ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
Received: 13 July 2025; Accepted: 1 October 2026

https://doi.org/10.17559/TV-20250713002830
Original scientific paper

Enhancing Facial Recognition Efficiency with Cloud-Based Parallel Radial Basis Function Networks

Xing YANG*, Xiao Yu ZHAO, Yong Hong ZHANG

Abstract: Facial recognition has high uniqueness and is difficult to forge. Facial recognition is more secure and reliable compared with other identity authentication methods, and can effectively prevent identity theft and fraud. Therefore, to ensure social and network security, a cloud computing Map-Reduce parallel optimized Radial Basis Function (RBF) neural network is built to improve the performance of facial recognition. Firstly, to optimize RBF networks, the K-means++ algorithm is taken to accurately determine the position of the hidden layer center and optimize the network structure. Secondly, to further improve the processing speed and scalability of the facial recognition system, the research also utilizes the Map-Reduce framework in cloud computing to perform parallel optimization on the RBF network. The average facial recognition accuracy was 99.6%, surpassing existing models. The model achieved a minimum recognition precision of 97.7% and an average recognition precision of 98.1%. In terms of recall rate, the lowest was 97.1% and the average was 97.7%, showing excellent performance. In addition, the F1-Score was as low as 0.982 and as high as 0.986 on average, demonstrating its efficiency in facial recognition tasks. The receiver operation characteristic curve was 0.987, further confirming its superior facial recognition ability. The above results indicate that the proposed cloud computing Map-Reduce parallel optimized RBF network has strong application potential in facial recognition, providing valuable references for future research.

Keywords: cloud computing; facial recognition; map-reduce; parallel optimization; radial basis function neural network

1 INTRODUCTION

Driven by computer technology, image processing technology, and machine learning, biometric technology has gradually become a crucial direction in identity authentication and security. Biometric technology utilizes the biometric features, such as fingerprints, irises, and faces, for identity recognition, which is unique, stable, and difficult to forge. Facial recognition, as one of the non-contact, natural, and intuitive biometric methods, has unique advantages and has been largely applied in security monitoring, financial payments, and smartphone unlocking. However, facial recognition technology faces many challenges, such as changes in lighting, facial expressions, posture, and complex backgrounds, which may affect the accuracy and robustness [1, 2]. In addition, with the continuous growth of data volume, how to improve the computational efficiency and scalability of facial recognition algorithms has become crucial. Cloud computing, as an emerging computing model, provides the possibility for large-scale facial recognition tasks with its powerful data processing capabilities and elastic scalability [3, 4]. Map-Reduce, as a distributed computing model, can effectively process and generate big data, providing new ideas for training parallel neural networks. Radial Basis Function (RBF) has shown great potential in the field of facial recognition due to its excellent generalization ability and ability to handle nonlinear problems [5]. However, the traditional RBF network training process often requires a large number of parameter adjustments and has low computational efficiency, making it difficult to meet real-time requirements. Therefore, in response to the above issues, a facial recognition method that integrates cloud computing Map-Reduce and RBF networks has been proposed to optimize the accuracy and efficiency of facial recognition. This method first optimizes the hidden layer center of the RBF network through K-means++, so that it can more accurately reflect the distribution of input data, thereby improving the feature extraction ability of the RBF and enhancing the accuracy of facial recognition. The study also introduces the Map-Reduce framework of cloud computing, which decomposes large-scale datasets into multiple subtasks and processes them in parallel on multiple machines, reducing the computation time of individual tasks and improving the processing speed of facial recognition systems. In addition, due to the powerful elastic scalability of cloud computing platforms, computing resources can be dynamically adjusted according to the needs of tasks. Therefore, as the data volume of facial recognition tasks increases, the processing capacity of the system can be improved by adding computing nodes, thereby ensuring the high-performance operation of the system. The innovation lies in using K-means++ algorithm to optimize hidden layer centers to improve the performance of RBF networks. Secondly, the cloud computing Map-Reduce framework is utilized for parallel optimization of RBF networks. By combining the parallel computing capability of cloud computing with the nonlinear modeling capability of RBF networks, the research aims to significantly improve the processing speed and scalability of facial recognition systems while ensuring recognition accuracy.

2 LITERATURE REVIEW

Facial recognition technology, as a biometric technology based on analyzing and comparing facial visual feature information for identity authentication, is widely used in fields such as public safety, healthcare, and social media. However, the recognition rate of current facial recognition technology is easily affected by factors such as changes in lighting, facial expressions, and posture, leading to a decrease in its recognition accuracy. Therefore, to promote the application of facial recognition technology, it is necessary to develop facial recognition technologies with higher recognition accuracy. Betcy Thanga Shoba and Shatheesh Sam proposed a feature extraction method based on empirical mode decomposition and local binary patterns for facial recognition problems. This method first decomposed the image into intrinsic

mode function images, estimated the intrinsic mode function image pairs using 2D discrete Fourier transform, and extracted intensity compensated local binary mode features. This method had high recognition rate and low time complexity [6]. Although the above method achieves a recognition rate of over 90% and a low time complexity through multi-scale feature decomposition, it relies on the assumption of image stationarity. When dealing with non-stationary facial images in complex backgrounds such as mining areas and construction sites (such as dust occlusion and dynamic blurring), feature decomposition is prone to mode mixing, which limits scalability - it can only adapt to datasets in controllable laboratory environments and is difficult to transfer to actual industrial scenarios. Wang et al. proposed an attention cosine similarity knowledge extraction method for the high-precision real-time facial recognition model on mobile and embedded devices. This method combined attention mechanism and knowledge extraction technology, and transmitted the attention map of the teacher network to the student network through an innovative cross stage connection review path. This method could significantly improve the real-time facial recognition accuracy of mobile and embedded devices [7]. However, the above methods overly rely on large-scale annotated high-quality facial datasets, have insufficient generalization ability for small sample datasets, and do not consider the impact of hardware performance differences on model deployment on different mobile devices. The scalability is weak on edge devices with limited hardware resources. Cao et al. proposed a lightweight semantic transformer network method to address the semantic defects caused by small sample sizes and cross modal differences in sketch facial recognition. This method first used meta learning training strategy to solve the small sample problem, and then constructed a lightweight semantic transformation network using hierarchical linear transformation and parameter sharing. The recognition rate of this method reached 92.59% [8]. However, the above methods only focus on cross modal matching of "sketch photo", and the diversity of the dataset is limited to static sketches and standard photos, without involving dynamic faces (or face data under extreme lighting). Moreover, when processing high-resolution sketches, the calculation time of the semantic conversion module of the model significantly increases, making it difficult to balance robustness and efficiency. Rajpal et al. built a facial recognition method relying on extreme learning machines and their online sequence variants to recognize unknown face samples in an uncontrolled environment. This method utilized the Viola-Jones object detection framework to track the facial features of sample faces, extracted facial features through gradient oriented histograms, and then input these features into an extreme learning machine classifier. The results demonstrated that the method exhibited high operational speed, low computational time complexity, and was applicable in real-time facial recognition domains requiring high-accuracy outcomes [9]. Although the above method achieves fast face detection through the Viola Jones framework, its feature extraction relies on gradient direction histograms, which are sensitive to feature deformation caused by facial expression changes. The recall rate decreases by about 5-8% on datasets with rich facial expressions, and the weight random initialization characteristic of extreme learning machines leads to insufficient stability of the model on large-scale datasets, with scalability limited by the order of magnitude of training samples.

As a feedforward artificial neural network, RBF network has strong nonlinear mapping ability and can approximate any complex nonlinear function. Heidari et al. built a detection model relying on blockchain and RBF network to address the balance between detection accuracy and efficiency faced by intrusion detection systems. This model enhanced data integrity and storage capabilities through blockchain technology, supported intelligent decision-making between IoT networks, and effectively applied and shared deep learning methods in a decentralized manner. The results showed that the model had better specificity, F1-Score, recall rate, accuracy, and precision than existing cutting-edge methods [10]. Although the above method improves data integrity, the center of the hidden layer of the RBF network in this model is determined through random selection, without considering the category discrimination of features. This can lead to "center redundancy" when processing high-dimensional facial features, resulting in redundant model parameters and reduced computational efficiency, making it difficult to adapt to the real-time requirements of facial recognition. Prasanna et al. proposed a new method for energy aware resource allocation, which combined RBF network with successive concave approximation depth expansion. This method utilized RBF to approximate the relationship between resource allocation decisions and network performance indicators. A successive concave approximation framework was adopted for learning through progressive training and stochastic gradient descent. This method could be applied to various network topologies with different sizes, densities, and channel dispersion, and improve the convergence and quality of solutions in resource allocation [11]. Although the above method optimizes convergence through successive concave approximations, the RBF kernel function parameters of this method rely on manual tuning, resulting in poor parameter adaptability and insufficient robustness when the facial feature dimension dynamically changes. Rubio et al. proposed a genetic optimizer to address the step size selection in the optimization process of RBF networks. This optimizer was based on simplex optimizer and bat optimizer, which optimized the performance of RBF network by seeking an acceptable step size in gradient strategy. The genetic optimizer could improve the performance of RBF networks [12]. Among the above methods, the iterative process of genetic algorithm takes a long time. In large-scale face data training, the convergence time of the model increases by about 30% compared to the parallel optimization method in this study, and its scalability is limited. Dayev proposed a modeling method using RBF network for the flow coefficient of differential pressure flowmeter. This method calculated the flow coefficient of the orifice plate using the angular pressure tapping method. A corresponding RBF network structure was developed to approximate this coefficient. RBF network could effectively approximate the flow coefficient [13]. Although the above method achieves accurate approximation, the RBF network of this model is a single node serial architecture, and when processing data

exceeding 100000, the computation time increases exponentially, which cannot meet the engineering requirements of "real-time retrieval of millions of face databases" in face recognition.

In summary, there are many studies on facial recognition, but the high-performance performance of current mainstream facial recognition models highly relies on "large-scale annotated training samples", and their generalization ability significantly decreases in scenarios with limited sample sizes. Meanwhile, with the development of facial recognition applications towards large-scale and high concurrency directions (such as airport security checks and urban monitoring), the computational efficiency shortcomings of existing models are becoming increasingly prominent. Therefore, to improve the accuracy and speed of facial recognition and achieve small sample learning, a facial recognition model combining cloud computing Map-Reduce and RBF network is proposed.

3 RESEARCH METHODOLOGY

To improve the accuracy of facial recognition and accurately analyze facial expressions, a facial recognition method relying on RBF is built. This method uses K-means++ to select the center of the hidden layer and utilizes cloud computing Map-Reduce for parallel optimization.

3.1 Construction of Facial Recognition Model Based on RBF Network

Facial recognition, as a non-contact recognition method, has had a profound impact in fields such as security, convenience, artificial intelligence, social services, commerce, healthcare, and education. Facial recognition, as a complex nonlinear problem, is susceptible to various factors such as lighting, facial expressions, posture, etc. RBF networks can use RBFs as activation functions for hidden layers, thereby modeling complex nonlinear relationships [14, 15]. RBF network can map the input facial features to a high-dimensional space, making the originally linearly inseparable categories linearly separable. This enables the RBF network to have strong classification ability when dealing with complex changes in facial images, and can more accurately recognize different faces [16]. The RBF is displayed in Eq. (1).

$$\phi(X, C) = \phi(\| X - C \|) = \phi\left(\sum_{i=1}^{n} (x_i - c_i)^2 / 2 \right) \quad (1)$$

In Eq. (1), $\varphi(.)$ represents the RBF. x_i represents the input vector. c_i signifies the center point. As the core activation function of the RBF neural network's hidden layer, the essence of the above equation is to quantify the similarity between the input facial feature vector and the center of the hidden layer by calculating the distance between the two. Fig. 1 displays the RBF network.

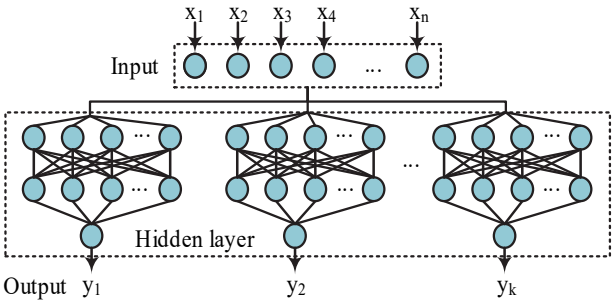


Figure 1 Structure of RBF network

In Fig. 1, the RBF is a feedforward neural network. The hidden layer is the core part of the RBF network, which is composed of multiple RBF units. Each hidden layer neuron corresponds to a RBF, and its output value depends on the distance between the input vector and the center of the neuron [17, 18]. The output of the hidden layer neuron is displayed in Eq. (2).

$$h_i(x) = \phi(\| x - c_i \|) \quad (2)$$

In Eq. (2), $h_i(x)$ represents the output of the *i*-th hidden layer neuron. Common RBFs include Gaussian functions, multiple quadratic functions, inverse multiple quadratic functions, etc. To better adapt to different sizes of faces and expression changes, the RBF function selected for the study is a Gaussian function, as shown in Eq. (3).

$$\phi(r) = \exp\left(-\frac{r^2}{2\sigma_i^2} \right) \quad (3)$$

In Eq. (3), *r* represents the Euclidean distance between the input vector and the center point. The output layer is shown in Eq. (4).

$$y_k = \sum_{i=1}^{M} w_{ki} h_i(x) + b_k \quad (4)$$

In Eq. (4), y_k represents the output of the *k*-th output layer neuron. *M* represents the number of hidden neurons. w_{ki} represents the connection weight between the *k*-th output layer neuron and the *i*-th hidden layer neuron. b_k represents the bias of the *k*-th output layer neuron. The training process is as follows. There are several training samples, and these samples do not belong to the same category. At this point, the first step is to use an RBF network to find a function that satisfies Eq. (5).

$$f(X) = y_i, i = 1, 2, ..., p \quad (5)$$

In Eq. (5), $f(X)$ represents the expression surface. *p* represents the number of sample categories. The core goal of network training, as clarified through this equation, is to ensure that the "expression surface" output by the model accurately covers all training sample categories. Since the above equation must pass through all training samples, the technical function of the RBF function is presented in Eq. (6).

$$f(X)=\sum_{i=1}^{p} w_i \varphi(\| x-x_i \|) \tag{6}$$

In Eq. (6), w_i represents the weight. $\phi(\|x-x_i\|)$ represents the set of RBF functions. Eq. (6) decomposes the network output into a weighted combination of multiple RBF functions, essentially decomposing the complex face classification problem into multiple "local feature matching subproblems". The above equation is expanded, as shown in Eq. (7).

$$\begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1p} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_{p1} & \varphi_{p2} & \cdots & \varphi_{pp} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_p \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix} \tag{7}$$

In Eq. (7), ϕ_{pp} represents the elements of the set of RBF functions. w_p represents the network connection weight. y_p represents the expected output. In practical applications, if the number of training samples is too large, it may lead to overfitting. To solve this problem, it is necessary to use regularization theory to optimize the generalization ability by minimizing the Tikhonov functional. The Tikhonov functional is presented in Eq. (8).

$$\delta(f)=\delta_B(f)+\lambda\delta_c(f) \tag{8}$$

In Eq. (8), $\delta(f)$ represents the Tikhonov functional. $\delta_B(f)$ represents the standard error. λ represents the regularization parameter. $\delta_C(f)$ represents the regularization term. To address the issue of overfitting caused by too many training samples in facial recognition, a regularization term is introduced to limit excessive fluctuations in weight parameters. The solution to the regularization problem can be obtained through the above equation. If calculating the weights, the inverse of the matrix must be obtained first. However, considering its high computational complexity to reduce the computational complexity, it can be achieved by solving the approximate value of the regularization solution [19]. The regularization problem is solved, as shown in Eq. (9).

$$\begin{cases} f(X)=\sum_{i=1}^{p} w_i G(x, x_i) \\ w=(G+\lambda I)^{-1} y \end{cases} \tag{9}$$

In Eq. (9), $G(.)$ represents the Green function of the self-adjoint differential operator. I represents the identity matrix. To implement RBF network, considering the adaptability and robustness of the model, the chosen implementation method is self-organizing selection center. The RBF network process based on self-organizing selection center is shown in Fig. 2.

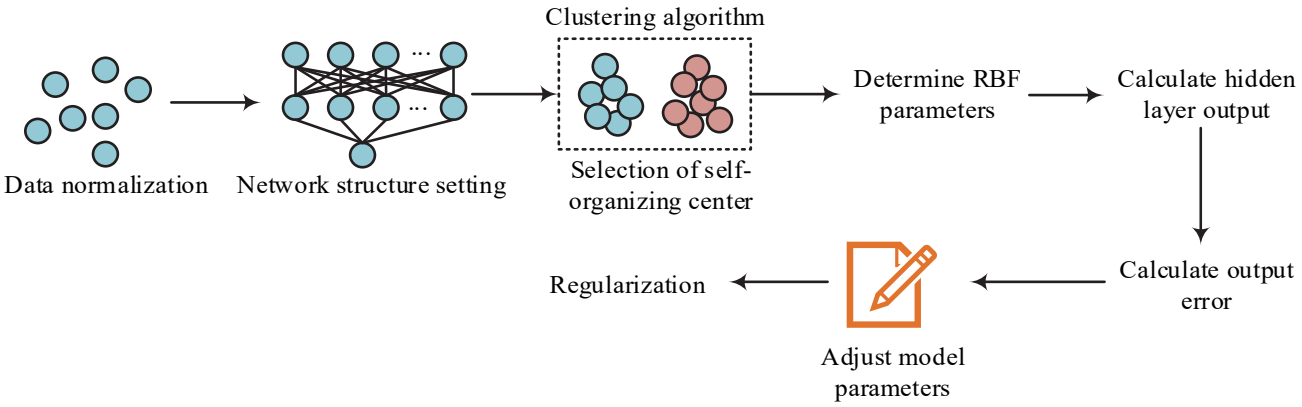


Figure 2 RBF network process based on self-organizing selection center

In Fig. 2, the model first needs to normalize the input data and initialize the RBF function. Next, the training data is clustered using clustering algorithms to determine the center of the RBF network's hidden layer. Then, the center of each RBF is determined based on the clustering results. Next, the output of the hidden layer is calculated, which is the activation value of each RBF neuron. Finally, the model parameters are adjusted, and the regularization technique is used to prevent the model from over fitting. Based on the above method, facial recognition can be achieved through an RBF network based on self-organizing selection centers.

3.2 RBF Network Facial Recognition Model Based on Cloud Computing Map-Reduce Parallel Optimization

Although facial recognition can be achieved through the above method, the RBF network requires a large amount of parameter tuning, resulting in low learning efficiency and training accuracy. To improve the above problems, a facial recognition model relying on cloud computing Map-Reduce parallel optimized RBF network is proposed. The RBF network relying on self-organizing selection center mentioned above needs to determine the center of the hidden layer through clustering algorithm. Considering the quality and speed of clustering, the chosen clustering algorithm for the study is K-means++, as presented in Fig. 3.

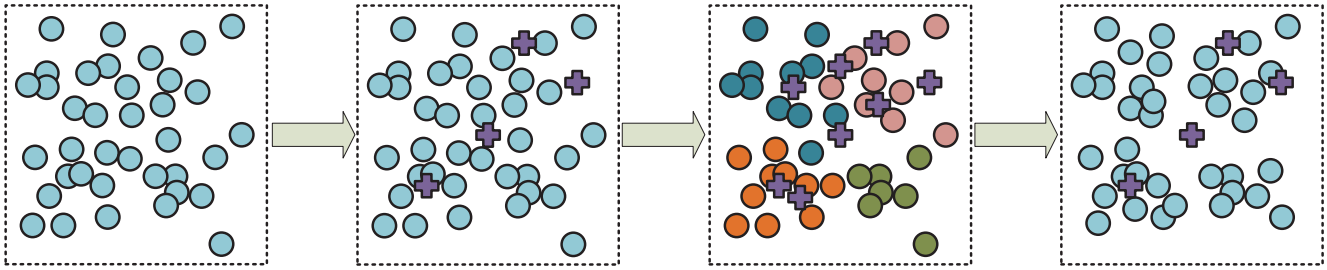


Figure 3 K-means++ algorithm

In Fig. 3, the core principle of K-means++is to select initial cluster centers through an iterative method to ensure that these centers are more evenly distributed in the data space, thereby reducing the risk of the algorithm getting stuck in local optima. Firstly, a point is randomly selected from the dataset as the first center, and the nearest distance from each point in the dataset to that cluster center is calculated to measure the proximity of the sample point to the current cluster center [20, 21]. Finally, a new point is taken as the cluster center. The process is repeated until all cluster centers are selected. The probability of selecting a sample point as the cluster center is presented in Eq. (10).

$$P(x)=\frac{D(x)^2}{\sum_{y\in X} D(y)^2} \tag{10}$$

In Eq. (10), $P(x)$ signifies the probability that the sample point is taken as the cluster center. $D(x)$ represents the distance from the sample point to the cluster center. S represents the sample set. The above formula can clarify the rule of "selecting the initial center based on distance probability" in the K-means++algorithm, that is, the farther the sample is from the existing center, the higher the probability of being selected as the new center. When selecting the initial center, the Map-Reduce is introduced. To improve the quality of clustering, the study adopts the initial center selection method as the hierarchical selection method. Firstly, the distance between each sample and the cluster center is calculated and accumulated to obtain the total distance. Then, a distance is randomly selected and the corresponding sample is determined as the new center. This process is repeated until all centers are selected [22]. There exists a sample set that satisfies Eq. (11).

$$M=M_1+M_2+\ \ldots\ +M_n \tag{11}$$

In Eq. (11), M represents the total sample set. M_n represents sub sample sets. n represents the number of Maps. At this point, the distance between each sample in the sample subset is shown in Eq. (12).

$$d_{\text{sum}}=\sum_{i=1}^{M_n} d_i \tag{12}$$

In Eq. (12), d_{sum} represents the distance sum between each sample in the subset. d_i represents the distance of sample i. At this point, the probability of the sample point being the candidate point is presented in Eq. (13).

$$p_i=\frac{d_i}{d_{\text{sum}}} \tag{13}$$

In Eq. (13), p_i signifies the probability of the sample point being the candidate point. The probability of selecting a sample point in the next round of the election is shown in Eq. (14).

$$p_m=\frac{d_{\text{sum}}}{\sum_{k=1}^{M} d_{\text{sum}}} \tag{14}$$

In Eq. (14), p_m signifies the probability of the sample point being selected in the next round of elections. At this point, the probability of the sample point being the initial cluster center is shown in Eq. (15).

$$p=p_i\cdot p_m=\frac{d_i}{d_{\text{sum}}}\cdot\frac{d_{\text{sum}}}{\sum_{k=1}^{M} d_{\text{sum}}}=\frac{d_i}{\sum_{k=1}^{M} d_{\text{sum}}} \tag{15}$$

In Eq. (15), p signifies the probability that the sample point is the initial cluster center. The implementation of the Map function is shown in Fig. 4.

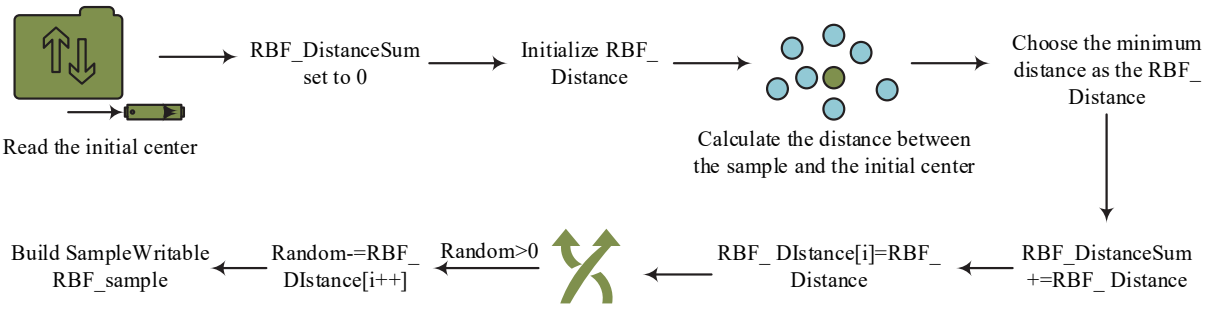


Figure 4 Implementation of the map function

In Fig. 4, the initial center is first read from the Hadoop Distributed File System (HDFS). Next, the system initializes two important data structures. One is the RBF distance sum used to record the sum of the distances from each data point to its nearest center point, with an initial value of 0. The other is the RBF distance array, which stores the specific distance values from each data point to its corresponding nearest center point. Subsequently, the algorithm traverses all training samples and calculates the distance between each sample point and all initial center points. Among these distances, the smallest one is selected as the RBF distance of the sample point and recorded in the RBF distance array. Meanwhile, this minimum distance is added to the sum of RBF distances to update the value of the sum. After completing the distance calculation and sum update of all sample points, the algorithm randomly generates a value within the range of 0 to the sum of RBF distances. If the random value is greater than 0, the algorithm will perform an index increment operation on the RBF distance array [23, 24]. Finally, based on the above results, a sample writable RBF sample can be constructed. The implementation of the Reduce function is shown in Fig. 5.

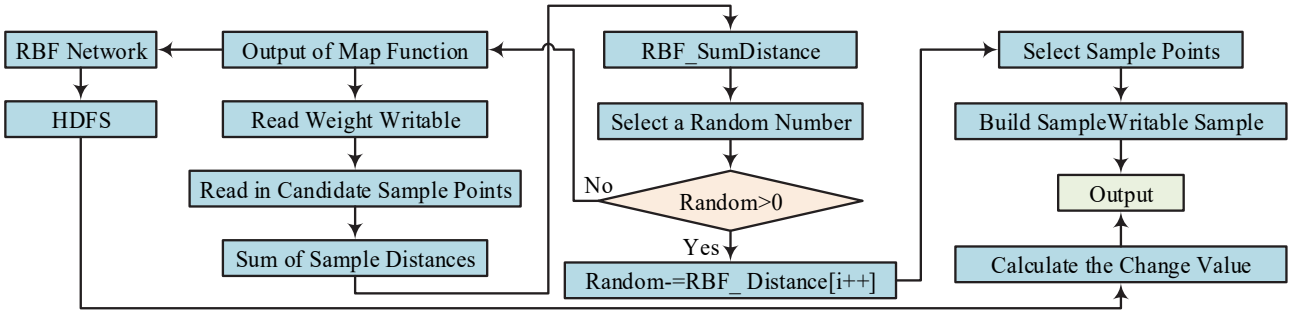


Figure 5 The implementation of the Reduce function

In Fig. 5, firstly, all output results of the Map function are summarized and processed. The distance from each sample point to its nearest cluster center is calculated and accumulated to obtain the total RBF distance. A value is randomly generated, which will be used to determine which sample point is the cluster center. Specifically, if the generated random number is greater than 0, the algorithm will determine an index based on the values in the RBF distance array and add 1 to the index value to obtain a new random number. Finally, relying on the data of the selected sample points, the algorithm will construct a SampleWritable object, which contains all the relevant information of the sample points, such as their eigenvalues and corresponding clustering labels [25, 26]. To implement K-means++ relying on Map-Reduce parallel optimization, the class RBF_SampleSumWritable is defined to implement the Writable interface. To reduce communication volume, the Combine function is adopted to sum the intermediate results and the processed data is taken as input to the RBF_SampleSumWritable to adjust the cluster centers. Finally, the implementation of the Drive function is shown in Fig. 6.

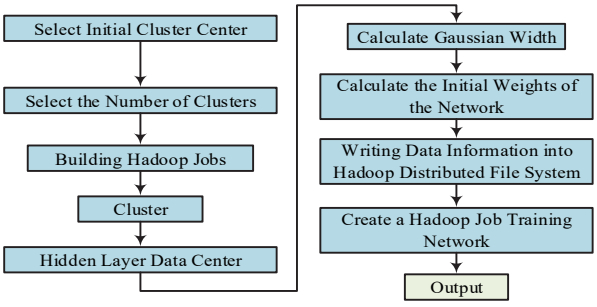


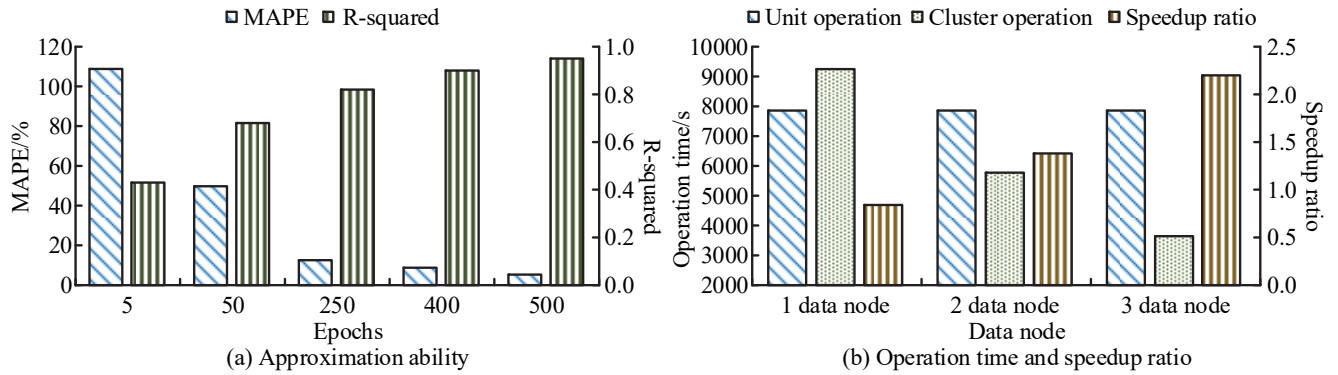
Figure 6 Implementation of the Drive function

In Fig. 6, it is necessary to carefully select one or more initial clustering centers from the sample data first. Next, based on prior knowledge and experience, the appropriate number of clusters is determined. Subsequently, multiple jobs are constructed using the Hadoop framework to perform parallel clustering operations on the dataset to obtain the initial data center. After obtaining preliminary clustering results, it is necessary to calculate the Gaussian width of each cluster center. Meanwhile, the initial weights of the network are determined and updated to optimize model performance. Afterwards, the new data information generated during the training process is written into HDFS, and Hadoop jobs are created and run again to further train the network until the model converges, obtaining the final clustering and network training results. A cloud computing Map-Reduce parallel optimized RBF network facial recognition model is constructed through the above method.

4 RESULTS AND DISCUSSION

4.1 Test Results of Approximation Ability and Speedup Ratio

To explore the approximation ability and speedup ratio of cloud computing Map-Reduce parallel optimization RBF network, the Hermit function and Carnegie Mellon University Pose, Illumination, Expression (CMU-PIE) dataset are used to test it. The value range of the Hermit function is [−4, 4], with a sample size of 100. The CMU-PIE dataset contains 41368 facial images, covering 68 categories.In the experiment, the number of cluster centers for K-means++is 50, and the distance threshold is 10^{-4}. The implementation of the Map Reduce parallel framework is based on Apache Hadoop version 3.3.4. The Hadoop cluster adopts an architecture of "1 Master node+3 DataNode nodes", and all nodes are deployed on the same local area network (1000 Mbps Ethernet connection). The approximation ability and speedup ratio of the model are shown in Fig. 7.



(a) Approximation ability (b) Operation time and speedup ratio

Figure 7 Approximation ability and speedup ratio of the model

In Fig. 7a, with the increase of training epochs, the Mean Absolute Percentage Error (MAPE) of the model gradually decreased, while *R*-squared gradually increased. When the number of training epochs was 5, the MAPE and *R*-squared of the model were 108.8% and 0.43. When the epoch was 500, the MAPE and *R*-squared of the model were 5.3% and 0.95. In Fig. 7b, with the increase of data nodes, the cluster operation time gradually decreased, while the speedup ratio gradually increased. When the number of data nodes was 1, the cluster operation time and speedup ratio were 9248 s and 0.84, respectively. When the number of data nodes was 3, the cluster speedup ratio time and speedup ratio were 3641s and 2.2, respectively. It can be seen that within a reasonable range of node numbers (1-3), increasing parallel nodes can significantly shorten the running time, and due to the distributed data processing characteristics of the Map Reduce framework (such as local computation+global aggregation), accuracy is guaranteed. However, we need to be cautious of the risk of "node redundancy": if we further increase the number of nodes (such as more than 3), although theoretically it can continue to shorten the time, it may lead to an increase in communication overhead between nodes. The above results indicate that after cloud computing Map-Reduce parallel optimization, the RBF network has good approximation ability and high acceleration ratio. The results of cluster convergence verification are shown in Tab. 1.

Table 1 Cluster convergence verification results

Epoch	Within-Cluster Sum of Squares	Between-Cluster Sum of Squares/Within-Cluster Sum of Squares	Center Displacement Distance
1	128.6	3.2	0.021
20	19.3	14.8	0.003
28	18.5	15.3	0.001

According to Tab. 1, as Epoch increases, the Within Cluster Sum of Squares and Center Displacement Distance gradually decrease, ultimately dropping to 18.5 and 0.001, respectively; The Between Cluster Sum of Squares/Within Cluster Sum of Squares gradually increased and eventually stabilized at 15.3. The above results indicate that the converged clustering results have good consistency and generalization, which can provide high-quality hidden layer centers for RBF networks and avoid network training bias caused by unstable clustering.

4.2 Facial Recognition Test Results

To verify the facial recognition performance of the proposed cloud computing Map-Reduce parallel optimized RBF network, it was tested and compared with the Light Illuminated Surface Feature Separation Network (LFSepNet) and Improved MultiScale Visual Transformer (IMViT) models. The dataset used in the experiment is the Labeled Faces in the Wild (LFW) dataset, which contains 13233 facial images and covers 5749 categories. The image is an 8-bit grayscale image with a uniform size of 250 × 250 pixels. The CPU used in the experiment is Intel core i5 6700K, with 8 GB of memory and Windows 10 operating system. The recognition accuracy and precision of different models are shown in Fig. 8.

According to Fig. 8a, the recognition accuracy of LFSepNet and IMViT was the highest at 96.2% and 97.2%, respectively, the lowest at 95.3% and 96.2%, respectively, and the average accuracy was 95.8% and 96.7%, respectively. The facial recognition model proposed in the study had a minimum recognition accuracy of 99.2% and an average accuracy of 99.6%. According to Fig. 8b, the recognition precision of LFSepNet and IMViT was the highest at 94.2% and 95.2%, respectively, the lowest at 93.1% and 94.2%, respectively, and the average precision was 93.6% and 94.7%, respectively. The facial recognition model proposed in the study had a minimum recognition precision of 97.7% and an average precision of up to 98.1%. According to Fig. 8c, the recall rates of LFSepNet and IMViT were the highest at 93.8% and 94.7%, and the lowest at 92.8% and 93.1%, respectively. The average recall rates were 93.2% and 93.9%, respectively. The lowest recall rate of the facial recognition model proposed in the study was 97.1%, with an average recall rate of 97.7%. From Fig. 8d, the F1-Score of LFSepNet and IMViT was the highest at 0.927 and 0.941, and the lowest at 0.918 and 0.931. The average F1-Score was 0.923 and 0.936. The F1-Score of the proposed facial recognition model was as low as 0.982, with an average F1-Score of up to 0.986. The above results indicate that the proposed cloud computing Map-Reduce parallel optimized RBF network has higher facial recognition accuracy. The number of misrecognitions and recognition errors of different models are shown in Fig. 9.

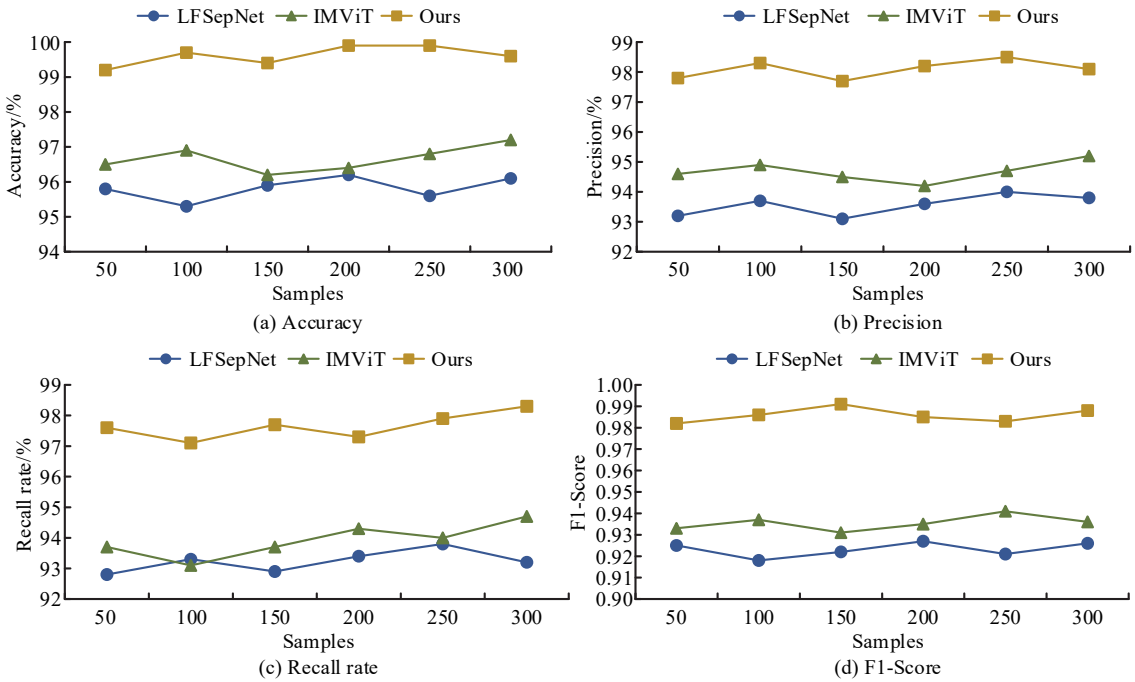


Figure 8 Recognition accuracy and precision of different models

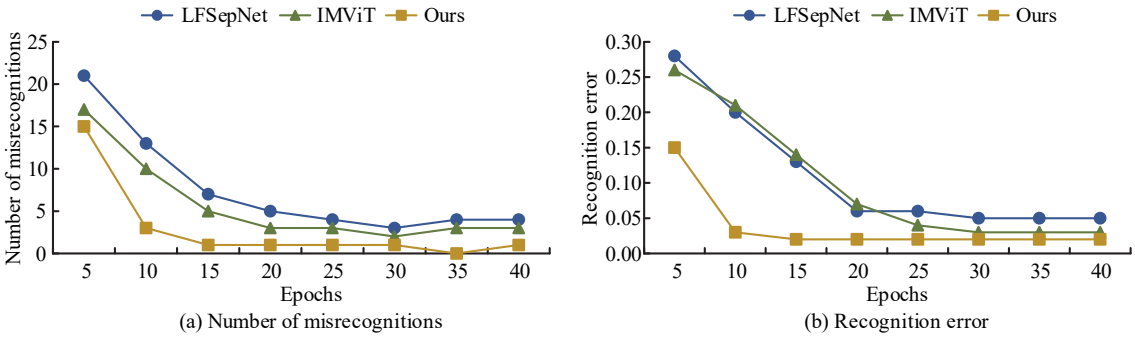


Figure 9 The number of misrecognitions and the recognition error of different models

From Fig. 9a, the number of misrecognitions for LFSepNet and IMViT tended to stabilize after 20 and 20 epochs, respectively. At this point, the number of misrecognitions for LFSepNet and IMViT was 4 and 3, respectively. The number of misrecognitions in the facial recognition model proposed by the study tended to stabilize after 10 epochs. At this point, the number of misrecognitions did not exceed 1. As shown in Fig. 9b, the recognition errors of LFSepNet and IMViT tended to stabilize after 30 and 25 epochs, respectively. At this point, the recognition errors of the two were 0.05 and 0.03, respectively. The recognition error of the proposed facial recognition model tended to stabilize after 10 epochs. At this point, the recognition error of the model was only 0.02. The above results indicate that the proposed cloud computing Map-Reduce parallel optimized RBF network has better facial recognition performance and faster training speed. The Receiver Operating Characteristic Curve (ROC) and recognition speed of different models are shown in Fig. 10.

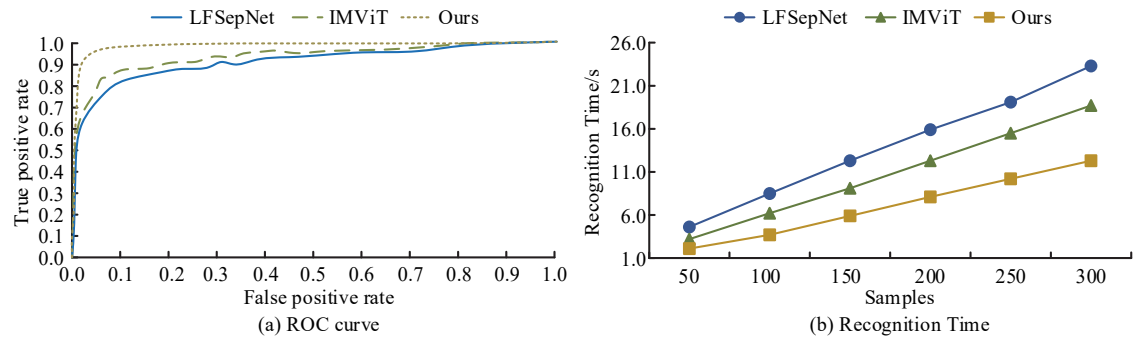


Figure 10 ROC curves and recognition speed of different models

According to Fig. 10a, the ROC curve of LFSepNet and IMViT was 0.923 and 0.936, respectively, while the proposed parallel optimized RBF network had an area under the ROC curve of 0.987, which was higher than other

algorithms. Compared with other models, the proposed facial recognition model has better performance. As shown in Fig. 10b, compared with other models, the proposed cloud computing Map-Reduce parallel optimized RBF network had a faster recognition speed. When the sample size was 150, the recognition time of LFSepNet, IMViT, and the proposed model was 12.3 s, 9.1 s, and 5.9 s, respectively. The recognition time of the proposed method was reduced by more than 35%. The recognition precision and F1-Score of different models for facial expressions are shown in Fig. 11.

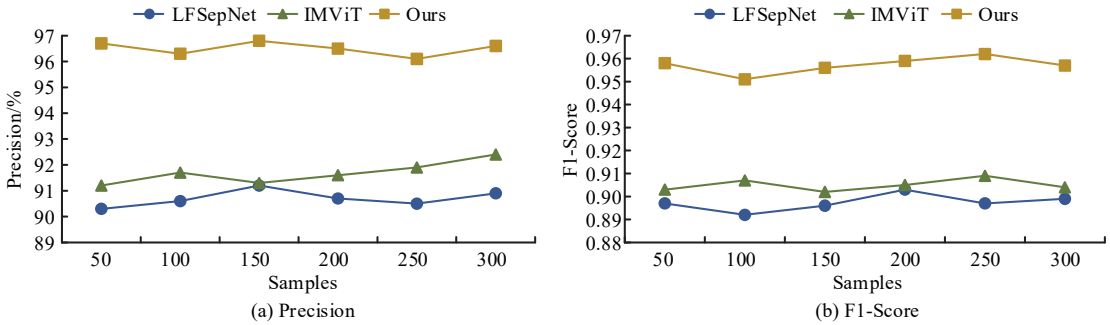


Figure 11 Recognition precision and F1-Score of different models for facial expression

According to Fig. 11a, LFSepNet and IMViT had the highest recognition precision of 91.3% and 92.4% for facial expressions, and the lowest recognition precision of 90.3% and 91.2%, respectively. The average recognition precision was 90.8% and 91.7%, respectively. The proposed parallel optimized RBF network had a minimum precision of 96.1% and an average precision of 96.5% for facial expression recognition, which was much higher than other algorithms. As shown in Fig. 11b, in terms of facial expression recognition, the F1-Score of LFSepNet and IMViT were the highest at 0.903 and 0.909, the lowest at 0.892 and 0.902, and the average F1-Score was 0.897 and 0.905, respectively. The F1-Score of the proposed facial recognition model was as low as 0.951, with an average F1-Score of up to 0.957. The above results indicate that the proposed parallel optimized RBF network can not only achieve accurate facial recognition, but also accurately recognize facial expressions. The significance test results are shown in Tab. 2.

Table 2 Results of significance test

Index	LFSepNet	IMViT	Ours	*p*
Average accuracy	95.8%	96.7%	99.2%	< 0.001
Average Precision	93.6%	94.2%	97.7%	< 0.001
Average recall	93.2%	93.9%	97.7%	< 0.001
Average F1-Score	0.923	0.936	0.982	< 0.001
AUC	0.923	0.936	0.987	< 0.001

According to Tab. 2, the significance test results show significant differences in accuracy, precision, recall, F1 Score, and AUC between the proposed method and other methods. To further analyze the RBF network based on Map-Reduce parallel optimization, ablation experiments were conducted, as presented in Tab. 3.

Table 3 Results of ablation experiments

Module	K-means++	Map-Reduce	Cloud computing	Precision/%	F1-Score
A	×	×	×	85.3	0.845
B	√	×	×	88.2	0.878
C	×	√	×	93.6	0.942
D	×	×	√	87.8	0.869
E	√	√	×	92.6	0.903
F	√	×	√	93.3	0.915
G	×	√	√	94.1	0.928
H	√	√	√	98.1	0.986

According to Tab. 3, the facial recognition precision and F1-Score of the simple RBF network were 85.3% and 0.845. After introducing K-means++, Map-Reduce, and cloud computing, the precision and F1-Score increased to 98.1% and 0.986, respectively, with an increase of 15.0% and 16.7%. The above results indicate that K-means++ and Map-Reduce can effectively improve the performance of RBF networks and achieve accurate facial recognition.

5 CONCLUSION

To improve the accuracy and speed of facial recognition, a facial recognition model based on cloud computing Map-Reduce parallel optimized RBF network is proposed. This model utilized K-means++ to implement the center selection of hidden layers and utilizes cloud computing Map-Reduce for parallel optimization. The facial recognition accuracy reached 99.6%, with an average recognition accuracy of up to 98.1%, significantly better than the LFSepNet and IMViT models. The recall rate and F1-Score were 97.7% and 0.986, which were also superior to other models, further proving its efficiency and accuracy in facial recognition tasks. Through ROC curve analysis, the AUC value of the model reached 0.987, demonstrating its superiority in classification performance. The ablation experiment results further validated the effectiveness of K-means++clustering algorithm and Map-Reduce parallel optimization strategy in improving

the performance of RBF network, resulting in a 15.0% and 16.7% improvement in facial recognition precision and F1-Score, respectively. The above results indicate that the proposed cloud computing Map-Reduce parallel optimized RBF network not only achieves significant improvements in facial recognition accuracy, but also in processing speed, providing new ideas and methods for the development of facial recognition technology. However, the cloud computing platform used in the study is Hadoop, which consumes a lot of resources and has poor security, making it prone to information leakage. Therefore, in the future, symmetric encryption algorithms will be used to improve the privacy and security of the model.

Acknowledgements

This work was supported by the 2025 Sichuan Provincial Education Digitalization Research Project (No. 2025LXKTPS393) and the 2025 Research Project on Computer Basic Education Teaching of the Association of Fundamental Computing Education in Chinese Universities (No. 2025-AFCEC-454).

6 REFERENCES

[1] Teng, L., Du, L., Leng, Z., & Wang, X. (2024). Chaotic image encryption based on partial face recognition and DNA diffusion. *Applied Intelligence*, *54*(21), 10360-10373. https://doi.org/10.1007/s10489-024-05613-9

[2] Wu, J. L., Chen, S. W., Lee, C. E., & Tien, C. H. (2025). Under-display face-recognition system with neural network-based feature extraction from lensless encrypted images. *Applied Optics*, *64*(3), 567-577. https://doi.org/10.1364/AO.534177

[3] Farouk, E. R., Abd-Elnaby, M., Ashiba, H. I., El-Banby, G. M., El-Shafai, W. et al. (2024). Secure cancelable face recognition system based on inverse filter. *Journal of Optics, 53*(3), 1667-1688. https://doi.org/10.1007/s12596-023-01233-7

[4] Yan, L., Zhang, Y., & Zhang, Y. (2023). A fast face recognition system based on annealing algorithm to optimize operator parameters. *The Imaging Science Journal*, *71*(3), 323-330. https://doi.org/10.1080/13682199.2023.2182261

[5] Yang, X., Xu, L., Pang, T., Dong, Y., Wang, Y., Su, H., & Zhu, J. (2025). Face3dadv: exploiting robust adversarial 3d patches on physical face recognition. *International Journal of Computer Vision*, *133*(1), 353-371. https://doi.org/10.1007/s11263-024-02177-6

[6] Betcy Thanga Shoba, V. Shatheesh Sam, I. (2023). Empirical mode decomposition and local binary patternbased feature extraction for face recognition. *The Imaging Science Journal*, *72*(6), 791-807. https://doi.org/10.1080/13682199.2023.2226892

[7] Wang, Z., Zhao, S. W., & Guo, W. Y. (2024). Knowledge distillation of face recognition via attention cosine similarity review. *IET Computer Vision*, *18*(7), 875-887. https://doi.org/10.1049/cvi2.12288

[8] Cao, L., Yin, J., & Zhang, D. F. (2023). Sketch face recognition based on light semantic transformer network. *IET Computer Vision*, *17*(8), 962-976. https://doi.org/10.1049/cvi2.12209

[9] Rajpal, A., Sehra, K., Mishra, A., & Chetty, G. (2023). A low-resolution real-time face recognition using extreme learning machine and its variants. *The Imaging Science Journal*, *71*(5), 456-471. https://doi.org/10.1080/13682199.2023.2183544

[10] Heidari, A., Jafari Navimipour, N., & Unal, M. (2023). A Secure Intrusion Detection Platform Using Blockchain and Radial Basis Function Neural Networks for Internet of Drones. *IEEE Internet of Things Journal*, *10*(10), 8445-8454. https://doi.org/10.1109/JIOT.2023.3237661

[11] Prasanna, B. T., Ramya, D., Shelke, N., Fernandes, J. B., Galety, M. G., & Ashok, M. (2023). Radial basis function neural network-based algorithm unfolding for energy-aware resource allocation in wireless networks. *Wireless Networks*, *30*(8), 7041-7058. https://doi.org/10.1007/s11276-023-03540-0

[12] Rubio, J. D. J., Islas, M. A., Garcia, D., Pacheco, J., Zacarias, A., & Aguilar-Ibaez, C. (2024). Optimized radial basis function network for the fatigue driving modeling. *Journal of Supercomputing*, *80*(7), 8719-8741. https://doi.org/10.1007/s11227-023-05775-2

[13] Dayev, Z. A. (2024). Modeling of the discharge coefficient of differential pressure flowmeters: approximation by radial basis function networks. *Measurement Techniques*, *67*(9), 668-675. https://doi.org/10.1007/s11018-025-02387-5

[14] Zhu, P. (2024). Retracted article: application of rbf network structure and data mining in e-commerce network marketing. *Soft Computing*, *28*(2), 715-715. https://doi.org/10.1007/s00500-023-08791-9

[15] Gorbachenko, V. I., & Stenkin, D. A. (2023). Physics-informed radial basis-function networks. *Technical physics*, *68*(8), 151-157. https://doi.org/10.1134/S1063784223050018

[16] Kumar, H. & Saxena, V. . (2024). Effective test cases generation with harmony search and rbf neural network. *International Arab Journal of Information Technology (IAJIT)*, *21*(5) 786-799. https://doi.org/10.34028/iajit/21/5/2

[17] Qian, C., Hua, C., Zhang, L., & Ahn, C. K. (2024). Modeling and adaptive vibration control for variable-length strip rolling systems with actuator faults and output constraints. *Nonlinear Dynamics*, *112*(2), 995-1009. https://doi.org/10.1007/s11071-023-08769-0

[18] Zhou, S., Zhang, H., Zhang, Y., & Zhang, H. (2024). Novel hyperchaotic image encryption method using machine learning-rbf. *Nonlinear Dynamics*, *112*(20), 18527-18550. https://doi.org/10.1007/s11071-024-09966-1

[19] Singh, A. & Pandey, A. (2023). Performance of adaptive radial basis functional neural network for inverter control. *Electrical Engineering*, *105*(2), 921-933. https://doi.org/10.1007/s00202-022-01706-1

[20] Vardakas, G. & Likas, A. (2024). Global k-means++: an effective relaxation of the global k-means clustering algorithm. *Applied Intelligence*, *54*(19), 8876-8888. https://doi.org/10.1007/s10489-024-05636-2

[21] Wang, K., Fu, Y., Zhou, S., Zhou, R., Wen, G., Zhou, F., & Li, L. (2024). Cloud detection from Himawari-8 spectral images using K-means++ clustering with the convolutional module. *International Journal of Remote Sensing*, *45*(3), 930-953. https://doi.org/10.1080/01431161.2024.2305625

[22] Bi, Y., Wigger, M., & Wu, Y. (2024). Normalized delivery time of wireless mapreduce. *IEEE Transactions on Information Theory*, *2024*(10), 7005-7022. https://doi.org/10.1109/TIT.2024.3423710

[23] Sleeman, W. C., Roseberry, M., Ghosh, P., Cano, A., & Krawczyk, B. (2024). Improved kd-tree based imbalanced big data classification and oversampling for mapreduce platforms. *Applied Intelligence*, *54*(23), 12558-12575. https://doi.org/10.1007/s10489-024-05763-w

[24] Yadav, S. K. & Kumar, R. (2024). ASME-SKYR framework: a comprehensive task scheduling framework for mobile cloud computing. *Wireless networks*, *2024*(3), 1221-1244. https://doi.org/10.1007/s11276-023-03565-5

[25] La Salvia, M., Torti, E., Marenzi, E., Danese, G., & Leporati, F. (2024). Edge and cloud computing approaches in the early diagnosis of skin cancer with attention-based vision transformer through hyperspectral imaging. *Journal of Supercomputing*, *80*(11), 16368-16392. https://doi.org/10.1007/s11227-024-06076-y

[26] Zheng, J., Qu, J., Cai, Z., Xue, Y., & Li, X. (2024). Modelling and simulation of cloud-native-based edge computing

terminals for power distribution. *IET generation, transmission & distribution*, *18*(21), 3365-3377. https://doi.org/10.1049/gtd2.13283

Contact information:

Xing YANG
(Corresponding author)
Geely University of China,
Chengdu, Sichuan, 610000, China
E-mail: 18980091368@163.com

Xiao Yu ZHAO
Geely University of China,
Chengdu, Sichuan, 610000, China
E-mail: zxy2570332345@gmail.com

Yong Hong ZHANG
Sichuan Huaxin Modern Vocational College,
Chengdu, Sichuan, 610000, China
E-mail: 553741881@qq.com