

Retraction

Retracted: Research on Grid Monitoring Algorithm of Educational Resources Based on Economic Structure Transformation

Security and Communication Networks

Received 25 July 2023; Accepted 25 July 2023; Published 26 July 2023

Copyright © 2023 Security and Communication Networks. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] M. Wang and J. Liu, "Research on Grid Monitoring Algorithm of Educational Resources Based on Economic Structure Transformation," *Security and Communication Networks*, vol. 2022, Article ID 2988308, 12 pages, 2022.

Research Article

Research on Grid Monitoring Algorithm of Educational Resources Based on Economic Structure Transformation

Meihui Wang and Jia Liu 

Capital University of Business and Economics, Beijing 100070, China

Correspondence should be addressed to Jia Liu; zfr791557@163.com

Received 8 November 2021; Revised 8 December 2021; Accepted 19 January 2022; Published 15 February 2022

Academic Editor: Jian Su

Copyright © 2022 Meihui Wang and Jia Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to overcome the problems of the traditional educational resource grid monitoring algorithm, such as high signal noise, high data acquisition time, and high monitoring error rate, an educational resource grid monitoring algorithm based on the transformation of economic structure was proposed. This paper analyzes the structure of educational resource grid and constructs the monitoring structure of educational resource grid by using hierarchical tree structure model. With the support of this architecture, information sensors and data collection detectors are used to collect relevant data, and convolutional neural networks are used to denoise the collected data. According to the processed data, the educational resource grid under the transformation of economic structure is monitored to obtain relevant monitoring results. Experimental results show that the maximum value and minimum value of signal noise are 15 dB and 9 dB, respectively, the data acquisition time is always lower than 0.3 s, and the monitoring error rate is always below 4%, which has high practical application value.

1. Introduction

In recent years, with the popularization of information technology and the development of digital campus, primary and secondary schools all over the country are vigorously promoting the construction of campus network and digital education resources and have accumulated a large number of educational and teaching resources; the form and content of these resources show diversity. It includes teaching plans mainly based on electronic documents, teaching courseware mainly based on demonstration documents, and high-quality course presentations mainly based on video and audio [1]. After years of accumulation, these resources have become a considerable amount and become a valuable treasure in the development of China's education [2]. However, due to the various forms of resources in educational institutions, the management methods of educational resources in different institutions are different, and the construction of educational resources has its own characteristics, which makes educational resources present heterogeneity [3]. The shared platform and mechanism of education make the use of educational resources often limited to

local or small-scale local area networks; it is difficult to share educational resources between different schools, which is not conducive to communication and common development among educational institutions [4]. In particular, schools tend to establish their own resource management platform, which greatly increases the cost of repeated development, resulting in a huge waste of resources. Therefore, realizing effective sharing of educational resources without infringing intellectual property rights has become an important challenge and urgent problem to be solved for educational informatization [5].

Grid is a new type of network computing platform after the world wide web. It aims to provide users with a comprehensive infrastructure to share all kinds of resources including web pages. By running a management software that supports the grid mechanism on each node that accesses the grid, the loose resources on the grid are closely connected, so as to realize resource sharing [6, 7]. In the grid environment, resources distributed in different regions and on different hosts can be organized flexibly and effectively to form a virtual organization (VO) to work together to complete computing tasks. Grid breaks the traditional restrictions imposed on resources and

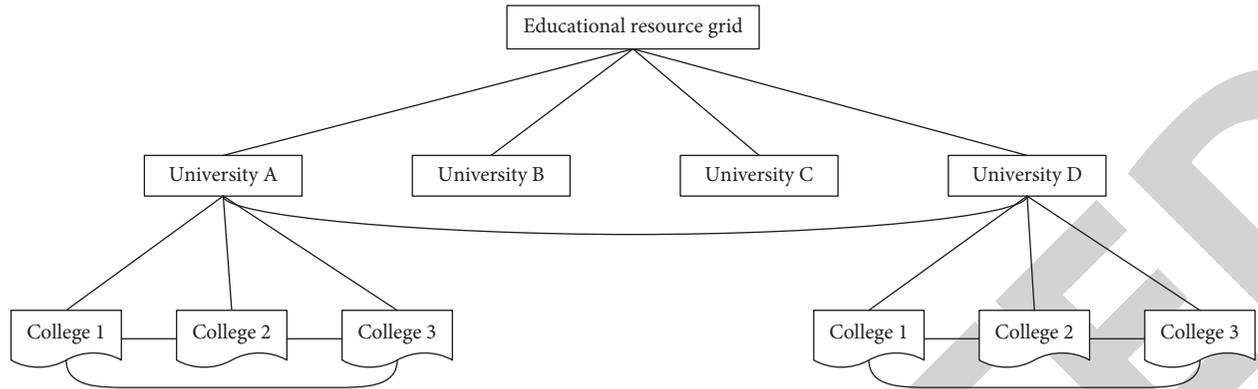


FIGURE 1: Logical structure of educational resource grid.

manages a large number of heterogeneous resources in a unified manner. The inherent characteristic of grid technology is to integrate resources in distributed computing environment to achieve maximum sharing of resources and maximum play of functions. Moreover, grid has the characteristics of distribution and sharing, self-similarity, dynamic and diversity, autonomous management, and multiple management, which is very suitable for the dynamic integration of complex educational resource sites. It is of great significance to study how to realize the monitoring algorithm of educational resource grid.

However, the current educational resource grid monitoring algorithms based on fuzzy set and RSS and those based on JNI and web service technology are difficult to adapt to the needs of educational resource grid monitoring under the background of economic structure transformation due to the low level of technology used, leading to the reduction of resource allocation balance. Therefore, this paper designs a new grid monitoring algorithm of educational resources based on economic structure transformation to solve the problems of traditional algorithms and improve the balance of educational resources allocation. For the fuzzy set, it is a class of objects with a continuum of grades of membership. Such a set is characterized by a membership (characteristic) function which assigns to each object a grade of membership ranging between zero and one. The notions of inclusion, union, intersection, complement, relation, convexity, etc., are extended to such sets, and various properties of these notions in the context of fuzzy sets are established.

The framework of the paper is as follows: in Section 1, we briefly introduced the background information of the educational resource monitoring and we also listed some difficulties in the field of monitoring algorithm. In Section 2, we analyzed the grid structure of educational resources. Section 3 presents the proposed grid monitoring algorithm for education resources. In Section 4, the experiments and the corresponding results are illustrated and analyzed. Finally, Section 5 concludes this study.

2. Grid Structure of Educational Resources

Grid architecture is about how to build grid technology; it includes two levels of meaning: first, identify which parts the grid system consists of and clearly describe the functions,

purposes, and characteristics of each part; the second is to describe the relationship between each part of the grid and explain how to organically combine all parts together to form a complete grid system. The actual grid system usually includes the following three layers:

Network resource layer: it constitutes the hardware foundation of the grid system, including various computing resources, such as supercomputers, valuable instruments, visualization equipment, and existing application software, which are connected by network devices. The grid resource layer only realizes the physical connection of computing resources, but from the point of view of logic, these resources are still isolated, and the resource sharing problem has not been solved. Therefore, the effective sharing of wide-area computing resources must be accomplished through grid middleware based on grid resource layer.

Grid middleware: it is a series of tools and protocol software, whose function is to shield the distribution and heterogeneity of computing resources in grid resource layer and provide transparent and consistent interface for grid application layer. At the same time, the programming interface and the corresponding development environment are provided to support the development of grid applications.

Grid application layer: it is the concrete embodiment of user needs. With the support of grid operating system, grid users can use its tools or environment to develop various application systems. Whether the application system can be developed to solve all kinds of large computing problems is the key to evaluate the advantages and disadvantages of grid system.

Based on the problem of educational resources sharing, this paper studies the new structure of educational resources grid and realizes the sharing of educational resources among universities through the grid. Users can query and browse the educational resources that meet their needs, as shown in Figure 1.

Figure 1 shows the logical structure of educational resource grid, which forms a hierarchical tree structure from top to bottom, with the root node as the center node. The leaves of the tree are the lowest layer of the grid, the

colleges of the school, which are responsible for the maintenance of their respective resources. Each intermediate node of the tree is each university. These intermediate nodes are responsible for storing the attribute information of all resources within their jurisdiction [8]. At the same time, they can establish a copy of resources that are requested by users in this area with a large delay. In order to improve the reliability of the system, a ring structure is established between sibling nodes. Note that this structure is the logical structure of the educational resource grid, which combines the ring with the hierarchical structure. It not only has the characteristics that the hierarchical structure is easy to expand but also has the advantages of high availability and reliability of the data of the planar structure. In this structure, educational resources are scattered in each school, and the middle node stores the directory information of resources within its jurisdiction and stores the copies of corresponding resources according to the copy creation policy. Under the condition of satisfying the existing distributed storage of educational resources, this structure realizes the hierarchical management of educational resources, so as to avoid the problems of centralized storage management, difficult implementation, distributed storage peer-to-peer management, and large network traffic.

At present, there are a large number of management systems for various types of educational resources. However, most of these systems use a centralized management model for educational resources, which limits the autonomy of resource owners, and users are not motivated to share resources and cannot make existing education. Resources are fully utilized, so there is an urgent need for a way to share resources. Currently, there is no platform that can realize the distributed management of these educational resources. To solve this problem, this paper introduces an education grid architecture based on distributed education resource management. However, due to the highly dynamic characteristics of the grid, various internal resources are in frequent dynamic changes, and there are few static and persistent sharing relationships, which may bring serious obstacles to the use of educational grid resources. It may even lead to a decrease in the performance of the entire grid. Therefore, a real-time dynamic and scalable resource monitoring mechanism is provided, which is responsible for collecting node resource usage and resource current status information in the grid so that users and applications can timely grasp all the current resources added to the education grid. Status information has become a top priority.

3. Design of Grid Monitoring Algorithm for Educational Resources

In order to promote the sharing of educational resources in grid environment, it is necessary to monitor and manage the resources. This paper designs the grid monitoring architecture of educational resources. The monitoring architecture of educational resource grid uses object technology and hierarchy to represent various educational resources, manages nodes by domain, and deploys

distributed resource monitoring strategy in the education grid. In the hierarchical tree structure model, the education grid is divided into several virtual organizations VO according to different geographical locations, as shown in Figure 2.

In the monitoring architecture diagram of educational resource grid, the leaf node of the tree is the lowest layer of the grid, that is, the resource node of each school, which is responsible for the maintenance of their own educational resources. Each middle node of the tree stores the directory information of all node resources of the school. It can quickly find the basic information and real-time status information of all nodes in the subtree rooted by this node. The root node of the tree is an interorganization monitoring management center that stores and monitors resource node catalogs in the entire system.

3.1. Data Collection

3.1.1. Information Sensor. The information sensor is directly responsible for the collection of real-time status data of resource nodes, describes the collected information with XML, and transmits it to the data collection detector, providing a unified interface to the upper data collection detector. XML can separate data from HTML. That is, the data can be stored in the XML document outside the HTML file so that developers can concentrate on using HTML to display and layout the data and ensure that the HTML file will not need to be changed when the data are changed, so as to facilitate the maintenance of the page. XML can also store data in HTML pages in the form of “data islands,” and developers can still focus on formatting and displaying data using HTML. XML can be used to exchange data. Data can be exchanged between incompatible systems based on XML. There are many forms of data stored in computer systems and database systems. For developers, the most time-consuming work is to exchange data between systems all over the network. Converting data into XML format will greatly reduce the complexity of data exchange and make these data read by different programs.

In the implementation, Java program calls VC++ dynamic link library JNI technology and encapsulates web services technology in Java Applet. For the JNI technique, firstly, it is the abbreviation for the term “Java Native Interface” and it can be used in many different situations. For example, functions in Java programs call functions in Native programs, where Native generally refers to functions written in C/C++. Function calls in Native programs Functions in other programs. Among them, Java multithreading mechanism is adopted to make data collection thread and data display thread start at the same time and keep synchronization [9, 10]. Keytool-genkey-alias after certificate creation, run the jarsigner-keystore command to digitally sign the Applet and encapsulate it in the information sensor to obtain resource information in a timely and accurate manner. However, because the signal of the information sensor will be interfered by many factors, it is necessary to reconstruct the signal of the sensor.

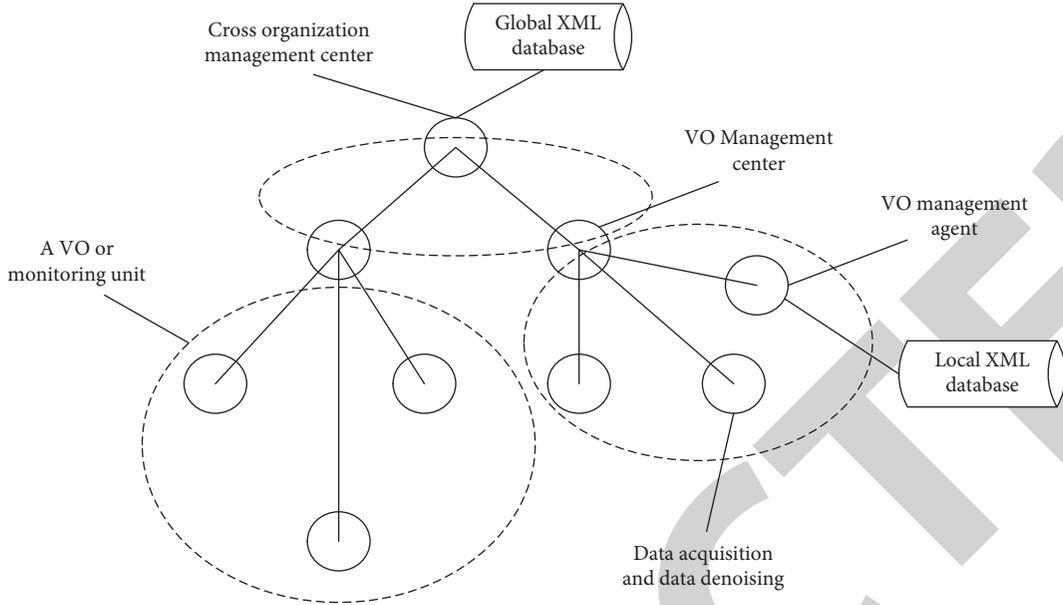


FIGURE 2: Educational resource grid monitoring architecture.

K-L transform, also known as principal component analysis, is also called feature vector transform and an orthogonal transform by extracting waveform features from signal waveform data and reconstructing waveform [11]. As a mainstream in principal component analysis, K-L transform not only reduces the dimension of the original data to the greatest extent but also retains the information in the original data to the greatest extent, with low information loss, so here we use the K-L transform to construct waveform. The data observed in data acquisition is the superposition of signal and noise, and noise is caused by various disturbances. In order to reduce the noise to the minimum, K-L transformation is adopted. Since a signal contains N data and the sensor node is P , the signal can be represented as a two-dimensional matrix X , whose element is x_{ij} , and i is the node serial number and sequence j , namely,

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1p} \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{ip} \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ x_{N1} & \cdots & x_{Nj} & \cdots & x_{NP} \end{bmatrix} \quad (1)$$

Then, the given signal matrix is $X((x_i), i = 1, 2, 3, \dots, n)$, where x_i is the row vector of X [12]. Its mean vector is $mX = E(X) = [E(x_1), E(x_2), \dots, E(x_n)]^T$, and the covariance matrix C_X of X is [13]

$$C_X = E[(X - mX)(X - mX)]^T = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1N} \\ c_{21} & c_{22} & \cdots & c_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N1} & c_{N2} & \cdots & c_{NN} \end{bmatrix}, \quad (2)$$

where C_X is the real symmetric orthogonal matrix of $n * n$, so there must be an orthogonal matrix $U = (u_1, u_2, \dots, u_n)$, so that

$$U^T C_X U = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & \cdots \\ \vdots & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & \lambda_N \end{bmatrix}. \quad (3)$$

In the formula, λ_1 is the eigenvalue of covariance matrix C_X , and u_i is the normalized zero flushing eigenvector corresponding to λ_1 , namely, $C_X u_i = \lambda_1 u_i$, $C_X u_i = \lambda_i u_i, i = 1, 2, \dots, N$. Orthogonal transformation of random vector X :

$$Y = U^T X = \begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,M} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N,1} & y_{N,2} & \cdots & y_{N,M} \end{bmatrix}. \quad (4)$$

Then, the N -dimensional random vector Y is called the K-L transformation of X , or $y_i (i = 1, 2, 3, \dots, N)$ is the principal component of X . Here, y_1 is the first principal component, and y_N is the N -th principal component, each component is unrelated to each other, and the variance of the i -th principal component is equal to the i -th eigenvalue of U , then

$$\begin{aligned} X &= UY = [U_1, U_2, \dots, U_N], Y = y_1 U_1 + y_2 U_2 + \dots + y_N U_N \\ &= \sum_{i=1}^N y_i U_i. \end{aligned} \quad (5)$$

It is called the K-L expansion of random vector X .

Assuming that the main signal X_s is composed of the first K principal components [14], the effective signal can be reconstructed by the following formula, then the reconstructed signal is

$$X_s = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1k} \\ k_{21} & k_{22} & \cdots & k_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ k_{N1} & k_{N2} & \cdots & k_{Nk} \end{bmatrix} \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1M} \\ y_{21} & y_{22} & \cdots & y_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k1} & y_{k2} & \cdots & y_{NM} \end{bmatrix}. \quad (6)$$

K-L transform filtering is based on the coherence difference between multichannel signals, based on the statistical characteristics of the centralized correlation moment (covariance) as the theoretical basis, through the selection of principal components in the transform domain to achieve the filtering purpose.

3.1.2. Data Collection Detector. The data acquisition detector first runs to the registration center and is mainly responsible for collecting the monitoring information provided by the information sensor and effectively manages the collected information in the XML model. It is also responsible for receiving a monitoring service information for monitoring resource nodes. The request information is sent to the requester, and SSL encryption is used in the data transmission process to ensure security. The full English name of SSL is “secure sockets layer” and the Chinese name is “secure socket layer protocol layer.” It is a security protocol based on web applications proposed by Netscape. SSL protocol specifies a mechanism to provide data security layering between application protocols (such as HTTP, TELNET, NMTP, and FTP) and TCP/IP protocols. It provides data encryption, server authentication, message integrity, and optional client authentication for TCP/IP connections. Meanwhile, SSL session is mainly divided into three steps which are as follows:

- Step1. The client requests and verifies the certificate from the server;
- Step2. Both parties negotiate to generate “session key,” pairing key.
- Step3. Both parties use “session key” for encrypted communication. By analyzing the monitoring data, the prediction process can predict the future performance of the resource and respond to the client’s query.

3.2. Data Denoising. Convolutional neural network (CNN) is different from artificial neural network (ANN). When facing some complex image recognition tasks, CNN can solve various tasks of image processing well by using its precise and simple framework. This is also the reason why CNN is widely popular among deep learning tools [15]. CNN is usually composed of one or more convolutional layers and is a deep neural network with convolutional structure. The typical topology of convolutional neural network is shown in Figure 3.

It can be observed from Figure 3 that a typical CNN architecture is composed of multiple layers, and the whole network is composed of input layer, hidden layer, and output layer. The hidden layer is a kind of multilayer feedforward network, which is composed of one or more pairs of convolution layer, pooling layer, and full connection layer alternately connected. Each convolutional layer in the hidden layer consists of a set of filters. Each filter has a certain perceptual learning domain, which is also the key to learning and identifying input data. The function of pooling layer is to assist the existence of convolution layer, which reduces the spatial size of data after convolution processing and reduces network parameters and computation [16]. The core of the network still lies in the convolutional layer and pooling layer, and the relevant content will be introduced in detail next. In convolutional neural networks, convolution is the most basic and important operation, also known as feature extraction layer. The 2D data vector to be processed is taken as the input, and then the learnable convolution kernel is used for convolution operation with the feature graph output from the previous layer to obtain the characteristics of local information to improve the processing performance on 2D data. Finally, the output result is obtained by adding bias and activation function. The calculation formula of convolution layer is as follows:

$$y_{mm} = f \left(\sum_{j=0}^{q-1} \sum_{i=0}^{p-1} x_{m+1,n+y} w_{ij} + b \right), 0 \leq m \leq M, 0 \leq n \leq N. \quad (7)$$

In formula (7), x represents the input two-dimensional data; w represents the convolution filter kernel of size $p \times q$; b indicates bias; and y represents the feature graph with size $M \times N$ output after the nonlinear activation function $f(\cdot)$ through convolution operation. The convolution layer constitutes a learnable filter through the local receptive field connected to the upper layer by each neuron input [17]. The local features of the data are extracted by convolution learning training in the perceptual learning domain of the filter through the convolution kernel. When these local features are extracted, the position relationships of other features are also determined. The convolutional structure of CNN replaces feature extraction of the full connection layer in ANN, reduces the amount of memory occupied by the deep network, reduces the amount of network parameters, and alleviates the over-fitting phenomenon of the model.

The size of the convolution kernel and the sliding step determine the size of the output. Its calculation formula is as follows:

$$y = \frac{x - k}{\text{stride} + 1}. \quad (8)$$

In formula (8), x represents the size of the input; stride represents the step size; k represents the convolution kernel size; and y indicates the size of the output data. After many complex convolutional operations, the number of parameters of feature map gradually increases, which is easy to produce over-fitting phenomenon. To solve this problem, pooling operation is

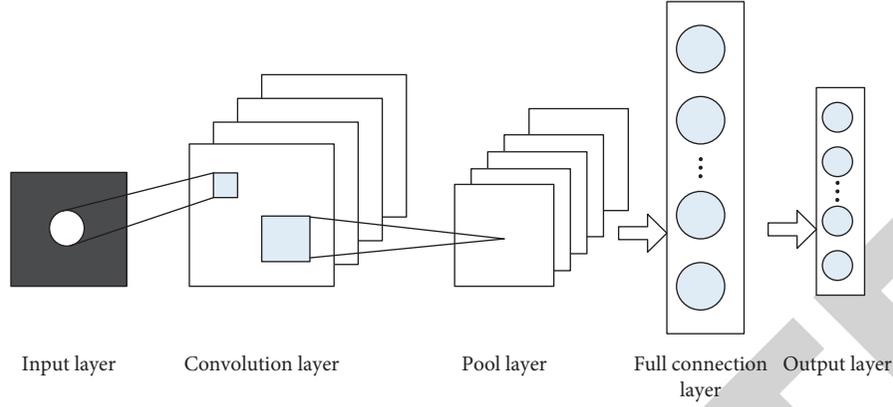


FIGURE 3: Typical topology of convolutional neural network.

usually used. Its function is to reduce the parameter scale by scaling the local area of data and make statistics on the values of feature mapping maps of different positions output after the convolution operation and then replace the corresponding values on the original feature mapping map as the output results. The common pooling operations are maximum pooling and average pooling. The maximum pooling algorithm selects the maximum value of all values in the pool as the pooled value. The average pooling algorithm calculates the average of all the values in the pool as the pooled value. However, these two pooling methods have some defects in training network. The drawback of maximum pooling is that when a feature appears multiple times, maximum pooling can only be observed once and only retain a maximum value. The location information of feature items is lost when the dimensionality reduction operation is performed. The drawback of average pooling is that the pool area is large and all elements are counted, making the calculation too complicated. Therefore, in the process of dimensionality reduction using pooling layer, effective signals will be lost, feature repetition will waste space, and high computational complexity will be caused. It is easy to cause over-fitting phenomenon in the training process. In this paper, the pooling layer in the network will be removed and the neural network structure containing only convolutional layer will be constructed.

The propagation algorithm of convolutional neural network is mainly composed of two parts, namely, forward propagation algorithm and back-propagation algorithm [18, 19]. The forward propagation algorithm calculates the CNN model from a series of input layers, hidden layers, and output layers to get the output results. Back-propagation algorithm is the reverse calculation of output layer, hidden layer, and input layer. In order to minimize the error between the model output and the actual results, the model was optimized by adjusting the weight of parameters. Through the loop iteration of forward propagation and back propagation, the weights between neurons are constantly updated and the training model is obtained to obtain the network with the minimum error value. Finally, the data denoising task is realized.

The forward propagation algorithm formula is as follows:

$$\bar{y}_i = h_{w,b}(x_i) = f(W^T x_i) = f \sum_{i=1}^n (W_i x_i + b). \quad (9)$$

Formula (9) is to input sample data (x_i, y_i) to the input layer of the network. Among them, $x_i, i = 1, 2, \dots, n$ input values, $y_i, i = 1, 2, \dots, n$ is the expected output, and x_i goes through the hidden layer to calculate the activation value of each layer, layer by layer. Finally, it is transmitted to the output layer to complete the forward propagation process, and the output result obtained is the predicted value, denoted as \bar{y}_i . Here, we use the ReLU function as our activation function and its formulation is as follows:

$$f(x) = \max(0, x). \quad (10)$$

Back-propagation algorithm is also known as BP algorithm. BP algorithm is an iterative algorithm, which is used to train the model and obtain the best combination of model parameters. Its parameters determine the accuracy of the output results of the model, and the parameters mainly refer to the weight of the hidden layer and the bias items and other information. BP algorithm is a process of constantly adjusting the parameters during the training of the model. The specific process of back-propagation algorithm is as follows: usually a loss function is defined to evaluate the model, and the commonly used loss function is root mean square error and cross entropy. In this paper, root mean square error is selected as the loss function, that is, the sum of the squares of the errors between the actual output and the predicted value, which is loss function E . The calculation formula of root mean square error is as follows:

$$E(W, b) = \frac{1}{2} \sum_{i=1}^n \|y_i - \bar{y}_i\|^2 = \frac{1}{2} \sum_{i=1}^n \|y_i - g[h_{w,b}(x_i)]\|^2. \quad (11)$$

In (11), y_i represents the actual output and \bar{y}_i represents the expected output. To build a model and update various parameters by adjusting the error, the error between the predicted value and the actual value becomes smaller and smaller. The smaller the error is, the better the model is. The training process of network structure is regarded as the extremum point of minimizing loss function $E(W, b)$.

Gradient descent method is often used to solve the extreme point to update the weight of the hidden layer. Formulas (12) and (13) represent the parameter matrix W and parameter vector b , and in the gradient descent method, the updated changes after each iteration.

$$w_{ij}^{(l+1)} = w_{ij}^{(l)} - \eta \frac{\partial}{\partial w_{ij}^{(l)}} E(W, b), \quad (12)$$

$$b_i^{(l+1)} = b_i^{(l)} - \eta \frac{\partial}{\partial b_i^{(l)}} E(W, b), \quad (13)$$

where l represents the number of layers; η represents the learning rate, which is the change used to train the model to realize parameter updating. $w_{ij}^{(l+1)}$ stands for the weight matrix between layer l and $l + 1$; if the number of nodes in layer l is m , the number of neurons in layer $l + 1$ is n , then the shape of $w_{ij}^{(l+1)}$ is $n * m$. The output data of each layer in the network are determined by the weight between layers. The errors generated between layers are put into the network in the form of weight for forward propagation. This process represents back propagation. After the output error is obtained, the weight can be used to back-propagate to the hidden layer, and the chain rule is used to obtain the partial derivative, which becomes the key step of the back-propagation algorithm. Through loss function $E(W, b)$ in formula (11), partial derivatives of weight $w_{ij}^{(l)}$ and bias term $b_i^{(l)}$ are obtained:

$$w_{ij}^{(l)} = \frac{\partial E(W, b)}{\partial w_{ij}^{(l)}} = \left[\frac{1}{n} \sum_{i=1}^n \frac{\partial}{\partial w_{ij}^{(l)}} E(W, b) \right] + \lambda w_{ij}^{(l)}, \quad (14)$$

$$b_i^{(l)} = \frac{\partial E(W, b)}{\partial b_i^{(l)}} = \frac{1}{n} \sum_{i=1}^n \frac{\partial}{\partial b_i^{(l)}} E(W, b).$$

The following is the calculation process of the convolutional network propagation algorithm: firstly, formula (10) is used for forward propagation calculation, and all activation values in the convolutional neural network are obtained through layer by layer calculation, and the output value of each neuron is $h_{w,b}(x_i)$.

In the case of known activation value, in order to obtain the residual of each output unit in the output layer, the nodes in the output layer are directly taken as the final output value of the unit, and then compared with the actual value y_i , the error $\delta_i^{(n)}$ is obtained. $\delta_i^{(n)}$ is calculated by the following formula:

$$\delta_i^{(n)} = \frac{\partial}{\partial z_i^{(n)}} E(w, b; x, y) = -\left(y_i - a_i^{(n)} \cdot f' \left(z_i^{(n)} \right) \right). \quad (15)$$

The error l of the i -th node in layer $\delta_i^{(l)}$ is calculated as follows:

$$\delta_i^{(l)} = \left(\sum_{j=1}^{S_{l+1}} w_{ji}^{(l)} \delta_j^{(l+1)} f' \left(z_i^{(n)} \right) \right). \quad (16)$$

Finally, it was substituted into formulas (11) and (12) to obtain the partial derivative:

$$\frac{\partial}{\partial w_{ij}^{(l)}} E(W, b) = a_j^{(l)} \delta_i^{(l+1)}, \quad (17)$$

$$\frac{\partial}{\partial b_i^{(l)}} E(W, b) = \delta_i^{(l+1)}. \quad (18)$$

After the partial derivatives of formulas (17) and (18) are obtained, the weight parameters of the whole neural network can be optimized in combination with the forward propagation and back-propagation algorithms, and the optimized neural network can be used to denoise the collected data, so as to improve the monitoring accuracy of the subsequent educational resources grid.

In our CNN denoising model, its hyperparameters include the learning rate, the number of kernels, the kernel size, and the convolutional stride. By the cross-validation, we choose the best values for those parameters, they are as follows: the learning rate is 0.0001, the number of kernels is 64, the kernel size is 3, and the stride is 1.

3.3. Data Storage. The monitored data are frequently updated, have a large amount of data, and have strong timeliness. Therefore, it is meaningless to store all the monitored data for a long time. However, some important and relatively stable data (such as the operating system) may need to be stored for historical analysis. ERGM uses two different storage methods for these two types of data: a RRD is used for short-term performance data storage. For some important and relatively stable data to be stored for a long time, XML is used for storage.

RRD database is a kind of database which uses circular queue mode to store. The physical storage space of data is fixed. A pointer indicates the current storage location. When new monitoring data need to be stored, the new data will be stored in the current location indicated by the pointer, and the pointer will move back accordingly. After all the space in the loop queue is occupied, the pointer points to the original space (the queue head), overwriting the previous content. In this way, the size of storage space does not grow gradually over time. But the drawback is obvious: historical data will not be saved. If the initial space is specified too large, the footprint will still be large. If the initial space is specified too small, only short-term data will be provided. To solve this problem, the circular database is set up in a variety of granularity. A database consists of a series of data of different sizes. The most fine-grained data are the monitored original data, while coarse-grained data are obtained based on the original data through certain functions, such functions generally have average, maximum, or minimum values, etc. In this way, a circular database can provide both short-term very accurate data and approximate summaries of long-term data. For a user's query, the RRD database automatically selects the most appropriate granularity of data to serve to the user based on the time range provided by the user. For important and relatively stable data that need to be stored for a long time, or for data that need to be stored entirely on important nodes, a circular database is not suitable. For these

data, ERGM stores the monitoring data as XML. XML is a readable form of text that users can browse or analyze without tools.

3.4. Educational Resource Monitoring Algorithm Design Based on Economic Structure Transformation. Due to the heterogeneity of resource endowment and market capacity, it is impossible for each region of a country to maintain the same economic growth rate. However, if the regional economic differences are large for a long time, the pattern will affect the efficiency of resource allocation and the spontaneous extension of the market, which is not conducive to improving the overall economic efficiency and maintaining sustainable growth. At the same time, regional economic growth has not had a negative impact on social order. In building a moderately prosperous society in an all-round way and pursuing modernization, China seeks not only overall economic growth, but all-round economic and social progress on the basis of growth so that the fruits of growth are relatively evenly shared among residents in different regions. Based on the above considerations, the recent government, under the concept of globalization and with the scientific concept of development, has placed the adjustment of regional disparities and the coordination of regional economic development in a more prominent position and promoted the transformation of economic structure. During the transition, there have also been some imbalances in the allocation of educational resources. To improve the quality of education and promote the transformation and development of modern education mode, it is necessary to design a new grid monitoring algorithm of educational resources under the background of information and economic structure transformation.

For grid monitoring of educational resources under the background of economic structure transformation, the most effective scheme is to reduce the amount of monitoring target data as much as possible without losing its statistical performance. Obviously, it is necessary to find an appropriate threshold for this purpose. Start by defining two important parameters G and Q related to the quality of resource monitoring.

The first parameter G is defined as follows:

$$G = 1 - \frac{N(t)}{N(t^0)}, \quad (19)$$

where $N(t)$ represents the number of samples collected by the monitoring algorithm, $N(t^0)$ represents the number of samples sampled by the theoretical sampling frequency, and t^0 represents the highest theoretical period (i.e., 1 s). The value of G ranges from 0 to 1. The higher the value of G is, the shorter the sampling period is, and the higher the calculation and bandwidth cost is.

The second parameter, Q (quality), represents whether the monitoring algorithm can accurately reflect the changes of system resources. The quality of Q must take into consideration two factors:

- (1) Monitoring is incomplete due to excessive sampling period
- (2) The ability to monitor resource changes with spike wave

Therefore, Q is defined as the combination of two important statistical parameters: $NRMSE$ is the normalized root mean square error, and $Fmeasure$ is the weighted mean value of the accuracy and recall rate of spike wave detection.

$$Q = \frac{Fmeasure + (1 - NRMSE)}{2}, \quad (20)$$

where the value range of $Fmeasure$ and $NRMSE$ is $[0, 1]$, and the value range of Q is $[0, 1]$. The reason for integrating $Fmeasure$ and $NRMSE$ pairs of resource evaluation is that cloud resource changes are extremely drastic, and only $NRMSE$ cannot accurately respond to its changes, while $Fmeasure$ can have a better detection effect on spines resources. Therefore, the accuracy of resource evaluation can be improved by integrating the two parameters. $Fmeasure$ calculation method is as follows:

$$Fmeasure = \frac{2 \cdot precision \cdot recall}{precision + recall}. \quad (21)$$

In the formula, $precision$ represents precision, $recall$ represents recall rate, and the closer the value of $Fmeasure$ is to 1, the higher the quality of detection.

The threshold value of the monitoring algorithm is expressed as the weighted sum of G and Q :

$$E = \omega \cdot G + (1 - \omega) \cdot Q, \quad (22)$$

where $\omega \in (0, 1)$ is a tuning constant set by the server side administrator. If $\omega > 0.5$, G is more important than Q . Conversely, G is less important than Q ; $\omega = 0.5$, both are of equal importance.

The existing monitoring algorithm collects samples in a fixed sampling period. Only when the new data is different from the historical data, the new data are saved and forwarded to the analysis module. Although this scheme achieved a higher value of G , it resulted in a higher detection error and a lower value of Q (more spikes were lost).

3.4.1. Algorithm Parameter Setting. The monitoring algorithm analyzes the collected data. When the resource is relatively stable, the number of monitoring data is reduced. When the resource changes dramatically, the number of monitoring data is increased. In this way, the computation and bandwidth costs are reduced, while ensuring that important system spike changes are not missed. The algorithm dynamically sets two key variables: sampling period t and variability Δ . The shorter the sampling period t , the greater the amount of data collected. Let t_m represent the minimum value of the sampling period and t_M represent the maximum value of the sampling period, obviously $t_M > t_m$. Δ represents the deviation of continuous samples. If Δ is lower, monitoring resources are considered stable. This article considers two parameters related to Δ :

Peak variability Δ_p represents the threshold of continuous intersample deviation. When $\Delta > \Delta_p$, it represents spike wave.

Fault-tolerant variability Δ_q represents continuous intersample deviation, and $\Delta_p \geq \Delta_q$ represents high variability. If the variation of monitoring data is too high, set the sampling period to t_m to capture the details of resource changes. The optimal values of t and Δ related thresholds are set as t_m^* , t_M^* , Δ_p^* , Δ_q^* .

3.4.2. Algorithm Training. In the training stage, the optimal threshold parameters t_m^* , t_M^* , Δ_p^* , Δ_q^* were solved. In the training phase, the number of training data samples was set to λ by maximizing E of formula (20). The algorithm training process is as follows:

In the initialization stage, the optimal value E^* of E is set to 0, and the minimum sampling period is set to t^0 , and X_{tmp} is set to the monitoring data sequence sampled in period t^0 . In cycle iteration, monitoring cycle t and variability Δ are iterated. AdaptiveMonitor is a data monitoring algorithm. The result data are stored in X variables. Then, quality parameters Q , G and threshold E are calculated. Finally, the optimal threshold parameters t_m^* , t_M^* , Δ_p^* , Δ_q^* are updated, and the resulting threshold is assigned to the core stage of the monitoring algorithm.

3.4.3. Implementation of Monitoring Algorithm. During initialization, the sampling period is set to the minimum value t_m^* , Δ to 0, and ϵ_{inc} to 10. This variable is used to trigger the minimum sampling period (t^0). After $\epsilon \cdot \lambda$ samples, a value of X is obtained. Variable k is the count of the sampling sequence X . If $k > \lambda$, the sampling of X is finished.

The main body of the monitoring algorithm is an infinite cycle. Firstly, real-time sampling is used to update the value of deviation Δ . If the value of Δ does not change, the sampling period length is increased; if the value of Δ changes greatly, the current sampling period is reduced. In the monitoring process, by calculating the Q parameters and the optimal parameters in the training stage, the real-time adaptive dynamic adjustment is realized, so as to achieve the balance between high accuracy sampling and low cost monitoring. Under this balance condition, the result is the final monitoring result of the education resource grid.

3.4.4. The Measurement Used for Evaluating the Algorithm. In our paper, we use the signal noise ratio (SNR) to evaluate the effectiveness of the algorithm. For the SNR, it is the ratio of the average power of the signal to the average power of the noise and can be calculated by

$$\text{SNR} = 10 * \lg\left(\frac{P_{\text{signal}}}{P_{\text{noise}}}\right) \text{ (dB)}, \quad (23)$$

where P_{signal} and P_{noise} are the average power of the signal and noise, respectively.

4. Experimental Design and Result Analysis

In order to evaluate the performance of educational resource grid monitoring algorithm based on economic structure transformation, this paper uses OptorSim grid simulator to simulate the strategy and compares the experimental results with the monitoring algorithm based on fuzzy set and RSS, as well as the monitoring algorithm based on JNI and web services technology.

The network topology used in the experiment is shown in Figure 4.

The topological structure consists of four layers, the lowest layer is the user and resource nodes in the grid; they are in different campus networks; the third layer has four university-level management nodes, each university-level management domain has three user nodes. Layers 2 and 1 correspond to county- and city-level regional hub nodes in the education resource grid. All data requests are initiated by leaf nodes, and the size of each requested data file is 1 GB. Initially, all data are stored on leaf nodes, and the storage space of other nodes does not store any data.

In order to distinguish the difference between the bandwidth of the educational resource grid based on the wide-area network and the campus network, the network bandwidth between the lowest nodes is set as 100 MB/s, and the bandwidth of the upper layer is set as 50 MB/s based on the traditional wide-area network. In addition, in order to distinguish the characteristics of higher nodes in the education resource grid with stronger storage and processing capabilities, the storage and processing capabilities of nodes are set, as shown in Table 1 (wherein, the node processing capabilities are realized by setting work nodes in CE in OptorSim). Initially, 200 data files were randomly distributed to the underlying nodes.

Firstly, the proposed algorithm is compared with the monitoring algorithm based on fuzzy set and RSS, as well as the monitoring algorithm based on JNI and web service technology. The lower the signal noise is, the higher the signal quality is. The comparison results are shown in Table 2.

By analyzing the data in Table 2, it can be seen that the maximum signal noise of the monitoring algorithm based on fuzzy set and RSS is 52 dB and the minimum is 38 dB. The maximum signal noise of the monitoring algorithm based on JNI and web service technology is 38 dB, and the minimum signal noise is 29 dB. The maximum signal noise of the proposed algorithm is 15 dB, and the minimum is 9 dB, which is far less than the two algorithms, indicating that the proposed algorithm has better signal denoising effect.

On the basis of the above experiments, the data acquisition time of the proposed algorithm is compared with the monitoring algorithm based on fuzzy sets and RSS, as well as the monitoring algorithm based on JNI and web services technology, and the comparison results are shown in Figure 5.

The analysis of the data in Figure 5 shows that, compared with the monitoring algorithm based on fuzzy sets and RSS and the monitoring algorithm based on JNI and web services technology, the data acquisition time of the algorithm in this paper is always lower than 0.3 s, indicating that compared with

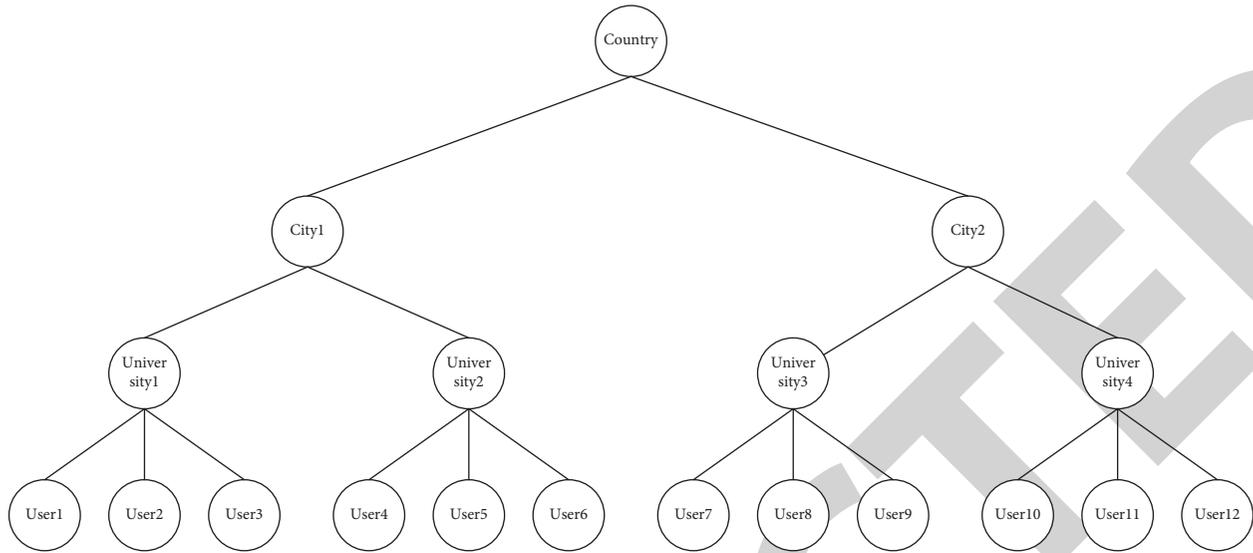


FIGURE 4: Network topology.

TABLE 1: Experimental parameters.

Layer no.	Node storage capability (GB)	Node processing capability per node
1	500	30
2	300	20
3	200	10
4	30	5

TABLE 2: Comparison of signal noise (unit: dB).

Number of experiments	Monitoring algorithm based on fuzzy set and RSS	Monitoring algorithm based on JNI and web service technology	Monitoring algorithm without denoising model	Paper algorithm
10	45	32	21	15
20	52	29	23	12
30	43	35	18	10
40	46	34	13	9
50	38	38	11	10
60	42	31	15	11
70	41	38	24	13
80	47	28	17	10

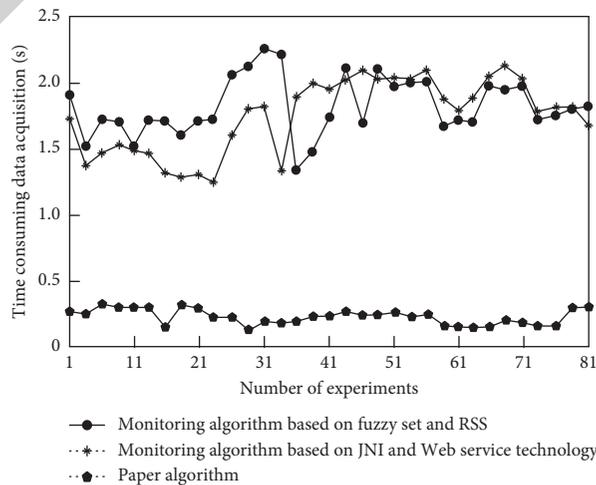


FIGURE 5: Data acquisition time comparison.

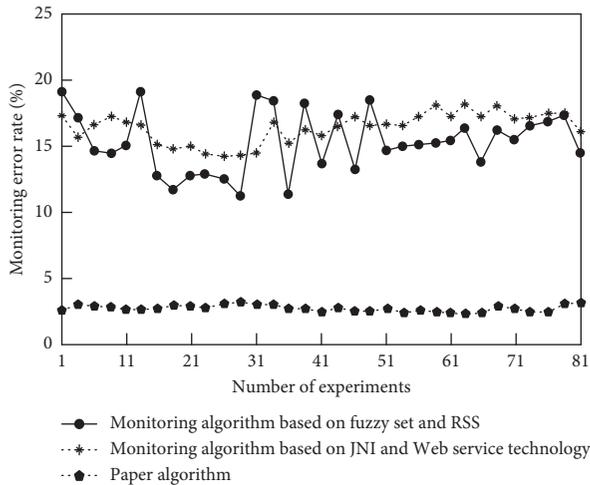


FIGURE 6: Monitoring error rate.

the two algorithms, the data acquisition time of the algorithm in this paper is lower and the collection efficiency is higher.

Finally, the monitoring error rate of the proposed algorithm is compared with the monitoring algorithm based on fuzzy sets and RSS, as well as the monitoring algorithm based on JNI and web services technology, and the results are shown in Figure 6.

By analyzing the data in Figure 6, it can be seen that the monitoring error rate of the fuzzy set and RSS-based monitoring algorithm varies between 12% and 19% and that of the JNI and web services-based monitoring algorithm varies between 14% and 18%. Compared with these two algorithms, the monitoring error rate of the algorithm in this paper always stays below 4%. It shows that the monitoring error rate of this method is lower and the practical application effect is better.

5. Conclusion

Grid is a highly complex distributed computing environment composed of many elements. It requires a geographically distributed heterogeneous system that connects various computing resources through high-speed Internet to jointly solve the problems of large-scale applications. The system can effectively organize and support internal access and realize resource and information sharing. Especially with the development of social economy, especially in the case of changes in economic structure, the height of the power grid has the characteristics of dynamic change, so that its internal resources are in frequent dynamic changes, which can be processed by using the educational resource grid. Therefore, this paper designs an educational resource grid monitoring algorithm based on economic structure transformation and verifies the superiority of the algorithm through experiments, so as to prove that the algorithm has strong practical application value.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] J. E. Warren, "Open educational resources (CLIPP 45)," *Journal of Access Services*, vol. 25, no. 1, p. 1, 2021.
- [2] L. Fischer, O. Belikov, T. K. Ikahihifo, J. Hilton III, D. Wiley, and M. T. Martin, "Academic librarians examination of university students' and faculty's perceptions of open educational resources," *Open Praxis*, vol. 12, no. 3, pp. 399–409, 2020.
- [3] T. Esposito, C. J. Campbell, and L. Wertman, "Optimizing library services-digital darwinism, open educational resources (OERs), and the academic library," *Against the Grain*, vol. 31, no. 2, pp. 1–10, 2021.
- [4] X. S. Tan and X. Y. Wei, "Construction and application of video resource management and publishing system," *Modern Educational Technology*, vol. 30, no. 4, pp. 107–112, 2020.
- [5] J. Pete, F. Mulder, J. Neto, and K. Omollo, "Differentiation in access to, and the use and sharing of (open) educational resources among students and lecturers at technical and comprehensive Ghanaian universities," *Open Praxis*, vol. 10, no. 4, pp. 405–421, 2019.
- [6] Y. N. Wang, "Application of OLAP technology and grid Architecture in educational resource sharing," *Digital World*, vol. 179, no. 9, pp. 278–279, 2020.
- [7] L. W. Chen and Y. M. Wang, "Simulation of emergency scheduling method for strongly correlated attribute resources in computational grid," *Computer Simulation*, vol. 36, no. 9, pp. 385–388, 2019.
- [8] В. В. Баранник, М. Тушиця, and А. Д. Сорокун, "The clustering technology of information resource data on a quantitative attribute," *Science-based Technologies*, vol. 40, no. 4, pp. 1–10, 2019.
- [9] O. Rodriguez-Prieto, A. Mycroft, and F. Ortin, "An efficient and scalable platform for Java source code analysis using overlaid graph representations," *IEEE Access*, vol. 8, no. 1, pp. 72239–72260, 2020.
- [10] T. Nakamaru and S. Chiba, "Generating a generic fluent API in Java," *The Art Science and Engineering of Programming*, vol. 4, no. 3, pp. 25–31, 2020.
- [11] M. Sánchez-Islas, J. C. Toledo-Roy, and A. Frank, "Criticality in a multisignal system using principal component analysis," *Physical Review*, vol. 103, no. 4, pp. 1–13, 2021.
- [12] A. K. Gupta, C. H. Hsu, and C. S. Lai, "Enhancement of the Au/ZnO-na plasmonic SERS signal using principal component analysis as a machine learning approach," *IEEE Photonics Journal*, vol. 12, no. 5, pp. 641–646, 2020.
- [13] A. Kacha, F. Grenez, J. R. O. Arroyave, and J. Schoentgen, "Principal component analysis of the spectrogram of the speech signal: interpretation and application to dysarthric speech," *Computer Speech & Language*, vol. 59, no. 1, pp. 114–122, 2020.
- [14] V. Gupta and M. Mittal, "Arrhythmia detection in ECG signal using fractional wavelet transform with principal component analysis," *Journal of The Institution of Engineers (India) Series B*, vol. 101, no. 6, pp. 1–11, 2020.
- [15] C. Xie and A. Kumar, "Finger vein identification using Convolutional Neural Network and supervised discrete hashing," *Pattern Recognition Letters*, vol. 119, no. 2, pp. 148–156, 2019.

- [16] S. Jeremias, A. Aviad, B. Amir, and E. Michael, "On multi-layer basis pursuit, efficient algorithms and convolutional neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 1968–1980, 2019.
- [17] A. Liu, Z. Zhao, C. Zhang, and Y. Su, "Smooth filtering identification based on convolutional neural networks," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 26851–26865, 2019.
- [18] R. Rojas, *The Backpropagation Algorithm*, Springer, Berlin, Germany, pp. 149–182, 1996.
- [19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

RETRACTED