Echo State Network-Based Content Prediction for Mobile Edge Caching Networks

Zengyu Cai, Zhengzhou University of Light Industry, China

Xi Chen, Zhengzhou University of Light Industry, China

Jianwei Zhang, Zhengzhou University of Light Industry, China*

https://orcid.org/0000-0002-3178-0607

Liang Zhu, Zhengzhou University of Light Industry, China

Xinhua Hu, Zhengzhou University of Light Industry, China

ABSTRACT

With the rapid development of internet communication and the wide application of intelligent terminal, moving the cache to the edge of the network is an effective solution to shorten the delay of users accessing content. However, the existing cache work lacks the comprehensive consideration of users and content, resulting in low cache hit ratio and low accuracy of the whole system. In this paper, the authors propose a collaborative caching model that considers both user request content and content prediction, so as to improve the caching performance of the whole network. Firstly, the model uses the clustering algorithm based on Akike information criterion to cluster users. Then, combined with the clustering results, echo state network is used as the machine learning framework to predict the content. Finally, the cache contents are selected according to the prediction results and cached in the cache unit of the small base station. Simulation results show that compared with the existing cache algorithms, the proposed method has obvious improvement in cache hit ratio, accuracy, and recall rate.

KEYWORDS

Clustering, Content Prediction, Echo State Network, Mobile Edge Caching Networks

INTRODUCTION

As vast amounts of information travel through the network, most internet traffic is related to content distribution. Therefore, meeting the low-latency transmission and high throughput requirements of different types of traffic is an inevitable requirement for improving user experience and network computing performance (Du et al., 2021). In order to cope with the challenge of rapid growth of network traffic and alleviate the traffic pressure of the core network, caching has been studied as an effective tool to reduce latency by prestoring the most popular content in cache space. At the same time, with the wide application of artificial intelligence technology in people's lives, the research on

DOI: 10.4018/IJITWE.317219

*Corresponding Author

related technologies in the field of artificial intelligence has gradually received extensive attention. Echo state network is an important method in the field of artificial intelligence. As a nonlinear adaptive dynamic system, its fast machine learning speed has been successfully applied to the prediction of network traffic (Zhang et al., 2021).

In order to improve the user experience and reduce the backhaul data traffic, a lot of work related to caching technology has been done in recent years (Li et al., 2020; Hu et al., 2021; Thar et al., 2016; Chhangte et al., 2021). Li et al. (2020) propose a probabilistic cache placement method based on content centrality. This method improves the cache hit rate and cache content utilization by considering the content centrality and content acquisition delay to adaptively calculate the probability of node cache. However, its computational complexity is high, and it is only suitable for small and medium-sized networks. Hu et al. (2021) propose an edge network caching strategy based on social relationship awareness. This strategy maps the users' social relationship strength according to the similarity of user needs, and then selects the user as the auxiliary cache location according to relationship strength. This strategy has a certain improvement in cache hit rate and system cache delay, but there are deficiencies in edge cache updates. Thar et al. (2016) propose a core router cache decision algorithm, which improved cache hit rate and reduced content acquisition delay and hit distance, but did not consider user interest and popularity of hot content. Chhangte et al. (2021) propose a service that implements distributed caching at the edge of Wi-Fi. This service combines distributed caching of software-defined networks to effectively improve the user experience in the target network, but has a certain disadvantage in content transmission delay.

Although the existing caching strategy improves the cache hit rate, there are still many problems in practice (Serhane et al., 2021; Ren et al., 2020; Krishnendu et al., 2022). Serhane et al. (2021) propose a cache-optimization algorithm based on chemical reaction. Although the algorithm reduces energy consumption, it does not consider the importance of content. Ren et al. (2020) proposed a unmanned aerial vehicle deployment and caching strategy based on user preference prediction. This strategy is combined with UAV base station scheduling to optimize the average cache hit rate of the system, but does not consider the user distribution problem. Krishnendu et al. (2022) propose a strategy for wireless edge caching and content popularity prediction using machine learning. However, the popularity of content cached in the network lags behind the change of user preferences, resulting in a decrease in cache hit rate. In addition, the statistical results of content popularity in real networks are often discretely distributed. In order to solve this problem, Chen et al. (2017) propose an algorithm that combines the machine-learning framework of echo state networks with sub-linear algorithms to study the active caching problem of cloud wireless access networks. Compared with the traditional content popularity prediction algorithm, the echo state network can understand the distribution of user content requests without a lot of training, but this paper does not consider the content correlation.

Although content-centric networking is a promising new network architecture that can alleviate server bottlenecks, balance network traffic, and reduce user access latency, there are still many unresolved issues that need to be addressed (D. Zhu et al., 2020; Zeng et al., 2022; Ferdousi et al., 2015; M. Zhang et al., 2017; Jmal & Chaari Fourati, 2017; Shrisha & Boregowda, 2022). Naeem et al. (2018) believe that if only the content is considered, the problem of high cache redundancy and low accessibility of cached content will occur. Therefore, the probabilistic cache scheme based on content-centric networks is studied, but the scheme has low availability in content space. For example, in the same time period, multiple users are connected to the same cache node at the same time, and request different types of cache content. Because the content of the content-centric network is the same type of content, only one type of content will be hit at this time, which will lead to a very low cache hit rate and low resource utilization. When a cache node caches multiple different types of content, fewer users connect to the cache node, resulting in excessive content redundancy in the node.

In this paper, in order to solve the existing cache work's lack of comprehensive consideration of users and content, which results in a low cache hit rate and low accuracy of the whole system, a prediction-based caching strategy is proposed, which considers both user preference and content popularity. This paper first classifies the users' location and content using a clustering algorithm, and caches different types of content on the cache unit of a small base station. Due to the limited storage space of the cache unit, it is necessary to sort the probability of users requesting different files of the same content type, and to select several content types with the highest probability for caching. *Most popular content* in this article refers to the popularity of each type of content that we use the echo state network to predict and rank. We define the top ten types of content as the most popular content in this region. Then, the echo state network is used to predict the users' content request distribution and calculate the percentage of each type of content to ultimately determine which predicted content to cache. The main contributions of this paper are summarized as follows:

A clustering algorithm based on the Akaike information criterion is proposed. According to different user characteristics, users are divided into different user groups. This method can effectively improve the cache hit rate.

The echo state network is innovative when applied to the field of content prediction. According to the prediction results, the users' most popular content and the most popular content predicted will be cached in the small base station at the same time. This method can effectively reduce the content acquisition delay.

RELATED WORK

The research in this paper is at the intersection of three research themes: caching problems, content prediction, and echo state networks.

Caching Problems

Based on the degree of visibility of cached content to users, the existing base station caches are divided into transparent caches and explicit caches. In transparent caching, the cached content is transparent to both the origin server and the user (K. Li et al., 2007). The cache server is deployed at a base station on the link between the source server and the user, and it intercepts all user requests passing through this link. If the content requested by the user is already cached at the base station, the cache will hit and distribute the content directly from the caching server of the base station. If the content requested by the user is not cached on the base station, the base station forwards the user request to a higher-level server or source server until the higher-level server or source server can satisfy the user request. Alternatively, when the original base station cannot respond to the user request, the original base station can forward the user request to another base station, and the other base station can respond or forward the user request instead of the original base station (Poularakis et al., 2014). Most of the existing cellular network's edge caching is based on the research of transparent caching. Although the research on explicit base station caching is still in its infancy, Bastug et al. (2014) first proposed the concept of explicit base station caching. In the explicit base station caching system, the content requested by the user is limited by the locally cached content, and the content requested by the user will be directly distributed from the base station. Through mobile edge computing technology, the content that users are interested in and that can be reliably distributed can be optimized and filtered in real time and cached on small base stations to provide users with high-quality content distribution services. The explicit base station caching system is well adapted to the increasingly dense small base station network environment and is a promising and reliable solution to the network traffic explosion problem.

Content Prediction Methods

According to the technical means employed for prediction, the currently available methods can be broadly classified into two types of prediction algorithms: traditional prediction algorithms and deep learning-based prediction algorithms. The comparison of existing technology deficiencies is shown in Table 1. With the ARMA Model it is difficult to provide optimal predictions for content. With the

Table 1. Existing Technology Deficiencies Comparison

Methods	Existing Problems
ARMA Model	Difficult to provide optimal predictions for content
Principal Component Analysis	Constructing a heat prediction model based on random forest algorithm
Logistic Machine-Learning Algorithm	Difficult to adapt to content prediction with long historical data

logistic machine-learning algorithm it is difficult to adapt to content prediction with long historical data. But our model takes into account both user-requested content and content prediction. Traditional prediction methods convert the heat prediction problem into a regression or classification problem by extracting manually defined feature information and introducing machine-learning techniques to solve it. Hassine et al. (2017) studied the prediction capability of various ARMA models and revealed that ARMA models make it difficult to provide optimal predictions for content. C. Zhu et al. (2016) used principal component analysis to implement dimensionality reduction of the selected features, and then, based on the random forest algorithm to build a heat prediction model, they achieved fast prediction of video heat. Xu and Xiao (2019) used a logistic machine-learning algorithm based on user behavior information, and designed content prediction for scenarios with sparse consumption data, which is difficult to adapt to content prediction under long historical data. The above traditional prediction methods rely on heuristic feature ideas or on specific statistical methods with explicit mechanisms, and although they have good accuracy in samples, the prediction accuracy is always unsatisfactory in open practical-engineering scenarios due to the poor adaptiveness of the methods.

Echo State Network

In 2004, Jaeger and Haas proposed an artificial recurrent neural network named ESN (Echo State Network) (Jaeger & Haas, 2004). As a new type of neural network, it is based on the basic principle of neural networks in biology. It is similar to the response mechanism performed by the human brain structure when it is stimulated by external stimuli. The echo state network model includes the process of learning and prediction. It consists of input neurons, a reserve pool, and output neurons. The neurons in the reserve pool are connected to each other and are used to retain the information left at the previous moment. The connection weights of the echo state network from the input layer to the alternate pool are generated by random initialization. The reserve pool can maintain dynamic activity and can be continuously activated even without input. Currently, the reservoir pool of ESN has been extensively studied (Song & Feng, 2010; Cui et al., 2012; Qiao et al., 2016). Ortin et al. (2015) studied the relationship between the connection structure of the reservoir and the prediction performance. During training, only the connection weights from the spare pool to the output layer need to be trained, which becomes a linear regression problem. As a result, the training of ESN is very fast. It can overcome the shortcomings of traditional neural networks such as low training efficiency and slow algorithm convergence speed, and is often used to solve practical problems. However, there are also many problems that have not been addressed; for example, the full connectivity of the output synapses leads to the degradation of the network prediction performance.

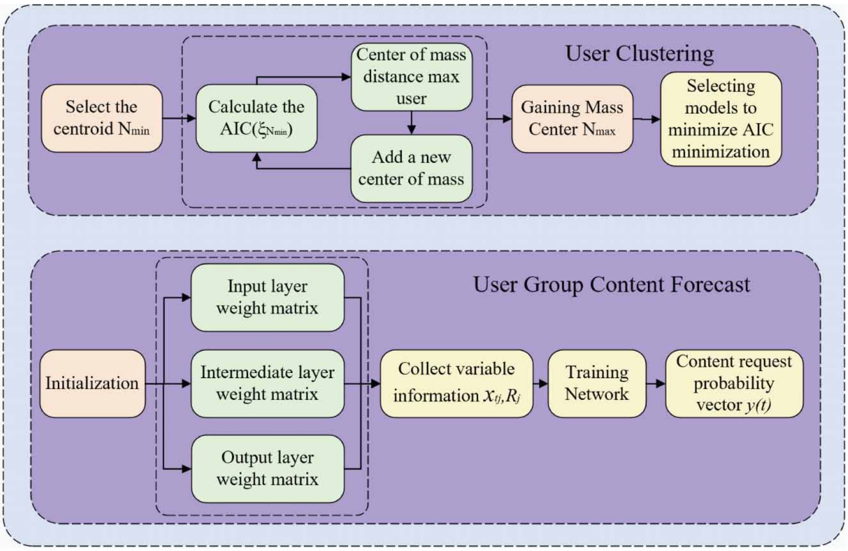
The method proposed in this paper unites data from multiple parties, bridges data silos, and solves the problems of poor prediction accuracy of echo state networks and low hit rate of edge caching in cellular networks while performing content prediction.

SYSTEM MODEL

This paper considers a small cellular network that uses clustering and supports caching. The network consists of a content source server, a macro base station, a small base station, and several users. Each small base station contains two content caching units for limited caching, which represent CCU (Content-centric Cache Unit) and CPU (User-centric Prediction Unit), respectively. CCU is a content-centric cache that saves the most popular content types at the moment, while CPU is a user-centric cache that saves the most popular content predicted in the next slot. This takes into account both the user and the content that needs to be cached. The cache size of CCU is very large, and therefore can cache multiple content types, while CPU can only cache several popular content types. Assume that each user is connected to only one small base station and can request up to one type of content per time period. Based on the diversity of users, users are first clustered into different user groups, and then the content with the highest user request probability is cached in the CCU. Secondly, since multiple users may request different content at the same time, caching only the same type of content cannot achieve a better caching effect. To solve this problem, this paper considers using the echo state network to predict the content request of each small base station by the user group, so that the content most likely to be requested by the user at the next moment is cached in the small base station's content prediction cache unit CPU in advance. This is mainly because in the same region, most users request roughly the same type of content, and only a few users may request different types of content in different slots, so only a small portion of the cache space needs to be replaced in the next slot. In the cache replacement process, the content update cycle in the CCU cache is long, and the content update cycle in the CPU cache is short, thus can better adapt to the dynamic changes of user-requested content. In addition, due to the lower content in the CPU cache, the update is easier and more efficient. The system flow chart is shown in Figure 1. The system flow chart uses a clustering algorithm based on red pool information criteria to effectively cluster users through machine-learning methods, so as to predict the type of traffic that users are likely to generate before they generate traffic requests. Compared with the latest research work, the model innovatively introduces an echo state network to effectively predict what users are interested in. It allows for personalized cache content preloading for individual users.

In the model used in this paper, the user information is collected every time. At the same time, the content is cached in the macro base station and the small base station. Considering the limitations of the backhaul capacity of the core network, this paper's goal is to maximize the traffic service directly from the small base station to the user by reducing the load of the core network, and to directly cache

Figure 1. System Flow Chart


User Clustering
Select the centroid N_{min}
Calculate the AIC($\xi_{N_{min}}$)
Center of mass distance max user
Add a new center of mass
Gaining Mass Center N_{max}
Selecting models to minimize AIC minimization
User Group Content Forecast
Initialization
Input layer weight matrix
Intermediate layer weight matrix
Output layer weight matrix
Collect variable information x_{tj}, R_j
Training Network
Content request probability vector $y(t)$

the popular content requested by all users in the small base station, which can effectively reduce the backhaul traffic. At the same time, caching popular content in small base stations can also reduce the delay of forward and backhaul content with the shunt traffic, and improve the quality of service.

As shown in Figure 2, the network modeling in this paper selects a tree-like hierarchical structure, and there is a subordinate relationship between the base stations. After the user requests the content, the interest packet first passes through the small base station. If the local small base station cannot respond to the request, the small base station forwards the interest packet to the previous macro base station; if it is still unable to hit, it then continues forwarding until it reaches the source content server. After the cache hit, the data packet will be transmitted along the request path in the reverse link, and the cache decision and replacement strategy will be executed on the base station along the path to achieve content distribution. In our simulation experiment, the number of cache servers in the path from the user to the original base station is two. The tree structure is a hierarchical topology. In the actual network, with this kind of topology it is easy to add nodes and branches, the fault isolation is also easy, and the expansibility is good.

In this paper, we use a novel prediction method, namely the echo state network. Considering the regularity of user content requests, the echo state network can establish the relationship between user information and request content, so as to achieve the purpose of prediction. Compared with the traditional prediction algorithm, the prediction method of the echo state network can obtain more accurate results. The echo state network is a special recurrent neural network that adds a dynamic repository. Due to the time-varying characteristics of dynamic systems, echo state networks are more suitable for dealing with dynamic system modeling problems, such as prediction. Generally speaking, echo state network system models consist of three layers: the input layer, the middle layer, and the output layer. The echo state network model diagram is shown in Figure 3.

ALGORITHM DESIGN

In this section, different users are divided according to their characteristics, users are clustered using clustering algorithms, and echo state networks are used to predict the most popular content for user groups.

User Clustering Algorithm Based on Akaike Information Criterion

Generally, users have different content popularity. However, users from the same social group may show some correlation in user content requests. The purpose of this article is to group users: since users in the same cluster are likely to request the same content, users can be effectively clustered according to content interests, so that their request differences can be minimized. This paper will cluster according to the model of H. Akaike (1974), where AIC (Akaike Information Criterion) is a

Figure 2. System Model Diagram

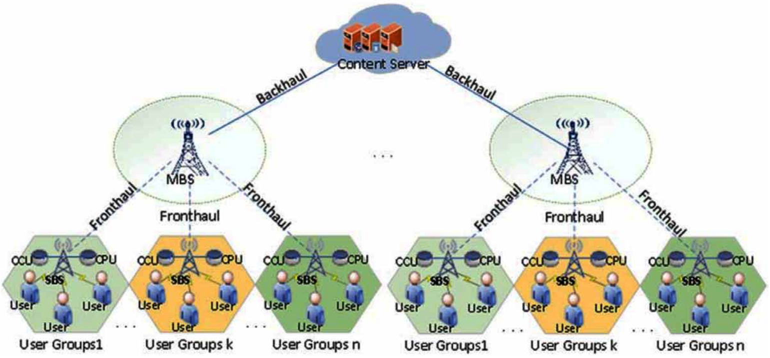
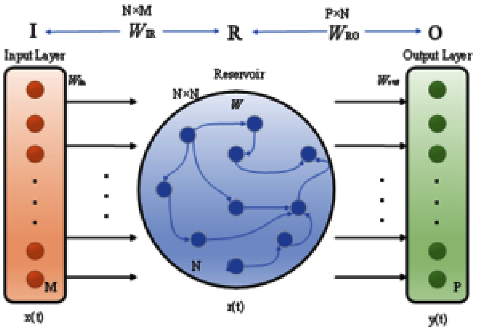


Figure 3. Echo State Network Model



statistical model selection criterion. Given a model set of data, AIC can estimate the mass of each model relative to each other. This paper considers a set of statistical models that represent the process of generating popularity vectors of file content for each user:

$$\Xi = \{\xi N_{min} ... \xi N_{max}\}$$

where the $\{N_{min} ... N_{max}\}$ indicates the range from which to select the number of clusters, and the ξ_i represents the statistical model of the observed popularity vector when users are grouped into the *I* cluster. Each model is characterized by a finite set of parameters which represent the variance and average popularity vector in each cluster. The AIC of each model is given by Equation (1):

$$AIC(\xi_i) = 2k_i - 2\ln(L_{\xi_i}) \tag{1}$$

In Equation (1), the k_i is the number of estimable parameters in model ξ_i, and L_{ξ_i} is based on the data likelihood of model ξ_i.

This paper aims is to minimize the AIC standard, as per Equation (2):

$$\xi AIC = \arg\min_{\xi} AIC(\xi_i) \tag{2}$$

The algorithm in this paper first clusters users while assuming the existence of N_{min} user groups, and then adds centroids based on certain criteria until the upper limit is reached N_{max}. If the AIC is decreasing over the entire interval $[N_{min}, N_{max}]$, the search should be extended until the minimum value is reached. For effective clustering, this paper needs to export a criterion that specifies where new centroids should be added. Therefore, this paper proposes a simple criterion in which at each step of the algorithm, a new centroid is added to the cluster with the largest average distance between its centroid popularity vector and the centroid popularity vector of the user. The new center will be chosen as the node with the greatest distance from its cluster centroid. Far away means that the user has very different behavior; that is, they are less likely to request the same file as other users in the cluster. Once a new centroid is selected, the algorithm influences users toward clusters where the correlation between their popularity vector and centroid is maximized. This will minimize differences in user behavior within the same cluster. In each iteration, the popularity state of cache cluster *k* is calculated as the centroid of the popularity state of all users in that cluster, as per Equation (3):

$$\overset{\wedge}{P_k} = \frac{\sum_{k=1}^{N_k} P_u}{N_k} \tag{3}$$

In Equation (3), the popularity status of $\overset{\wedge}{P_k}$ cache cluster *k* is expressed, the number of users in cluster *k* is expressed as N_k , and the popularity P_u vector of each user is expressed.

Therefore, the clustering algorithm proposed in this paper is shown in Table 2.

User Group Content Prediction Algorithm Based on Echo State Network

Assume that the number of nodes in the input layer, middle layer, and output layer of the echo state network are *M*, *N*, and *P*, respectively. The input vector $x(t) = \left[x_1(t), x_2(t), ..., x_M(t)\right]^T$ represents the users' context at time *t*, and *M* is the number of attributes that make up the users' context information, such as: content request time, gender, occupation, age, and device type.

The middle layer is essentially a recurrent neural network consisting of *N* nodes, called a reserve pool, and the intermediate vector is represented by $r(t) = \left[r_1(t), r_2(t), ..., r_N(t)\right]^T$. The state of the middle layer of the echo state network at time *t* is shown in Equation (4):

$$r(t) = \alpha \tanh(Ar(t) + W_{in} x(t) + \xi) \tag{4}$$

In Equation (4), the $\alpha \in (0,1]$ leakage rate is expressed. It is mainly used to control the speed of the weight update of each node in the reserve pool. *Tanh* represents the activation function of internal neurons, and A represents the weight adjacency matrix of the reserve pool, usually a sparse

Table 2. Algorithm 1: User Clustering Algorithm Based on Akaike Information Criterion

Input: User sample set $X = \{x_1, x_2, ..., x_m\}$.

Output: Clustering model with minimum AIC value.

1. **Initialization**: The cluster number interval $[N_{min}, N_{max}]$ randomly selects the first centroid N_{min} from the user.

2. **Calculate**: According to Equation (1) calculated $AIC(\xi N_{min})$.

3. **Select**: The user with the largest distance from their cluster centroid in the cluster with the largest variance to add a new centroid with the popularity of the selected user.

4. **Repeat**: Run steps 2 and 3 until you arrive at N_{max} .

5. **Select**: The model that minimizes AIC and cluster users accordingly.

End

matrix. W_{in}, which is represented by a matrix of N rows and M columns, is mainly used to convert the input signal of the M dimension into a form acceptable to the reserve pool. $x(t)$ represents the input vector, and the dimension is M, ξ which represents the bias.

The state update of the middle layer of the echo state network at time *t*+1 is shown in Equation (5):

$$r(t+1) = \alpha \tanh(Ar(t) + W_{in}x(t+1) + W_{back}y(t) + \xi) \tag{5}$$

In Equation (5), W_{back} represents the weight matrix from the output layer at the previous time to the middle layer at the next time, $x(t+1)$ and $r(t+1)$ denote the input state and intermediate state, respectively.

The output vector $y(t) = \left[y_1(t), y_2(t), ..., y_P(t)\right]^T$ represents the probability distribution of the users' content request, $y_P(t)$ is the probability that the user requested the content *P* at time *t*. The output layer state of the echo state network at *t*+1 is shown in Equation (6):

$$y(t+1) = f_{out}(W_{out}[x(t+1); r(t+1)]) \tag{6}$$

In Equation (6), f_{out} is the activation function of the output layer artificial neuron, [;] indicates that two vectors are concatenated. The goal of this paper's training is to train W_{out} to minimize the difference between $y(target)$ and $y(t+1)$. The calculation of W_{out} can be implemented as shown in Equation (7):

$$W_{out} = YR^T(RR^T + \lambda I)^{-1} \tag{7}$$

In Equation (7), $R = \{r_1(i), r_2(i), ..., r_P(i)\}$ $(i = p, p+1, ..., P)$ represents the state matrix of the middle tier, $Y = \{y(p), y(p+1), ..., y(P)\}$ represents the output values at different moments. R^T is the transpose matrix of R, λ is the normalization coefficient, and I is unit matrix. R^{-1} represents the inverse matrix of R. Considering the artificial neuron of the leakage integrator of the reservoir, as the number of iterations increases, the echo characteristics of the echo state network may decrease or even disappear, so this paper uses an improved echo state network algorithm to calculate the middle layer state R. The proposed user group content prediction algorithm based on the echo state network is shown in Table 3.

Based on the output of user population content request probability, this paper needs to update it in the content prediction cache unit with a cache replacement strategy. During the cache replacement process, this paper replaced the cached content with the less-recently-used algorithm to ensure that less-popular content should be replaced with new incoming content to better accommodate dynamic changes in user-requested content. Finally, the time complexity of the user clustering algorithm is *O(n)*, and the time complexity of the user group content prediction algorithm is *O(1)*.

EXPERIMENTAL RESULTS AND ANALYSIS

Datasets

This experiment uses MovieLens, an open-source dataset provided by the GroupLens project team of the University of Minnesota. The entire dataset includes 6040 users' information and 1,000,209

Table 3. Algorithm 2: User Group Content Prediction Algorithm

Input: User context information $x(t)$

Output: The vector of user group content request probability $y(t)$

1. **Initialization**: The main parameters include input layer artificial neuron *M*, output layer artificial neuron *P*, spectral radius *SR*, sparsity *SD*, middle layer size *N,* and input expansion *IS*.

2. **Building ESN model**: Construct an echo state network model based on initialization parameters. First, generate a weight matrix including input weight matrix $W_{in} \in X^{N\times M}$, middle layer weight matrix $W \in X^{N\times N}$, and output layer weight matrix $W_{out} \in X^{P\times(N+M)}$. W_{in} and W are initially randomly generated by uniform distribution. Once initialized, these two matrices will never change in subsequent training. W_{out} is also randomly initialized by a uniform distribution, but is constantly updated in subsequent training.

3. **Collect variable information**: Define $x_{tj} = \{x_{t1}, x_{t2}, ..., x_{tM}\}$ to represent the context message of user *j* at time *t*, intermediate state $r_{tj} = \{r_1(t), r_2(t), ..., r_N(t)\}$ and output variable $y_{tj} = \{y_{j1}(t), y_{j2}(t), ..., y_{jP}(t)\}$. This paper also needs to collect $R_j = \{r_{j1}(i), r_{j2}(i), ..., r_{jP}(i)\}$ and $Y_j = \{y_{j1}(i), y_{j2}(i), ..., y_{jP}(i)\}$, $\left(i = p, p+1, ..., P\right)$ form a matrix for subsequent training W_{out}.

4. **Training network**: Training W_{out} based on connection variables R_j and Y_j.

5. **Predict**: After the training is completed, the echo state network can be used to predict the popularity of content.

End

ratings for 3706 movies, with a rating range of 1–5. Because this paper aims at the problem of content popularity prediction based on echo state networks, only the user movie rating set and user information set are extracted as experimental data.

Performance Evaluation Criteria

Prediction accuracy and cache hit rate are important metrics to evaluate the performance of this paper. The cache hit rate is given by the probability that a given user will find the requested file in the cache of a small base station within radius L. In the simulation experiment, the range of content transmission depends on the transmission radius *r* value we set, which is 200 meters by default. $U = \{u_1, u_2, ..., u_i\}$ indicates a collection of users, $S = \{s_1, s_2, ..., s_j\}$ indicates a collection of small base station nodes, $M = \{m_1, m_2, ..., m_k\}$ indicates the set of macro base station nodes, $R = \{r_1, r_2, ..., r_l\}$ indicates a collection of content source server nodes. In the simulation, the default number of users requesting content was 2000, with 200 users per group. The base station or cache server cannot display the requested content and is ignored. In case of site failure, the number of requests forwarded to the neighbor cache server is zero. To formulate the cache-optimization problem, define that the size of all content blocks in the content source server is L and the cache capacity of each small base station as $C_c = \{C_{c1}, C_{c2}, .., C_{cn}\}$ and $C_p = \{C_{p1}, C_{p2}, ..., C_{pm}\}$. Respectively assume that at each time slot t, each user can only request one file. This paper defines $C = \{c_1, c_2, ..., c_i, ..., c_R\}$ to denote the file requested by the user in the cache space of the small base station, where $c_i \in C$ denotes the i content block requested by the user. If the request content is cached, then $c_i = 1$, or else $c_i = 0$. Then the optimization objective is to maximize the cache hit rate, as shown in Equation (8):

$$\max\sum_{i=1}^{N} c_i L \tag{8}$$

In Formula (8), $c_i \in \{0,1\}, \forall i \in N$, $L \leq \min\{C_c, C_p\}$.

For the evaluation of prediction accuracy, this paper often uses mean square error to evaluate, as shown in Formula (9):

$$MSE = \frac{\sum_{(u,i)\in\tau}(r'_{ui} - r_{ui})^2}{|\,\tau\,|} \tag{9}$$

In Formula (9), MSE is the mean square error, τ is the test set, r'_{ui} is the content prediction score, and r_{ui} is the content actual score.

To further assess the accuracy of the prediction, *N* most popular items are recommended for a user based on their actual data. R_u represents the users' prediction of the content of the collection, T_u is the test set of user favorite items, and you can use the accuracy and recall rate to evaluate the effect. The precision and recall rates are shown in Equations (10) and (11):

$$Precision = \frac{\sum_u |\,R_u \cap T_u\,|}{\sum_u |\,R_u\,|} \tag{10}$$

$$Recall = \frac{\sum_u |\,R_u \cap T_u\,|}{\sum_u |\,T_u\,|} \tag{11}$$

Experimental Results

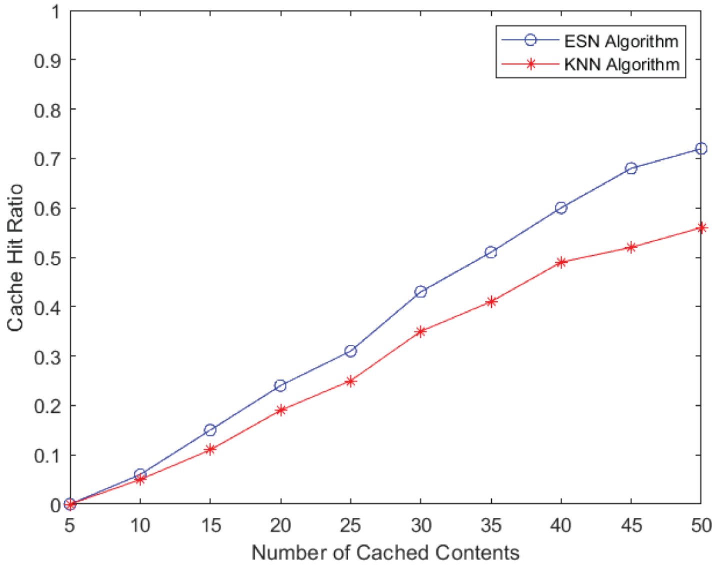
The Impact of Different Caching Methods on Cache Hit Rate

In this section, we illustrate the performance of this paper's proposed scheme with experimental simulation results. First, this study is compared with some classical caching models and papers using different caching models in terms of cache hit rate (Tran et al., 2017; J. Zhang et al., 2019). Table 4 shows the comparison results. The cache hit rate is improved by 27% compared to the hierarchical cache-based model in the paper. Compared with the node scenario-based caching model proposed in the paper, the cache hit rate in this paper is improved by 28%. Compared with the traditional caching model, the caching model proposed in this paper obtains better results. Second, the effect of different caching algorithms on the cache hit probability is verified. Figure 4 shows the comparison of different caching algorithms in terms of cache hit rate. When the cached content is small, the cache hit rate of the *k*-nearest neighbor classification cache is higher than that of this paper's cache because the KNN (*K*-Nearest Neighbor) cache is the most popular content among all caches. However, as the amount of cached content increases, the cache hit rate of this method is higher, mainly because the algorithm does not only cache the most popular content types. Also, it is predicted that most users

Table 4. Comparison of Hit Rates for Different Cache Models

Model	Hit rate (%)
LCE (Leave Cache Everywhere)	32
LCD (Leave Copy Down)	39
PROB (Copy with Probability)	35
KNN (*K*-Nearest Neighbor)	58
Hierarchical Cache Model	45
Node Situational Degree Caching Model	44
Echo State Network Caching Model	72

Figure 4. Impact of Different Caching Methods on Cache Hit Ratio

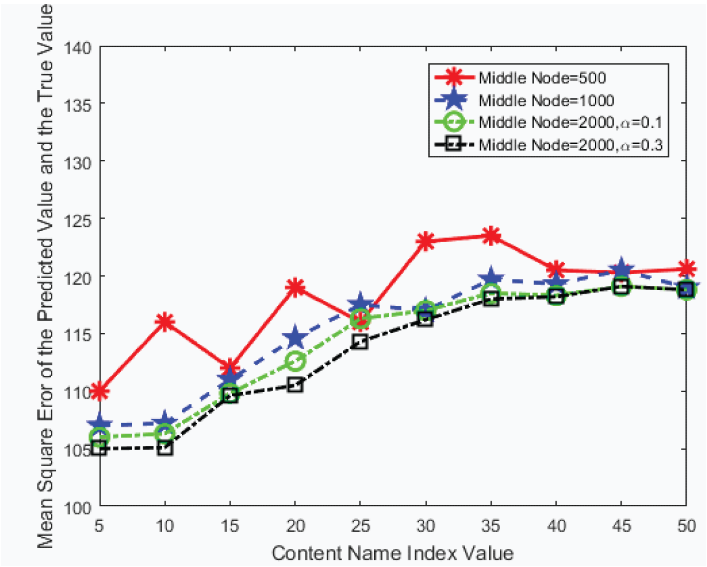


in the same clustering region have the same content request, so the cache hit rate increases with the increase in the number of caches.

Mean Square Error Between Predicted and Actual Values Under Different Intermediate Nodes

As shown in Figure 5, the abscissa is the index value of the content name, and the ordinate is the mean square error between the predicted value and the true value of the content. As the number of intermediate nodes *N* increases, the mean square error between the predicted value and the true value gradually decreases, indicating that the prediction results based on the echo state network model proposed in this paper are closer to the true value, and the final number of intermediate nodes *N* of the model is also obtained through this experiment. At the same time, when other parameters are fixed, the higher the update rate of the middle layer is, the more accurate the prediction result is. When the update rate is 0.3, the prediction effect is the best.

Figure 5. Mean Square Error Between Predicted and True Values Under Different Intermediate Nodes



Accuracy and Recall Rate Experimental Data Comparison

The dataset used in this paper is a five-point scoring system. In order to compare the experimental data of accuracy rate and recall rate, this paper assumes that users like the project when the score of the project is greater than or equal to three points, and dislike it when it is less than three points. Accuracy rate and recall rate will be calculated according to Formula (10) and Formula (11). Considering that the predicted content number *N* also has a certain influence on the evaluation of accuracy and recall rate, *N* is set to 5, 10, 15, 20, and 25, respectively, for five groups of experiments. The experimental results are shown in Figure 6 and Figure 7, respectively. Through the comparison of the above table data, it can be clearly seen that the algorithm in this paper is superior to the *k*-nearest neighbor algorithm in terms of accuracy and recall rate. When only considering the impact of the predicted content *N*, it is not difficult to see that the accuracy and recall rate are also increasing with the increase of *N*, which shows that this paper's algorithm has a strong impact on the recall rate of accuracy.

CONCLUSION

In this paper, propose a novel caching method to improve the cache hit rate of the user edge small base station cache. In this method, this paper considers the importance of both users and content, and optimizes the overall cache hit rate of the system by combining the content caching strategy with the user content request. Firstly, this paper clusters users by an AIC-based clustering algorithm. Second, use the machine-learning framework of the echo state network to predict the content preferences of the user community, and then select the cached content based on the predicted results. In this paper's model, select one of the most popular content types and predict the most likely popular content to be cached in two cache units. The simulation results show that the scheme improved the cache hit rate by 16%. When the update rate is 0.3, the mean square error is the smallest and the prediction effect is the best. At the same time, the prediction accuracy and recall rate have been significantly improved. In further research in the future, we propose to analyze user interest preferences by means of user portraits, and generate a user history preference matrix by means of multi-layer classifiers and deep learning models to learn precisely what users are likely to browse next, so that edge devices can complete traffic preloading more accurately and further optimize user experience.

Figure 6. Precision Data Comparison

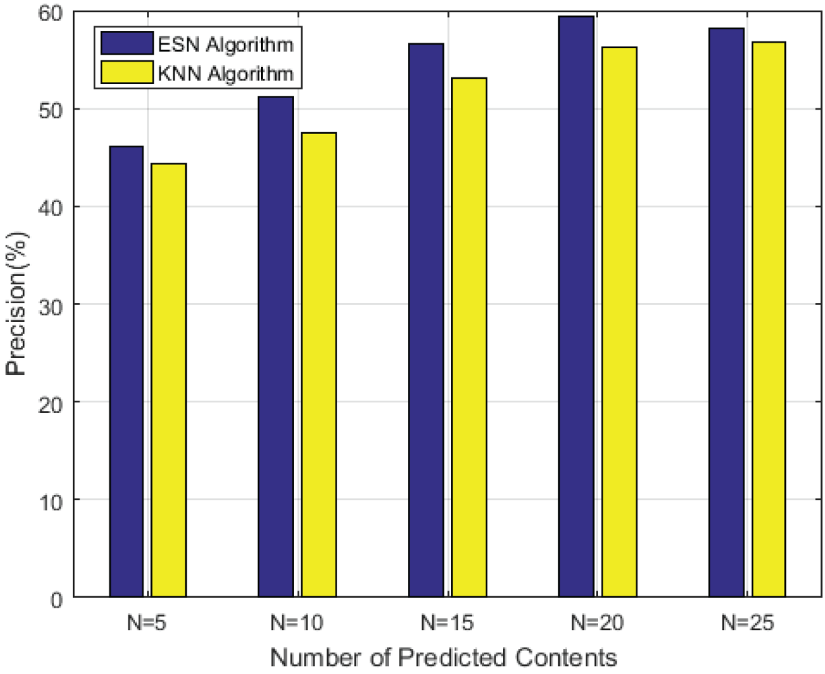
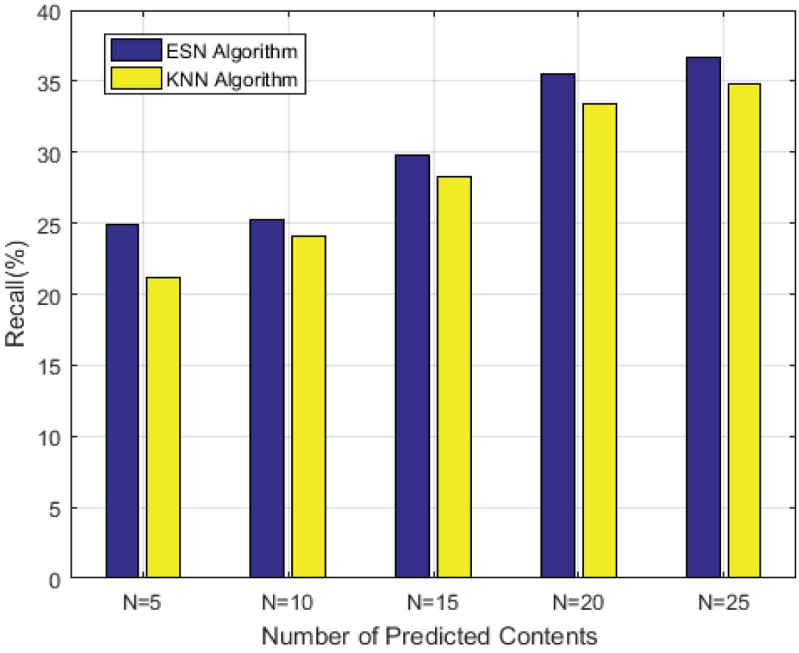


Figure 7. Recall Rate Data Comparison



REFERENCES

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723. doi:10.1109/TAC.1974.1100705

Bastug, E., Bennis, M., & Debbah, M. (2014). Living on the edge: The role of proactive caching in 5G wireless networks. *IEEE Communications Magazine*, *52*(8), 82–89. doi:10.1109/MCOM.2014.6871674

Chen, M., Saad, W., Yin, C., & Debbah, M. (2017). Echo state networks for proactive caching in cloud-based radio access networks with mobile users. *IEEE Transactions on Wireless Communications*, *16*(6), 3520–3535. doi:10.1109/TWC.2017.2683482

Chhangte, L., Karamchandani, N., Manjunath, D., & Viterbo, E. (2021). Towards a distributed caching service at the wifi edge using Wi-Cache. *IEEE eTransactions on Network and Service Management*, *18*(4), 4489–4502. doi:10.1109/TNSM.2021.3105496

Cui, H., Liu, X., & Li, L. (2012). The architecture of dynamic reservoir in the echo state network. *Chaos (Woodbury, N.Y.)*, *22*(3), 033127. doi:10.1063/1.4746765 PMID:23020466

Du, X., Xu, K., Li, T., Zheng, K., Fu, S., & Shen, M. (2021). Traffic control in data center networks: State of the art and trends. *Journal of Computational Science*, *44*(07), 1287–1309.

Ferdousi, S., Dikbiyik, F., Habib, M. F., Tornatore, M., & Mukherjee, B. (2015). Disaster-aware datacenter placement and dynamic content management in cloud networks. *Journal of Optical Communications and Networking*, *7*(7), 681–694. doi:10.1364/JOCN.7.000681

Hassine, N. B., Milocco, R., & Minet, P. (2017, March 29–31). *ARMA based popularity prediction for caching in Content Delivery Networks* [Conference paper]. Wireless Days conference, Porto, Portugal. doi:10.1109/WD.2017.7918125

Hu, M., Chen, Y., & Huang, H. (2021). Edge network caching strategy based on social relationship awareness. *Jisuanji Yingyong Yanjiu*, *38*(6), 1825–1829.

Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, *304*(5667), 78–80. doi:10.1126/science.1091277 PMID:15064413

Jmal, R., & Chaari Fourati, L. (2017). Content-centric networking management based on software defined networks: Survey. *IEEE eTransactions on Network and Service Management*, *14*(4), 1128–1142. doi:10.1109/TNSM.2017.2758681

Krishnendu, S., Bharath, B. N., Bhatia, V., Nebhen, J., Dobrovolny, M., & Ratnarajah, T. (2022, March 18). Wireless edge caching and content popularity prediction using machine learning. *IEEE Consumer Electronics Magazine,* 3160585–3160594. 10.1109/MCE.2022.3160585

Li, K., Shen, H., Chin, F. Y., & Zhang, W. (2007). Multimedia object placement for transparent data replication. *IEEE Transactions on Parallel and Distributed Systems*, *18*(2), 212–224. doi:10.1109/TPDS.2007.29

Li, L., Liu, H., & Lu, L. (2020). Probabilistic caching content placement method based on content centrality. *Jisuanji Yanjiu Yu Fazhan*, *57*(12), 2648–2661.

Naeem, M. A., Nor, S. A., Hassan, S., & Kim, B. S. (2018). Performances of probabilistic caching strategies in content centric networking. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 58807–58825. https://doi.org/10.1109/ACCESS.2018.2872614

Ortin, S., Soriano, M. C., Pesquera, L., Brunner, D., San-Martín, D., Fischer, I., Mirasso, C. R., & Gutiérrez, J. M. (2015). A unified framework for reservoir computing and extreme learning machines based on a single time-delayed neuron. *Scientific Reports*, *5*, 14945.

Poularakis, K., Iosifidis, G., & Tassiulas, L. (2014). Approximation algorithms for mobile data caching in small cell networks. *IEEE Transactions on Communications*, *62*(10), 3665–3677. https://doi.org/10.1109/TCOMM.2014.2351796

Qiao, J., Li, F., Han, H., & Li, W. (2016). Growing echo-state network with multiple subreservoirs. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(2), 391–404. https://doi.org/10.1109/TNNLS.2016.2514275

Ren, J., Tian, H., Fan, S., Lin, Y., Nie, G., & Li, J. (2020). Uav deployment and caching strategies based on user preference prediction. *Journal of Communication*, (6), 1–13. doi:10.11959/j.issn.1000-436x.2020104

Serhane, O., Yahyaoui, K., Nour, B., & Moungla, H. (2021). A survey of ICN content naming and in-network caching in 5G and beyond networks. *IEEE Internet of Things Journal, 8*(6), 4081–4104. 10.1109/JIOT.2020.3022243

Shrisha, H. S., & Boregowda, U. (2022). Quality-of-service-linked privileged content-caching mechanism for named data networks. *Future Internet 14*(5), 157. 10.3390/fi14050157

Song, Q., & Feng, Z. (2010). Effects of connectivity structure of complex echo state network on its prediction performance for nonlinear time series. *Neurocomputing*, *73*(10–12), 2177–2185. https://doi.org/10.1016/j.neucom.2010.01.015

Thar, K., Ullah, S., Lee, D. H., & Hong, C. S. (2016). On adaptive pre-fetching and caching the contents in content centric networking. *2016 International Conference on Information Networking,* 141–146. 10.1109/ICOIN.2016.7427103

Tran, T. X., Hajisami, A., & Pompili, D. (2017). Cooperative hierarchical caching in 5G cloud radio access networks. *IEEE Network*, *31*(4), 35–41. https://doi.org/10.1109/MNET.2017.1600307

Xu, D., & Xiao, Y. (2019). Website user behavior prediction based on machine learning technology. *Modern Electronic Technology, 42*(4), 94–96, 100.

Zeng, R., You, J., Li, Y., & Han, R. (2022). An ICN-based IPFS high-availability architecture. *Future Internet, 14,* 122. 10.3390/fi14050122

Zhang, H., Hu, B., Wang, X., Xu, J., Wang, L., Sun, Q., & Wang, Z. (2021). Self-organizing deep belief modular echo state network for time series prediction. *Knowledge-Based Systems*, *222*(17), 107007. https://doi.org/10.1016/j.knosys.2021.107007

Zhang, J., Du, C., Cai, Z., Wang, W., & Wu, Z. (2019). Content-centric network caching strategy based on node situational degree. *International Journal of Performability Engineering*, *5*(8), 2190–2198. https://doi.org/10.23940/ijpe.19.08.p19.21902198

Zhang, M., Tang, J., Rao, Y., Luo, H., & Zhang, H. (2017). Degree-based probabilistic caching in content- centric networking. *China Communications*, *14*(3), 158–168. https://doi.org/10.1109/CC.2017.7897331

Zhu, C., Cheng, G., Hu, Y., & Wang, Y. (2016). A caching strategy for internet plus TV based on popularity prediction. *Jisuanji Yanjiu Yu Fazhan*, *53*(04), 742–751. https://doi.org/10.7544/issn1000-1239.2016.20151143

Zhu, D., Liang, J., Li, T., Zhang, H., Geng, L., Wu, D., Zhang, T., & Liu, Y. (2020). A survey of security technologies in content-centric networking. *Journal of Information Security*, *5*(05), 121–143. https://doi.org/10.19363/J.cnki.cn10–1380/tn.2020.09.09

Zengyu Cai received his master's degree in computer application technology from Northeast Normal University, Changchun, China, in 2006. He is an associate professor at Zhengzhou University of Light Industry. His research interests include trusted computing, plan recognition, and information security.

Xi Chen is a master's student from the School of Computer and Communication Engineering, Zhengzhou University of Light Industry. Her research interests are content-centric network and artificial intelligence.

Jianwei Zhang received his Ph.D. degree in computer application technology from PLA Information Engineering University in 2010. He is a professor at Zhengzhou University of Light Industry. His research interests include broadband information network and network security.

Liang Zhu received his Ph.D. degree in computer science and technology from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in October 2017. He is currently a lecturer with the Institute of Computer and Communication Engineering at Zhengzhou University of Light Industry, Henan, China. His current research interests include mobile social network, personalized service recommendation, and preserving privacy.

Xinhua Hu is a master's student at the School of Computer and Communication Engineering, Zhengzhou University of Light Industry. His research interests cover network traffic prediction and network security.