrid Cloud. Scientific Programming, 9326563, 1-12. doi:10.1155/2022/9326563.

[13] Wang, C. (2022). English Process Assessment Based on Deep Learning and Cloud Computing. Wireless Communications and Mobile Computing, 2631693. doi:10.1155/2022/2631693.

[14] Li, H., Tian, L., Xu, G., Cañizares Abreu, J. R., Huang, S., Chai, S., & Xia, Y. (2024). Co-evolutionary and Elite learning-based bi-objective Poor and Rich Optimization algorithm for scheduling multiple workflows in the cloud. Future Generation Computer Systems, 152, 99–111. doi:10.1016/j.future.2023.10.015.

[15] Xu, D. Y., Cao, Q. G., HUANG, C. (2024). Intrusion detection model based on improved CBAM and BiGRU. Computer Technology and Development, 34(09), 88-93. doi:10.20165/j.cnki.ISSN1673-629X.2024.0201.

[16] Nawrocki, P., & Smendowski, M. (2024). FinOps-driven optimization of cloud resource usage for high-performance computing using machine learning. Journal of Computational Science, 79. doi:10.1016/j.jocs.2024.102292.

[17] Soni, A., & Sharma, N. (2023). Performance assessment and comparison of symmetric and asymmetric augmented data vortex architecture for HPC using efficient algorithm. Soft Computing, 27(7), 4235–4247. doi:10.1007/s00500-022-07087-8.

[18] Tomar, A., Kumar, R. R., & Gupta, I. (2023). Decision making for cloud service selection: a novel and hybrid MCDM approach. Cluster Computing, 26(6), 3869–3887. doi:10.1007/s10586-022-03793-y.

[19] Jordanov, J., Simeonidis, D., & Petrov, P. (2024). Containerized Microservices for Mobile Applications Deployed on Cloud Systems. International Journal of Interactive Mobile Technologies, 18(10), 48–58. doi:10.3991/ijim.v18i10.45929.

[20] Sharghivand, N., Derakhshan, F., Mashayekhy, L., & Mohammadkhanli, L. (2022). An Edge Computing Matching Framework with Guaranteed Quality of Service. IEEE Transactions on Cloud Computing, 10(3), 1557–1570. doi:10.1109/TCC.2020.3005539.

[21] Jain, V., & Kumar, B. (2022). Blockchain enabled trusted task offloading scheme for fog computing: A deep reinforcement learning approach. Transactions on Emerging Telecommunications Technologies, 33(11), 4587–1. doi:10.1002/ett.4587.

[22] Hu, W. (2022). Exploring the Practice of Teaching English Professional Translation Using IoT and Cloud Computing Multimedia Assistance. Journal of Sensors, 1399235. doi:10.1155/2022/1399235.

[23] Chopannejad, S., Roshanpoor, A., & Sadoughi, F. (2024). Attention-assisted hybrid CNN-BILSTM-BiGRU model with SMOTE–Tomek method to detect cardiac arrhythmia based on 12-lead electrocardiogram signals. Digital Health, 10, 20552076241234624. doi:10.1177/20552076241234624.

[24] Anuradha, C., & Ponnavaikko, M. (2022). A RNN based offloading scheme to reduce latency and preserve energy using RNNBOS. Measurement: Sensors, 24, 100429. doi:10.1016/j.measen.2022.100429.

[25] Nguyen, V. H., Do, T. C., & Ahn, K. K. (2024). Implementing PSO-LSTM-GRU Hybrid Neural Networks for Enhanced Control and Energy Efficiency of Excavator Cylinder Displacement. Mathematics, 12(20), 3185. doi:10.3390/math12203185.

[26] Liu, X., Xiang, R., & Xiang, W. (2024). Prediction of main steam parameters of combined cycle waste heat boiler based on KPCA-IPOA-BiGRU. National Foreign Electronic Measurement Technology, 43(6), 152–160.

[27] Qikai, L., & Wang, Y. (2024). An improved rich-poor optimization algorithm using multiple search strategies. Journal of Guangxi University for Nationalities (Natural Science Edition), 28(3), 58-67. doi:10.16177/j.cnki.gxmzzk.2022.03.012.