Cloud Gaming Resource Management Platform Based on Edge Intelligence

Hu Yang(✉), Xie Yunsong, Li Jiaye, Su Xunjie, Wang Maoyu, Li Guanlin, and Lin Shangjing

School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, China
{hu.yang,xie.yunsong,li.jiaye,su.xunjie,wang.maoyu,li.guanlin,lin.shangjing}@bupt.edu.cn

Abstract. This study thoroughly explores the rapid development of edge intelligence, emphasizing the synergy between cloud computing and edge computing to significantly enhance data processing efficiency. It highlights the advantages of edge intelligence-based cloud gaming platforms over traditional cloud gaming platforms. Traditional resource pooling techniques perform poorly and incur high costs during fluctuating user demands. To address this, we introduce edge intelligence to cloud computing and, employing the LSTM algorithm, construct a predictive model for resource pooling, demonstrating its efficiency and adaptability. The innovation of this paper lies in proposing a wireless communication traffic prediction model based on federated learning within a distributed architecture. Individual grid traffic prediction models are trained synchronously, and the central cloud server uses Jensen-Shannon (JS) divergence to select grid traffic models with similar distribution. It utilizes a federated averaging algorithm to merge parameters of grid traffic models with comparable distribution, aiming to enhance model generalization while accurately characterizing local traffic patterns. Additionally, the paper elaborates on optimizing resource caching through PID automatic control algorithms in the context of pooling strategies, addressing sudden spikes and drops in traffic.

Keywords: Edge Intelligence · Federated Learning · Pooling Techniques · PID

1 Introduction

With the advent of the 5G era, cloud gaming has undergone rapid development and maturation. Fundamentally, cloud gaming is an interactive online video stream, where games run on the server-side, and the rendered media is compressed and sent to users. Cloud gaming distinguishes itself from traditional

Supported by Research Innovation Fund for College Students of Beijing University of Posts and Telecommunications

X. Chen et al. (Eds.): IoTaaS 2023, LNICST 585, pp. 37–57, 2025.
https://doi.org/10.1007/978-3-031-70507-6_4

games by reducing user costs, enhancing content experience, ensuring cross-platform play, and countering piracy and cheats.

However, traditional cloud computing (centralized cloud model) faces challenges in terms of bandwidth, latency, connection quality, resource allocation, security, and more. To address the dilemmas posed by applications and scenarios that traditional cloud infrastructure may not adequately meet, the concept of edge computing has emerged. Edge computing involves shifting some of the capabilities of cloud computing from centralized data centers to the network edge. This creates a high-performance, low-latency, and high-bandwidth service environment, accelerating the response speed of various content, services, and applications in the network, providing consumers with uninterrupted high-quality network experiences.

Additionally, a key technology in cloud gaming is resource pooling. The pooling process involves two steps: first, predicting game traffic, and second, allocating cloud gaming cache resources based on these predictions. Traditional techniques, whether presetting queue lengths or dynamically adjusting them, fail to meet the demands of fluctuating traffic. Therefore, based on the edge intelligent time series prediction model, this paper introduces the PID algorithm to achieve real-time resource allocation. Compared to existing models, the newly constructed model in this paper exhibits higher adaptability. It demonstrates greater generalization in scenarios where game datasets frequently switch, improving pooling hit rates and game launch speeds, enhancing user experience, and reducing wastage.

The main contributions of this paper are:

1. This article utilized a real dataset provided by a collaborative industry-academic project to implement time series forecasting of game traffic using the LSTM algorithm. To enhance collaborative model training with data from multiple regions, this paper introduces a wireless communication traffic prediction model based on federated learning within a distributed architecture. Merging model parameters trained in different regions onto a central server aims to improve predictive performance. Experimental results indicate that the aggregated model outperforms individual models prior to federation.
2. Due to the inherent time-delay characteristics of time series forecasting based on historical data, there is a potential for slow responsiveness to sudden surges or drops in traffic, which may fail to meet user experience requirements or result in resource waste. Therefore, the paper proposes a resource pooling allocation model based on the PID automatic control algorithm. Simultaneously, to achieve the goal of enhancing user experience while reducing enterprise costs, the paper establishes a comprehensive evaluation metric to assess the resource pooling allocation model.

2 Related Research

2.1 Cloud Gaming

Cloud gaming is a novel gaming approach based on cloud computing. Under the cloud gaming operational mode, all games run on the server side, which compresses the rendered game scenes and sends them to the users over the network. Client devices do not require any high-end processors or graphics cards; basic video decompression capability suffices for a smooth gaming experience [1]. The concept of cloud gaming was first introduced by the Finnish company G-cluster and is also known as game-on-demand, a technology rooted in cloud computing [2].

Cloud gaming operates the game on the server side. When game updates are required, only a single upgrade on the server side is needed, eliminating waiting times on the client side and thereby enhancing the user experience. The process of users connecting to the cloud gaming service is as follows: firstly, the game client connects to the designated port server, which then processes the user's connection request [3]. Physical machines select an appropriate cloud gaming server based on an algorithm. Subsequently, the interface server provides the client with the IP address of the cloud gaming machine and a dynamic key [4]. Lastly, upon receiving the video streams, the client decompresses it and plays the game video on the cloud gaming client.

2.2 Mobile Edge Computing

Mobile Edge Computing (MEC) offers IT services and cloud computing capabilities close to users by leveraging wireless access networks, creating a telecommunication-grade service environment characterized by high performance, low latency, and high bandwidth [5]. It can reduce client waiting times through caching and place detachable computing tasks on edge nodes to alleviate network stress and computational loads on data centers [6].

Another significant application of Mobile Edge Computing involves utilizing MEC servers at the edge to process computing tasks, thereby offloading computational tasks from data centers. For user data uploaded under base station coverage, the MEC servers can filter and pre-aggregate this data, mitigating the computational pressure on data centers [7]. Within the design framework of edge computing, apart from the conventional content caching distribution and offloading of computational tasks from data centers, each base station's MEC server also supports a generic IT platform capable of interacting with software. Software service providers can deploy or withdraw edge computing services based on this platform to optimize their software service quality and performance [8].

2.3 Resources Pooling Technology

In the cloud gaming environment, resource pooling technology significantly influences the processing and scheduling of server-side cloud resources, thereby

impacting game response latency and user experience. To address this issue, researchers proposed a cloud gaming resource allocation method based on predicting game duration. By predicting game duration using users' historical game data, resources are pre-pooled, speeding up response times and reducing latency [9]. Subsequently, researchers approached the problem from a different perspective and proposed a novel cloud gaming resource allocation strategy. Based on game attribute tags, they constructed game content feature vectors, computed recommendation indices, and performed dynamic corrections. This approach determined the initial deployment scale for cloud game content, pre-pooling resources to ensure the overall business load of the cloud game server remains balanced and stable during peak times, thereby improving key business metrics [10].

However, the dynamic nature of the cloud gaming system's load means that a one-size-fits-all resource allocation method is elusive. Keeping this challenge in mind, researchers introduced a cloud gaming adaptive resource allocation method based on reinforcement learning. By utilizing the Q-learning model in reinforcement learning, it can adaptively implement resource allocation, making optimal resource pooling decisions and reducing response latency [11].

2.4 Traffic Forecasting Models

In the digital age, traffic forecasting models play a crucial role in predicting traffic changes and optimizing resource planning. Time series analysis is a common method, exemplified by a model known for capturing data trends and patterns and widely utilized. Some researchers applied this model to forecast enterprises' free cash flow, obtaining a relatively accurate forecast value to gauge enterprise value.

Nevertheless, traditional time series models encounter limitations, requiring substantial historical data and exhibiting suboptimal performance when confronted with anomalies or irregular fluctuations. To address this challenge, machine learning methods are frequently employed in traffic forecasting. For instance, a group of researchers proposed a network traffic forecasting method based on a combined model. By integrating traditional time series analysis with machine learning, they developed a hybrid model enhancing prediction accuracy. Similarly, researchers utilized techniques based on technology for network traffic intrusion detection. Leveraging the features of technology, they extracted data characteristics, achieving higher accuracy rates in network traffic intrusion detection. Additionally, some researchers introduced a traffic forecasting method, enhancing a particle swarm algorithm to optimize neural networks. Capitalizing on the self-similarity and predictability of network traffic, they devised a superior neural network forecasting model.

3 Analysis of Traffic Characteristics

This article uses all cloud game data of a certain company in a certain month to provide the access time of different games under different service operators in Internet Data Center (IDC) computer rooms in different regions.

3.1 Data Processing

This article processes the original data to facilitate subsequent cache prediction. First, clean the data, complete the missing time series data, and delete spaces and illegal characters in the data to ensure the tidiness and effectiveness of the data. Next, time slicing and integration are performed to resample and integrate the data of the same region, the same computer room, and the same game at a 5-min granularity to obtain the number of user visits for every 5 consecutive minutes. The 24-h user access traffic under different IDCs is plotted as shown in Fig. 1. It can be found from the figure that the traffic data at noon and evening is relatively large, while the traffic data in the morning and afternoon is relatively small, which is related to people's work and rest. It shows that traffic data is highly correlated with time.

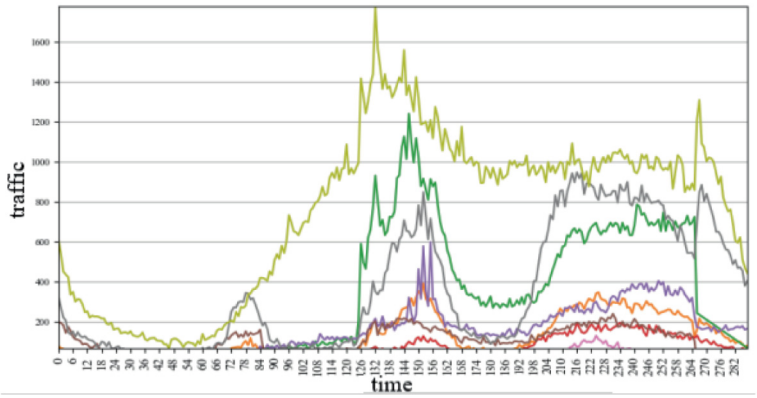

Fig. 1. Traffic changes every 5 min in different IDCs on a certain day

In order to accurately predict the cache pooling of cloud game resources below, this section conducts data analysis on game traffic for different IDC computer rooms, different games and different service operators. From the figure, it can be found that the traffic of different IDC computer rooms The difference is still quite large, and need to conduct a more detailed analysis.

3.2 Analysis of Data Traffic in Different IDC Computer Rooms

First, in order to explore whether there is a certain correlation in the data under different IDC computer rooms, the total number of single-day user visits under 10 different IDCs (from different provinces) was averaged, and the correlation coefficient matrix was made and visualized to obtain the average traffic of different IDCs. The correlation between them is shown in Fig. 2, in which the horizontal and vertical coordinates respectively represent 10 IDC computer rooms in various places.

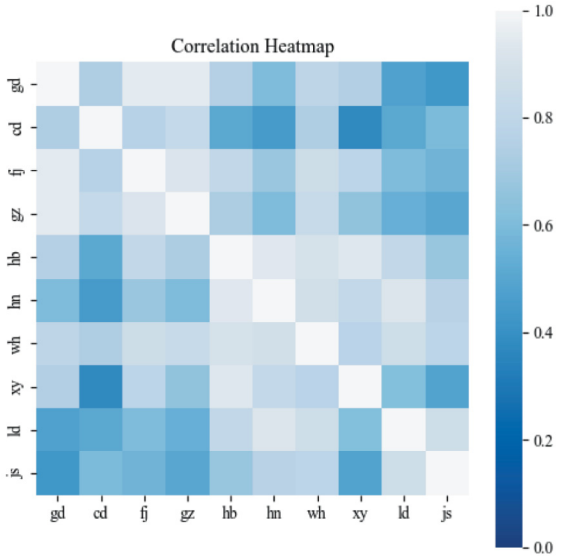

Fig. 2. Correlation between 24-h average traffic changes of different IDCs within a day

As can be seen from Fig. 2, the correlation between IDC traffic changes in different provinces within a single day is very high, and most of the correlation coefficients are above 0.8. Based on actual analysis, under the unified time system, the work and rest of users in each province are basically the same, so the data correlation is high, and the subtle differences will be more prominent under large time scale changes. The high correlation between traffic data is the prerequisite for applying transfer learning to the model.

Secondly, select the data of the same service operator under the base stations of two different provinces, and depict the traffic characteristics between them, as shown in Fig. 3: as can be seen from the box plot in Fig. 3, the overlap of the same service operator under different IDCs is small, the traffic difference is large, and the correlation is low.

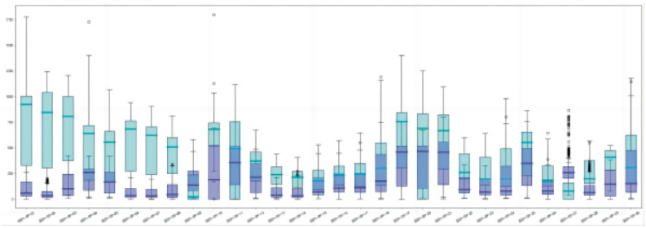

Fig. 3. Traffic characteristics of Region 1 and Region 2 under the same service operator

3.3 Traffic Data Analysis of Different Games

Analyzing the traffic data of different games, Fig. 5 shows the total traffic changes of different games within 30 days.

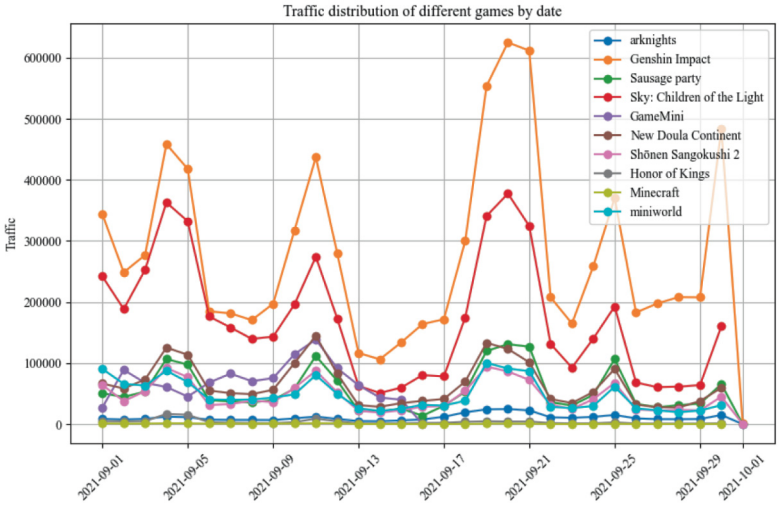

Fig. 4. Single-day total traffic characteristics of Top10 games

In Fig. 4, the games 'Genshin Impact' and 'Light Encounter' have the largest total traffic, and both reached their peak traffic of the month from the 19th to the 21st. The traffic change trends of different games within 30 days are generally similar. There will be a significant increase in traffic during holidays, while traffic is at a low point on weekdays. Special events and activities in the game will lead to sudden increases and decreases in game traffic. The traffic data of different games show a certain degree of correlation and high volatility in the time dimension.

3.4 Data Traffic Analysis of Different Service Operators

Analyzing the traffic data of different service operators, Fig. 6 shows the traffic changes of the top 10 service operators in total traffic within 30 days. The traffic change trend is similar to Fig. 5. The traffic changes of different service operators show high correlation and high volatility.

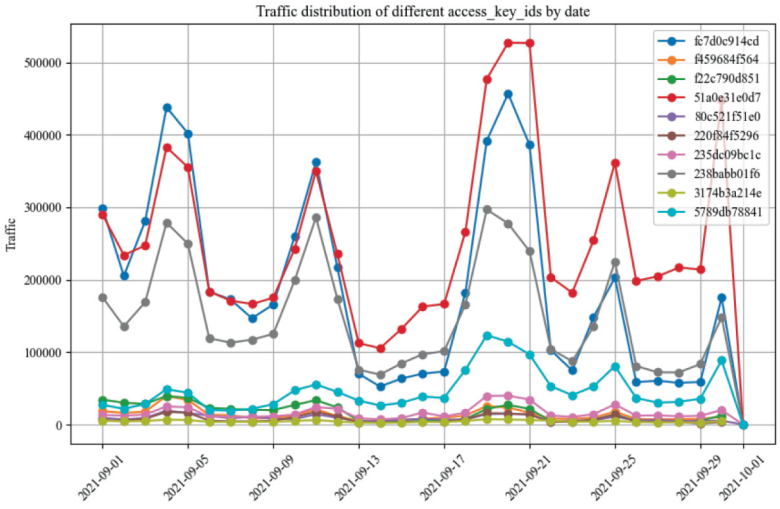

Fig. 5. Traffic characteristics among the top 10 service operators by day

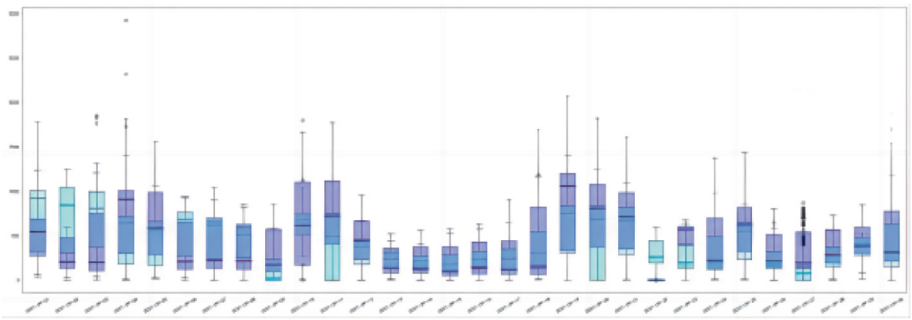

Fig. 6. Traffic characteristics of Top1 & Top2 IDC service operators

Further, select the service operators with Top1 & Top2 traffic in an IDC computer room within the top 10 total traffic. The traffic relationship characteristics between the two are shown in Fig. 6: It can be seen from the box plot that the traffic overlap of the two service operators in this area is relatively large,

and the overall traffic is basically similar, with a certain statistical correlation. In addition, the traffic of both service operators shows significantly higher peaks during holidays compared with weekdays.

4 Cloud Gaming Demand Prediction Model

4.1 Federated Learning Within a Distributed Architecture

Training deep learning models using traffic data from multiple enterprises in different IDC (Internet Data Center) facilities across various regions enables the model to learn more relevant features of the traffic data, thereby further improving the predictive accuracy of the model. However, in practical applications, due to concerns about data privacy, it is not feasible for traffic data between different enterprises to be openly shared for model training. Therefore, in this section, this article employ federated learning algorithms. Models trained in different IDC facilities are uploaded to a central server for collaborative training. The trained identical model is then distributed back to each IDC facility for local training, achieving an enhancement in the predictive performance of the model.

Federated learning adopts a decentralized approach to handle collaborative learning tasks. In this framework, a central server is responsible for coordinating learning objectives and aggregating models, while multiple client nodes use their local datasets to train local models. This method combines accuracy and efficiency while satisfying user privacy requirements. Assuming there is a dataset with N local nodes D_1,...,D_i,...,D_N, and $D_i := |D_i|$ represents the number of data samples each node possesses, the federated learning algorithm aims to minimize the expression:

$$\min F(W) := \sum_{i=1}^{N} \Psi_i F_i(W) \tag{1}$$

where W represents the global model weights, and $\psi_i = D_i / \sum_{i=1}^{N} D_i$ is the weight of the model during federated learning aggregation. The local objective function $F_i(W)$ is employed instead of the global objective function $F(W)$. The following describes the role of the local objective function $F_i(W)$. $F_i(W)$ typically assesses the local empirical risk $p^{(i)}$ that may arise due to different data distributions on nodes. $F_i(W)$ is defined using cross-entropy:

$$\min F_i(W) = -\sum_{j=1}^{C} p^{(i)}(y = j) E_{x[y=j]} \left[\log f_j(x, w)\right] \tag{2}$$

where $f_j(x, w)$ represents the probability of data sample being classified as the jth class by the specified model w, and $p^{(i)}(y = j)$ represents the data classification on node i for class $j \notin [C]$.

In a typical federated learning setup, participating nodes use the same configuration for local training. In each update round, a subset of the total node set, denoted as S_t ($|S_t| = K \ll N$), is selected, and the global model $W(t-1)$

from the previous iteration is sent to the selected nodes. Each node involved in federated learning performs stochastic gradient descent (SGD) to optimize its respective local objective function $F_i(W)$:

$$w_i(t) = w(t-1) - \eta \nabla F_i\big(w(t-1)\big) \tag{3}$$

where η represents the learning rate and the gradient at node i. The equation provides the general principles of SGD optimization, where $w_i(t)$ can be the result of one or multiple local updates of SGD. In the scenario below, SGD is applied to a small dataset of size B, and thus the local dataset is updated $\tau = \frac{D_i}{B} \times E$ times, where D_i and E are the number of samples trained on the node and the number of local training rounds, respectively.

Afterwards, these nodes update the models trained locally to the central server, which aggregates this data and updates the global model.

$$\Delta(t) = \sum_{i=1}^{|S_i|} \psi_i \Delta_i(t) \tag{4}$$

$$W(t) = W(t-1) + \Delta(t) \tag{5}$$

4.2 Transfer Learning

Due to the asynchronous construction of IDC (Internet Data Center) facilities in different regions, when a new IDC facility needs to be established in a new location, training a traffic prediction model for that IDC facility requires collecting a substantial dataset. This is often impractical for recently commissioned IDC facilities. The lack of effective traffic prediction methods for IDC facilities can lead to resource wastage or network congestion. Therefore, employing transfer learning techniques to migrate pre-trained model parameters from IDC facilities in other regions to the newly deployed IDC facility's traffic prediction model can significantly reduce training time and save costs.

Transfer learning is a form of machine learning in which a well-trained machine learning model A is obtained by training on a known dataset. The parameters of this model A are saved, and when developing another machine learning model B, there is no need to start training from scratch. Instead, model B is trained based on the foundation of model A, simplifying the training process and reducing training costs. The prerequisite for successful transfer learning is that there should be some similarity between the datasets and the training models of both models. It is through this similarity that a bridge is constructed from old knowledge to new knowledge, allowing for faster and more effective learning of new knowledge.

5 Pooling Resource Allocation Model

5.1 Construction of Comprehensive Evaluation Metrics

After achieving accurate predictions of cloud gaming traffic data, it is essential to proactively manage the pooling resource allocation on the servers of cloud

gaming service providers to meet the configuration demands of gamers. This paper considers the performance and rapid fitting capability of different prediction mechanisms.

On the server-side, within each 5-min interval, resource pooling is carried out based on the results of the prediction function. Assuming time slots in 5-min increments, let represent a specific time slot (where t denotes the t-th 5-min interval). Within time slot t, the relative pooling error rate ($PRER_t$) can be calculated as $PRER_t = (Pool_t - Real_t)/Real_t$, where $Pool_t$ represents the total pooled resources on the server and $Real_t$ signifies the total number of requests from all covered users $N = \{1, 2, ..., N\}$ during that time slot.

To quantify the pooling effect, this paper employs a comprehensive pooling algorithm metric, denoted as δ_t to measure the pooling efficiency. Within specific pooling scenarios, the primary trade-off lies between user experience and corporate costs. Companies aim to provide a superior user experience to enhance user satisfaction, but they must also control costs to maintain profitability. In order to determine the optimal allocation strategy that strikes a balance between providing an excellent user experience and sustaining reasonable costs, this paper utilizes a 'multi-index comprehensive scoring method' to decompose δ_t into weighted sums of user experience indicators denoted as UXI_t and cost indicators denoted as CI_t .This decomposition is specifically defined as follows:

$$\delta_t = 0.4 * UXI_t + 0.6 * CI_t \tag{6}$$

Subsequently, through prior investigations into user experience perception and corporate cost assessments under varying pooling quantities, this paper individually plots curves for the user experience indicator (UXI_t) and the cost indicator (CI_t) concerning the relative pooling error rate ($PRER_t$). These curves are fitted using an exponential modeling approach to closely capture the trends and characteristics of the original data. Consequently, expressions for UXI_t and CI_t as functions of $PRER_t$ are derived, and fitting curves are generated:

$$UXI_t = 1 - 0.5e^{-PRER_t} \tag{7}$$

$$CI_t = 1 - e^{PRER_t - 1.5} \tag{8}$$

When $PRER_t$ is equal to 0.2, meaning that the total pooling count is 1.2 times the total number of requests, the user experience indicator (UXI_t) reaches its maximum value of 1. This occurs because, under cost constraints, the total pooling count ($Pool_t$) should slightly exceed the number of requests ($Real_t$) for users to have a better experience. When pr is greater than 0, the cost indicator (CI_t) sharply decreases because unnecessary pooling waste leads to a meaningless increase in costs.

The main variables and their meanings in this study are summarized in Table 1. Representation of the Comprehensive evaluation metric (δ_t) in Terms of the Relative Pooling Error Rate ($PRER_t$), along with the corresponding figure:

$$\delta_t = 1 - 0.2e^{-PRER_t} - 0.6e^{PRER_t - 1.5} \tag{9}$$

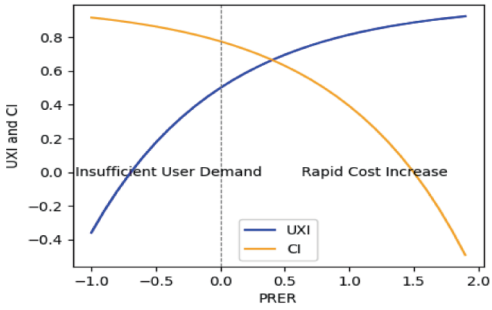

Fig. 7. Traffic characteristics among the top 10 service operators by day

Table 1. System Parameters

parameters	meaning
$PRER_t$	Relative pooling error rate in time slot
$Pool_t$	Total pooling in time slot
$Peal_t$	Actual number of user requests in time slot
$Pred_t$	Predicted number of user requests by servers in time slot
UXI_t	User experience indicator in time slot
CI_t	Cost indicator in time slot
δ_t	Comprehensive evaluation metric in time slot

From Fig. 8, it is evident that increasing the total pooling count ($Pool_t$) enhances user experience, but it also leads to a higher number of idle processes, increasing server load and operational costs. Conversely, reducing the total pooling count ($Pool_t$) decreases costs and server load but results in longer user wait times, diminishing the gaming experience. When the relative pooling error rate deviates from the optimal value, the comprehensive evaluation metric (δ_t) rapidly decreases. Therefore, the choice of the total pooling count (Pool_t) should strike a balance between user experience and enterprise costs.

In time slot, to minimize server-side pooling resource waste while ensuring an optimal user experience, the following constraints must be met: 1) Sufficient network link bandwidth for all users; 2) Available pooling resources on the server when user requests arrive; 3) Minimal server-side pooling quantity in the absence of user requests. Fulfilling constraints 1) and 2) is essential to guarantee a satisfactory user experience. Failure to meet constraint 3) may result in the server maintaining a high pooling quantity continuously, leading to resource wastage. Both the user experience indicator (UXI_t) and the cost indicator (CI_t) have been positively oriented. Consequently, the optimization goal of this paper is equivalent to finding the maximum value of the comprehensive evaluation metric (δ_t).

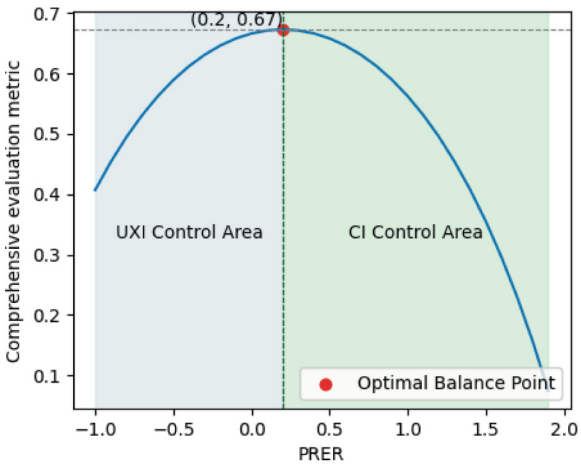

Fig. 8. Traffic characteristics among the top 10 service operators by day

5.2 PID-Based Delay Compensation Pooling Algorithm

When a cloud gaming player submits a web request, it typically corresponds to the initiation of a game. Consequently, this article correlate the access traffic of cloud games with the game pooling quantity, setting the pooling number every five minutes as a function related to the traffic prediction value. If this function represents a straightforward one-dimensional linear equation:

$$Pool_t = a \times Pred_t + b \tag{10}$$

Where both coefficients are to be optimized. Utilizing the comprehensive evaluation metric described earlier, this article can identify the coefficient values that yield the best user experience while maintaining cost-effectiveness. However, algorithms such as LSTM, ARIMA, and CATBOOST base their future predictions on historical data and inherently exhibit time-delay characteristics. This suggests their potential insensitivity in rapidly responding to sudden surges or drops in traffic. The following figure further elucidates this:

From the aforementioned figure, this article observe that at a specific time instance, the pooling value curve lags significantly behind the actual value. This observation suggests potential limitations of the linear pooling strategy in addressing sudden surges in user traffic. To reflect traffic trends in real-time and optimize our comprehensive algorithm metric, this study introduces a PID-based delay compensation strategy.

The PID algorithm is a widely-used feedback control strategy, adjusting the control output in real-time based on the error (the difference between the actual and desired outputs). In the context of this research, the PID strategy continuously monitors the discrepancy between the pooling value and the actual traffic value at the previous time instance, thereby adjusting the pooling strategy

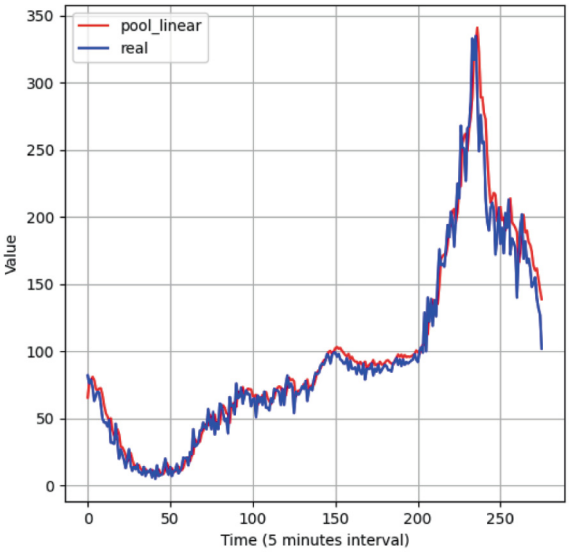

Fig. 9. Traffic characteristics among the top 10 service operators by day

for the next moment in real-time. If a significant deviation is detected between the pooling strategy and the real traffic value at the previous moment, PID will quickly recalibrate to ensure that subsequent pooling closely aligns with the true traffic trend.

The pooling function incorporating PID control is given by:

$$Pool_t = [a_t + P(Real_{t-1} - Pool_{t-1}) + D\nabla(Real_{t-1} - Pool_{t-1})] \times Pred_t + b_t \quad (11)$$

6 Performance Analysis

6.1 Analysis of Prediction Model

The fitting evaluation metrics used in this article are as follows:

1. Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \quad (12)$$

RMSE is the square root of the mean squared error between the predicted values and the actual values. A smaller RMSE indicates a better fit of the data.

2. Mean Absolute Error (MAE):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\widehat{y}_i - y_i| \tag{13}$$

MAE is the mean of the absolute differences between the predicted values and the actual values. A smaller MAE indicates a better fit of the data.

3. Coefficient of Determination: R-squared (R2):

$$R2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(\bar{y}_i - y_i)^2} \tag{14}$$

The numerator is the sum of squared differences between the actual values and the predicted values, and the denominator represents the sum of squared differences between the actual values and the mean value. R-squared values range from $[0, 1]$, with values closer to 1 indicating a higher degree of fit.

The article describes the process of training models using a divided training dataset, introducing a validation set to monitor loss convergence, and utilizing the trained models to make predictions on a test dataset. The results for two different algorithms, ARIMA and LSTM, are presented for predicting traffic data for three games: 'Genshin Impact',using access_key_id 51a0e31e0d7.

The results are visualized in Figs. 10 and 11, with explanations for each figure:

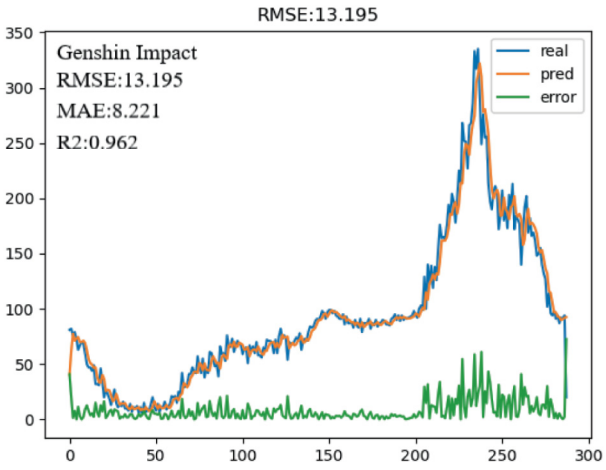

Fig. 10. ARIMA Predictive Results

In Fig. 10, the 'real' line represents the actual traffic data for the entire day on October 31st, while the 'pred' line represents the predictions made by the

ARIMA model after training on the traffic time series. The 'error' line shows the absolute differences between the actual and predicted values, providing an intuitive indication of prediction performance. Based on three evaluation metrics, it can be observed that the ARIMA algorithm can achieve reasonably accurate predictions for the high-traffic 'Genshin Impact' game.

Using the LSTM algorithm to predict traffic for the 'Genshin Impact' game with access_key_id 51a0e31e0d7 yields the results shown in Fig. 11.

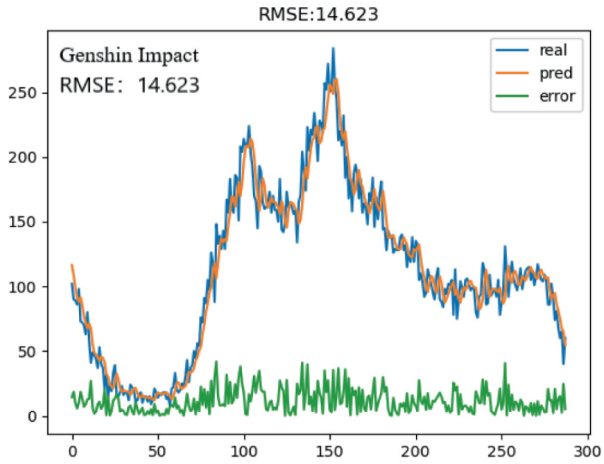

Fig. 11. LSTM Predictive Results

The 'pred' line represents the predictions made by the LSTM model after training on the traffic time series. The 'real' and 'error' lines serve the same purpose as in Fig. 11.

The text highlights that LSTM outperforms ARIMA in terms of accuracy, as indicated by a lower error (RMSE) for all three games: 'Genshin Impact' .

In summary, both ARIMA and LSTM algorithms are used for traffic data prediction, with LSTM demonstrating superior predictive accuracy compared to ARIMA, as indicated by lower error values. The results are presented graphically to provide a clear visual understanding of the models' performance.

To compare the predictive performance of the three algorithms, the article mentions using 'Genshin Impact' as an example and plotting the predictive results of the three algorithm models against the actual values in a single graph (Fig. 12). The evaluation metrics for each algorithm, namely RMSE, MAE, and R2, are also provided in Table 1 for this specific access_key_id and game.

From Fig. 12, it can be observed that the predictive results of the various algorithms closely match the actual values, indicating their capability to perform accurate predictions of game traffic. To assess the relative performance of these algorithms, evaluation metrics are necessary. Using 'Genshin Impact' as an

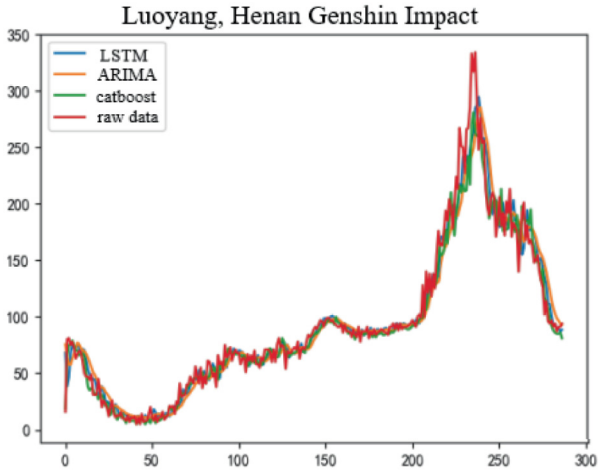

Fig. 12. Predictive Results for "Genshin Impact" Using Three Algorithms

illustrative example for this access_key_id, the RMSE, MAE, and R2 metrics for each algorithm are provided in Table 2.

Table 2. Three Algorithm Evaluation Metrics Comparison

	ARIMA	LSTM	Catboost
RMSE	13.195	12.474	13.148
MAE	8.221	8.137	8.049
R2	0.962	0.965	0.962

From the results, it is evident that the deep learning-based LSTM algorithm outperforms the traditional time series forecasting method ARIMA in terms of accuracy, as indicated by the significantly lower RMSE. The Catboost algorithm, based on machine learning, shows predictive performance similar to that of ARIMA. This suggests that LSTM, with its unique mechanisms like cell states and forget gates, has a significant advantage in cloud gaming traffic data prediction.

In summary, while all three algorithms provide reasonably good predictions for 'Genshin Impact' LSTM demonstrates superior accuracy compared to ARIMA and Catboost, with lower RMSE values.

6.2 Analysis of Resource Pooling Algorithms Model

Using the LSTM prediction strategy as an example and given different coefficients, the performance of the comprehensive evaluation metric $\delta(t)$ is illustrated by the contour lines in Fig. 13.

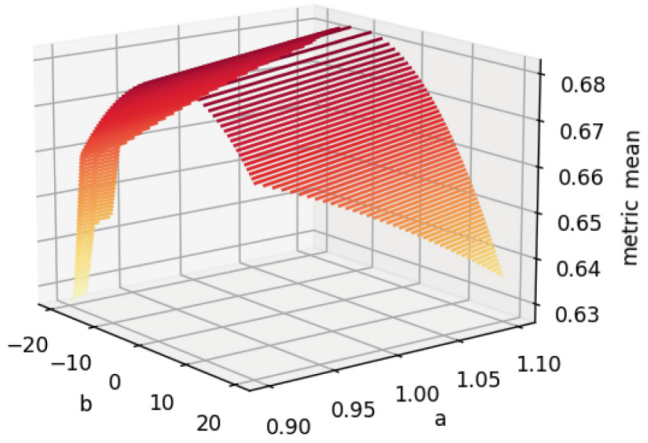

Fig. 13. Evaluation Metrics of Parameters a and b in the First Half-hour

In Fig. 13, the x-axis represents the range of possible values for parameter a, spanning from 0.9 to 1.1; the y-axis signifies the range for parameter b, ranging from −20 to 20; the z-axis showcases the average evaluation metric derived from various combinations of parameters a and b. A full day consists of 288 five-minute intervals. For a more precise analysis, these intervals are broken down into 48 half-hour segments. For each 5-min interval, user experience and cost metrics are computed using the provided formula, leading to a comprehensive pooling algorithm evaluation for that interval. By averaging the evaluation values across 30 consecutive 5-min segments, this paper obtain the average comprehensive metric for that particular half-hour. Based on this data, this article identified the a and b parameter values that maximize the average comprehensive evaluation metric.

Through such an analysis, this article determined an optimal set of a and b values for each half-hour segment. Subsequently, this article employed the PID algorithm to approximate these optimal a, b values, allowing for automatic adjustments of parameters a and b every half-hour in practical applications, aiming to further enhance the overall comprehensive evaluation metric.

After comprehensively considering user experience and cost metrics, this paper adjusted the predicted traffic value to arrive at what we term the 'optimal pooling value', represented by the red line in the figure. However, in real-time prediction, there is no prior knowledge of the actual traffic value. We can only adjust based on the previous moment's pooling situation and optimize parameters a and b using the PID algorithm. The goal is to make the actual pooling value align as closely as possible with the optimal pooling value. This adjusted pooling value is represented by the green line in the figure. Observations from the chart indicate that the revised PID approach allows the green line to fit the red line closely. Compared to the actual values, this significantly rectifies the

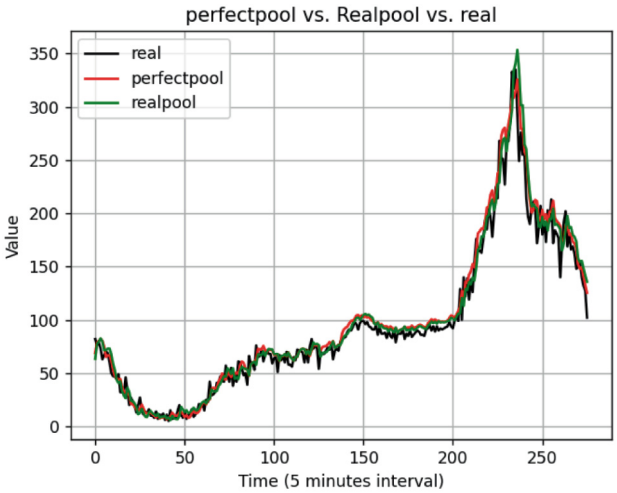

Fig. 14. Optimal Pooling Value, Actual Pooling Value, and Real Traffic Value

deviation caused by the delay between the pooling value and the actual traffic value.

This study primarily discusses the delay compensation method based on PID. Another effective solution is the 'accumulation' delay compensation method, which addresses the issue of pooling latency. The central idea behind this method is to set a higher pooling value at specific moments and then accumulate the excess pooling value to the subsequent intervals. This continuous accumulation ensures that when there's a sudden surge in traffic, the original pooling value augmented with the accumulated portion can meet the users' traffic needs, mitigating issues caused by the inherent delay.

Next, this paper will compare the PID-based delay compensation method with the 'accumulation' delay compensation approach, presenting a comparison of comprehensive evaluation metrics for each interval and an overall accumulated comprehensive evaluation comparison.

As illustrated in Fig. 15, the comprehensive evaluation metrics derived using the accumulation algorithm exhibited significant fluctuations, even reaching negative values at certain intervals. In contrast, the metrics calculated using the PID algorithm demonstrate much higher stability.

From Fig. 16, it is evident that the implementation of either the accumulation or PID algorithm has significantly enhanced the cumulative comprehensive evaluation metrics throughout the day, thereby achieving a better balance between user experience and enterprise costs. Notably, the PID algorithm outperforms the accumulation algorithm in terms of performance. This observation further confirms that, in the context of resource pooling allocation, the PID algorithm can more consistently and stably strike a balance between user experience and enterprise costs.

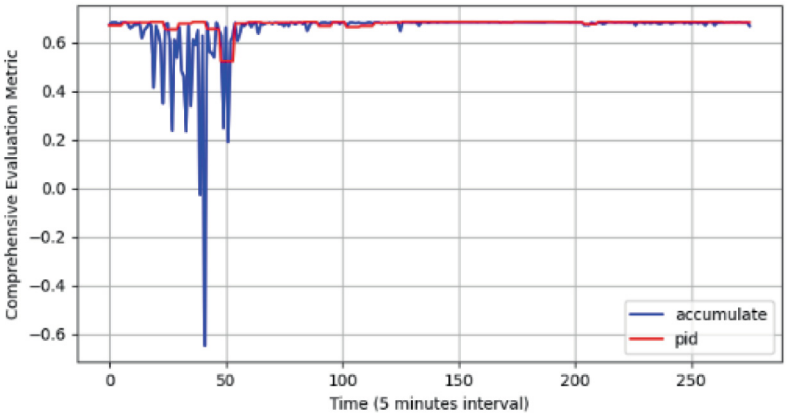

Fig. 15. Comparative Figure of Comprehensive Evaluation Metrics

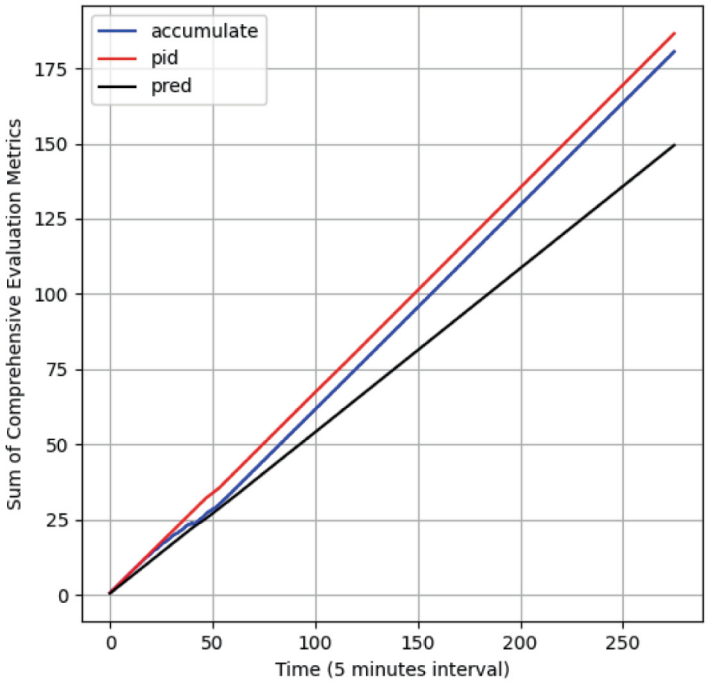

Fig. 16. Cumulative Comprehensive Evaluation Comparison

7 Conclusion

In the present era, marked by the rapid advancement of information technology, a new content distribution paradigm based on cloud gaming is diverging from the traditional modes of download-installation and interest recommendation,

steadily expanding its market reach. While the current phase of cloud gaming predominantly focuses on cloud adaptations of already popular games, the continuous growth in cloud gaming users indicates a shift. It's anticipated that native cloud games will soon be embraced within the creative scope of major game developers. Thanks to the unique advantages of cloud gaming, it holds the potential to further attract not only avid gamers but also those previously uninterested, establishing a positive feedback loop for its continued evolution. This underscores the pressing need for more efficient traffic prediction algorithms and resource pooling management techniques.

Simultaneously, with a constant rise in the number of internet data centers nationwide, IDC's traffic throughput witnesses consistent year-on-year growth. Accurate and timely predictions of internet data center traffic can foster real-time optimization of network resource allocations. This not only avoids potential network congestion but also significantly elevates the service stability of these data centers, ultimately leading to reduced operational and maintenance costs. Thus, research focused on traffic prediction for internet data centers harbors a vast horizon of application potentials.

References

1. Guan, P.: Analysis of the convergence development of cloud computing and animation/game industries in Fujian Province. Res. Fine Arts Educ. **17**, 110–111 (2013)
2. Liu, K., Lin, G.: Discussion on cloud game technology in 5G era. Wirel. Connect. Technol. **19**(06), 104–105 (2022)
3. Tang, F., Liu, X.: Research on a novel cloud gaming resource allocation model. Guangdong Commun. Technol. **41**(12), 2–5 (2021)
4. Han, Z.: Cloud gaming: a new industry based on cloud computing platform. China Comput. Commun. **17**, 47–51 (2017)
5. Tang, J., Xu, F., Pu, Q.: Research on the core network architecture for Guangxi Unicom's Internet of Things in the 5G era. Guangxi Commun. Technol. **131**(02), 23–27 (2018)
6. Hou, J., Zhang, Y., Xu, H., Zhu, X., Xing, K.: Research on mobile edge computing unloading based on deep reinforcement learning. J. Jinling Inst. Technol.
7. Shen, H., Wang, L.: Task offloading based on mobile edge computing and its privacy-preserving issues: a survey. **69**(02), 258–269 (2023)
8. Ismail, B., Goortani, E., Karim, M.: Evaluation of docker as edge computing platform. In: 2015 IEEE Conference on Open Systems (ICOS). IEEE (2015)
9. Wei, B., Wei, D.: Resource allocation for cloud gaming based on game-session-length prediction. Comput. Eng. Des.
10. Tang, Y., Liu, X.: Research on a Novel Cloud Gaming Resource Allocation Model
11. Liu, H.: Research of Adaptive Resource Allocation in Cloud Gaming
12. Xu, L., Zhao, W.: Prediction of Free Cash Flow of Enterprises Based on ARIMA Model
13. Cao, Q., Chen, X., Liu, H.: Network Traffic Forecasting Method Based on SARIMA-LSTM Hybrid Model
14. Shi, L., Zhang, J., Gao, F.: Intrusion Detection in Network Traffic Using Transformer and BiLSTM-Based Technology
15. Shen, Y., Li, L.:The Realization of Traffic Prediction by Improving Particle Swarm Optimization and Optimizing BP Network