# Research on Improved Shallow Neural Network Big Data Processing Model based on Gaussian Function

Lifang Fu
Faculty of Science
Henan University of Animal Husbandry and Economy
Zhengzhou, 450044, China

***Abstract*—The application of the current new generation communication technology is gradually diversified, and the global Internet users are increasing, leading some large enterprises to increasingly rely on faster and more efficient big data processing technology. In order to solve the shortcomings of the current big data processing algorithms, such as slow computing speed, computing accuracy to be improved, and poor online real-time learning ability, this research combines incremental learning and sliding window ideas to design two improved radial basis function (RBF) neural network algorithms with Gaussian function as the kernel function. The Duffing equation example and the data of "Top 100 single products for Taobao search glasses sales" were used to verify the performance of the design algorithm. The experimental results of Duffing equation example show that when the total sample is 100000, the mean square errors of IOL, SWOL, SVM and ResNet50 algorithms are 1.86e-07, 1.59e-07, 3.37e-07 and 2.67e-07 respectively. The experimental results of the data set of "Top 100 SKUs for Taobao Search Glasses Sales" show that when the number of samples in the test set is 800, the root mean square errors of IOL, SWOL, SVM and ResNet50 algorithms are 0.0060, 0.0056, 0.0069 and 0.0073 respectively. This shows that the RBF online learning algorithm designed in this study, which integrates sliding windows, has a stronger comprehensive ability to process big data, and has certain application value for improving the accuracy of online data based commodity recommendation in e-commerce and other industries.**

***Keywords*—Gaussian function; RBF; big data processing; incremental learning; sliding window***

## I. Introduction

With the popularization of cloud computing, the Internet of Things and other technologies, a large number of industries began to explosively appear massive data, and the types of data also increased significantly, which began to show more and more characteristics of big data [1]. However, although the large data set has a large amount of data and low value density, it is still of great analysis and calculation value to discuss it as a whole, and some key information that is conducive to improving the business competitiveness of enterprises or the level of government services can be mined from the huge data [2-3]. However, it puts forward higher requirements for the processing speed and precision of the algorithm. At the same time, under the big data environment, there is another kind of more application demand, that is, the analysis model is updated regularly and in real time according to the updated data, so that the algorithm can meet the requirements of real-time update with the input data [4]. Traditional offline algorithms can not meet the needs of big data application scenarios for data mining, and the processing accuracy and speed of mainstream online learning algorithms used in the market are still poor. Therefore, it is necessary to design data analysis algorithms that more closely match big data application scenarios, which is also the main motivation for this study, The expected result is that the improved neural network algorithm designed in this research can improve the efficiency and accuracy of big data processing.

## II. Related Works

Experts in the fields of Internet of Things, e-commerce, communication and information have carried out a large number of researches involving various aspects for the rapid analysis and mining of big data. Valerio et al. found that processing online social big data fused with multimedia data sources is highly complex, so the research team evaluated two current state-of-the-art big data processing architectures, namely Lambda and Kappa, the discussion results show that for the investigated problem, Lambda outperforms the Kappa architecture [5]. The Habeeb research team believes that network anomaly detection is the key to Internet security. However, the survey results show that the existing methods for detecting network anomalies are not effective enough. The reason is that the devices connected to the network accumulate a large amount of data, which greatly increases the difficulty of algorithm detection. Therefore, the author first explains the current main methods of real-time big data processing, anomaly detection machine learning algorithms, and then reviews big data processing techniques. Finally, the research challenges of real-time big data processing in anomaly detection are discussed [6]. Zhang et al. proposed a near-computation-enhanced big data processing architecture that can better handle big data and its corresponding applications. The structure consists of near-edge computing and far-edge computing units. Simulation and experimental results show that the task assignment and architecture proposed by the author have certain effectiveness [7]. Lwin designed an extended program toolkit that can assist GIS to process meteorological and geographic big data in view of the weak ability of processing big data in current geographic information systems. The test results show that the toolkit can effectively improve geographic the speed and precision with which information systems process big data [8]. Cigna F et al. found that with the increase of satellite tasks and the increase of task complexity, the big data processing method of communication system is more and more difficult to meet the needs of use, so they designed an improved satellite big data combining deep learning algorithm and online learning idea. The simulation experiment results

show that the model can shorten the time-consuming system processing satellite big data and improve the processing accuracy [9]. Muhammad et al. found that as the amount of data collected during manufacturing increases, monitoring systems are becoming an important factor in management decisions. Therefore, the researchers propose a real-time monitoring system that combines IoT sensors, big data processing, and hybrid predictive models. The results show that IoT-based sensors and the proposed big data processing system are sufficient to effectively monitor the manufacturing process. Furthermore, the proposed hybrid prediction model has better fault prediction accuracy than other models when sensor data is input [10]. Wang et al. believe that with the increasing global requirements for public safety governance, various surveillance sensors installed and used provide a large amount of data for public safety decision makers and managers, and traditional data mining algorithms cannot well summarize these large data. The data value contained in the data, so the author designed a big data mining algorithm that mixes simulated annealing algorithm and RBF neural network. The experimental results show that the algorithm compared with the neural network and machine learning algorithm can better mine the abstract high-dimensional features in the data, and improve the accuracy of the analysis results [11].

To sum up, experts designed a variety of improved algorithms or improved hardware systems to solve the problem of low efficiency of big data processing in communication, industrial production and other fields, and achieved various research results. However, most of the algorithms they designed are complex, difficult to train and time-consuming, and few of them can realize the function of real-time algorithm updating based on new samples. Most importantly, the solutions proposed in these studies are often only for a certain sub field of big data processing, with poor universality. In view of these shortcomings, this research attempts to use a relatively simple RBF neural network to build a processing algorithm for various types of big data in e-commerce, Internet of Things and other industries that can realize online learning.

## III. Design of Improved RBF Big Data Processing Algorithm based on Gaussian Function

### A. Construction of RBF Algorithm Integrating Gaussian Function and Incremental Learning

After the training data scale increases significantly, the algorithm training time will also increase. If the training time does not change significantly, the computer needs to have stronger computing power [12-13]. In addition, if there are sample types that do not exist before in the increased samples, the algorithm trained to a certain extent cannot adapt to this change well. According to the traditional solution, the model should be retrained at this time. However, the re-learning process needs to retain most of the data left before, which will put forward higher requirements for database data storage [14]. Therefore, the incremental learning method can be used to solve such problems, because the original data is not necessary for the training of the incremental learning algorithm, so that it is less affected by the data size and requires more data storage capacity. It is more suitable for processing continuous large-scale real-time data [15].

The algorithm used in this study is built on the basis of RBF neural network. The RBF neural network is a neural network used to simulate the signal reception and processing between neurons in the human brain, so as to convert the linearly inseparable data in low-latitude space into linearly separable data in high-latitude space, and then cooperate with the combination of data layers to classify and regress the data. The typical structure of the network is shown in Fig. 1.

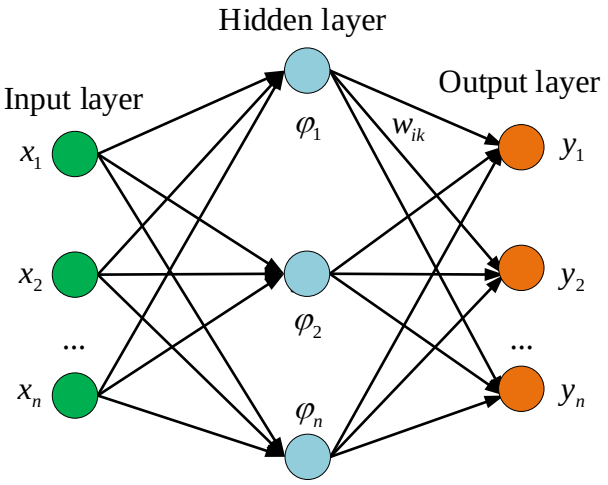


Fig. 1. Typical Structure of RBF Neural Network.

RBF neural network has the advantages of simple structure, fast training speed, good local optimality of algorithm output solution, and high approximation accuracy [16]. The RBF neural network consists of three structures: input layer, hidden layer, and output layer. It is a feedforward neural network and has the performance of approximating complex nonlinear functions. This ability is mainly provided by the hidden layer [17-18]. The hidden layer in the RBF neural network plays a linking role in the whole network. It is not only used to process the input vector using the specified basis function mapping, but also responsible for adjusting the weights to obtain the abstract integrated data of the output.

The following is a detailed discussion of incremental online learning and an analysis of the way it is incorporated into RBF neural networks. When a large data set is given, $u$ one sample is selected as the initial training set, and one sample is added during the incremental learning model construction process $L$ (taking into account the application scenario of this research, the $L$ value is 1), and the online model The input data matrix is constructed recursively, that is, the matrix of the previous stage is the calculation basis of the adjacent matrix, and the algorithm will continue to construct the data matrix until the samples are added, which is the core idea of incremental learning [19]. To achieve incremental learning, the following three conditions need to be met at the same time. First, the model must be able to learn data features that are different from the previous samples. That is to say, if the samples in the input data set $(x_j, y_j)$ have already appeared in the previous model training stage However, when it reappears in the application scene, it will be treated as a new instance of the old class. Finally, when there is a sample in the application scenario that the model has

never seen during training $(x_m, y_m)$, it will be treated as a new category. Secondly, during incremental learning, only new data and categories that have not appeared in the application scenario are $(x_k, y_k)$ used as training sets [20]. In general, incremental learning mainly has the following three implementations. The first is to filter the most informative samples, the second is to use a combination of multiple trained models to form an integrated model, and the third is to adjust the model. indicators and parameters.

It can be seen from the above content that the sample set in incremental learning is in an increasing state, that is, with $t$ the increase of running time, the sample set $\{X_i, Y_i\}_{i=t}^{i=t}$, or $\{(x_i, y_i)\}$ a fixed new $L=1$ sample will be added each time, if $x(t)=[x_1, x_2, ..., x_t]$, $y(t)=[y_1, y_2, ..., y_t]^T$, and $x_t \in R^n$, are assumed $y_t \in R$. Then the RBF matrix whose kernel function used in this study is a Gaussian function can be expressed in the form of Equation (1).

$$A=\begin{bmatrix} \phi(\|x_1-x_1\|) & \phi(\|x_1-x_2\|) & ... & \phi(\|x_1-x_t\|) \\ \phi(\|x_2-x_1\|) & \phi(\|x_2-x_2\|) & ... & \phi(\|x_2-x_t\|) \\ ... & ... & ... & ... \\ \phi(\|x_t-x_1\|) & \phi(\|x_t-x_2\|) & .... & \phi(\|x_t-x_t\|) \end{bmatrix}_{t\times t} \quad (1)$$

where $\phi()$ is the Gaussian kernel function, the radial basis function can be expressed by formula (1),

$$f(x)=\sum_{i=1}^{t}\beta_i\phi(\|x-x_i\|) \quad (2)$$

However, when using the RBF algorithm based on the Gaussian kernel function to build a big data processing model, it is also necessary to satisfy $f(x_i)=F(x_i)$ or $F=A\cdot\beta_i$, where $A$ is the radial basis kernel function, $F=[F(1), F(2), ..., F(t)]^T$, when the function $A$ is a positive definite function or the sample points of formula (2) do not coincide, there is formula ( 3).

$$\hat{\beta}=A^{-1}\times F \quad (3)$$

It can be seen from formula (3) that $\hat{\beta}$ the inverse function of the radial basis kernel function needs to be calculated first, but when the $A$ dimension is high, the calculation of the inverse matrix consumes a large amount of calculation, so the block matrix technique is used to calculate $A^{-1}$. It can $A_{t+1}$ be obtained by $A_t$ recursion, and the calculation formula between the two is deduced below. Observing $A_{t+1}$ the $A_t$ elemental composition of and, it is found that the former can be written in the form of formula (4).

$$A_{t+1}=\begin{bmatrix} A_t & H(t) \\ H(t)^t & f(t) \end{bmatrix} \quad (4)$$

In formula (4), $H(t)=[\phi(\|x_{t+1}-x_1\|), ..., \phi(\|x_{t+1}-x_t\|)]^T$, $f(t)=\phi(\|x_{t+1}-x_{t+1}\|)$. Now define a block matrix $B=\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$. And when $B^{-1}$ both $B_{11}^{-1}$ exist, there is the relationship of formula (5),

$$B=\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} =\begin{bmatrix} B_{11}^{-1} & 0 \\ 0 & 0 \end{bmatrix}+\begin{bmatrix} B_{11}^{-1} & B_{12} \\ -E & 0 \end{bmatrix}\cdot\left(B_{22}-B_{21}\cdot B_{11}^{-1}\cdot B_{12}\right)\left[B_{21}\cdot B_{11}^{-1}-E\right] \quad (5)$$

In Equation (5), $B_{11}$, $B_{12}$, are respectively a symmetric matrix and a column vector $I$, and . $B_{21}$., $B_{22}$ is $I^T$ a non-zero scalar vector $q$, $A=B_{22}-B_{21}\cdot B_{11}^{-1}B_{12}$ and if , Equation (5) can be simplified to Equation (6),

$$B^{-1}=\begin{bmatrix} B_{11}^{-1} & 0 \\ 0 & 0 \end{bmatrix}+R\cdot R^T\cdot Z \quad (6)$$

In formula (6) $R=\left[I^T\cdot B_{11}^{-1}-E\right]$, $Z=(q-I^T\cdot B_{11}^{-1}\cdot I)^{-1}$, can be obtained by using the block matrix inversion method $A_{t+1}^{-1}$, see formula (7).

$$A_{t+1}^{-1}=\begin{bmatrix} A_t^{-1} & 0 \\ 0 & 0 \end{bmatrix}+r_1(t+1)r_1(t+1)^T Z_1(t+1) \quad (7)$$

In formula (7) $r_1(t+1)=\left[H(t)^t\cdot A_t^{-1}, -E\right]^T$, it can be seen from the $Z_1(t+1)=\left[f(t)-H(T)^T\cdot A_t^{-1}\cdot H(t)\right]^{-1}$ observation of formula (7) that it $A_{t+1}^{-1}$ can be obtained by $A_t^{-1}$ calculation, and when it $t$ is small, it $A_t^{-1}$ can be directly obtained. According to the above two points, the large matrix inversion operation process can be effectively avoided, thereby improving the overall calculation efficiency of the algorithm. According to the above design content, an improved RBF neural network algorithm based on the fusion of Gaussian kernel function and incremental learning can be constructed, and the training process is shown in Fig. 2.

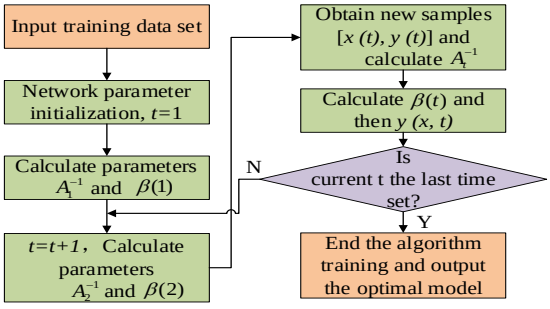


Fig. 2. The Training Flow Chart of the Improved RBF Neural Network Algorithm based on Gaussian Kernel Function and Incremental Learning.

*B. Design of Improved RBF Algorithm Combined with Windowed Online Learning Ideas*

The improved RBF algorithm based on incremental learning designed in the above content has significant advantages when dealing with static big data, which can simplify the training process and greatly reduce the training time of the algorithm. However, when using this algorithm to deal with another form of big data, that is, dynamic streaming big data that cannot be collected at one time, because dynamic streaming big data is real-time, volatile, bursty, disordered, and infinite. The improved RBF algorithm based on incremental learning previously designed is used to deal with static big data and dynamic big data, and the computing performance is obviously limited, so another processing algorithm needs to be designed for this data form. This research decided to build a windowed learning model by combining radial basis functions and sliding fixed-length windows on the basis of the improved algorithm of incremental learning RBF.

The windowed online learning algorithm is an online learning algorithm based on a sliding window. In this algorithm, new samples can only be added to the window after deleting an old sample, and the total length of the window remains stable. At the same time, in the process of dynamically updating the data set, the data newly added to the algorithm is processed, and the online processing of streaming dynamic big data is realized. Windowed online learning has the following advantages. First, the historical data used by the training algorithm does not need to be stored in a special technology, which can reduce computer memory consumption. Secondly, this type of algorithm can efficiently use the previously trained data while processing the new data, reasonably reduce the processing time of the new data, and reduce the time length of the algorithm. At the same time, in the windowed online learning algorithm, since the training samples are continuously updated and the historical training results are retained to a certain extent, the accuracy of the algorithm can be continuously improved. The last point, which is also the main reason for this research to choose this technology to improve the RBF network, is that the windowed online learning algorithm can update the window by overlapping online, which ensures that no matter how many samples are added, the single training time of the algorithm remains unchanged, so as to improve the training efficiency and effectively solve the processing problem of streaming dynamic big data.

The fixed-length sliding window strategy is adopted here, that is, each time a new sample is input in the algorithm, an old sample will be deleted at the same time, so as to maintain the total number of samples in each training. Assuming that the sample $T=\{x_i,y_i\}_{i=t}^{i=t+m+1}$ will be updated with time $t$, the sample set can be expressed as $\{x(t),y(t)\}$, $x_i$, $x(t)$ are the features of the variable sample, $t$ the feature vector of the input sample set at the time, $y_i$, $y(t)$ respectively represent the label of the variable sample, the input sample set at the $t$ time The label vector, and $x(t)=\{x_t,x_{t+1},...,x_{t+m+1}\}$, $y(t)=\{y_t,y_{t+1},...,y_{t+m+1}\}^T$, $x_t\in R^n$, and $y_t\in R$, $m$ are the window lengths (4 after adjusting the algorithm parameters many times in this study). $t$ At the moment, the Gaussian kernel function of the RBF radial basis function is in the $A_t$ form of an $m$ order square matrix, and its internal elements can be obtained by analogy with formula (1), which will not be repeated here $A_t$.

$$A_t=\begin{Bmatrix} h(t) & H(t)^T \\ H(t) & w(t) \end{Bmatrix} \tag{8}$$

Among them $h(t)=\phi(\|x_t-x_t\|)$, $H(t)=\left[\phi(\|x_{t+1}-x_t\|),...,\phi(\|x_{t+m-1}-x_t\|)\right]^T$, and the $w(t)$ calculation method is shown in formula (9).

$$w(t)=\begin{bmatrix} \phi(\|x_{t+1}-x_{t+1}\|) & ... & \phi(\|x_{t+1}-x_{t+m-1}\|) \\ ... & ... & ... \\ \phi(\|x_{t+m-1}-x_{t+1}\|) & ... & \phi(\|x_{t+m-1}-x_{t+m-1}\|) \end{bmatrix}_{(m-1)\times(m-1)} \tag{9}$$

Since $H(t)$, $h(t)$, $w(t)$ represent $m-1$ a column vector of dimensions, a scalar that is not 0, and $m-1$ a square matrix of dimensions, respectively. At the $t+1$ moment, new samples $(x_{t+1},y_{t+1})$ are added to the training set, and $(x_{t+1},y_{t+1})$ samples are deleted from the training set at the same time, which $A_{t+1}$ can be calculated according to formula (10).

$$A_{t+1}=\begin{bmatrix} w(t) & V(t+1) \\ V(t+1)^T & f(t+1) \end{bmatrix} \tag{10}$$

In formula (10) $V(t+1)=\left[\phi(\|x_{t+m}-x_{t+1}\|),...,\phi(\|x_{t+m}-x_{t+m-1}\|)\right]^T$, $f(t+1)=\phi(\|x_{t+m}-x_{t+m}\|)$. Therefore, the block matrix calculation method can be used to obtain $A_t^{-1}$, $A_{t+1}^{-1}$, and the former calculation method is shown in formula (11)

$$A_t^{-1}=\begin{bmatrix} 0 & 0 \\ 0 & w(t) \end{bmatrix}+R(t)\cdot R(t)^T\cdot Z(t) \tag{11}$$

In formula (11) $R(t)=\left[H(t)\cdot w(t)^{-1}-E\right]^T$, $Z(t)=1/\left[h(t)-H(t)^T\cdot w(t)^{-1}\cdot H(t)\right]$, can be described by formula (12) in the same $A_{t+1}^{-1}$ way.

$$A_{t+1}^{-1}=\begin{bmatrix} w(t)^{-1} & 0 \\ 0 & 0 \end{bmatrix}+R(t+1)\cdot R(t+1)^T\cdot Z(t+1) \tag{12}$$

In formula (12) $R(t+1)=\left[V(t+1)^T\cdot w(t)^{-1}-E\right]^T$, $Z(t+1)=1/\left[f(t+1)-V(t+1)^T\cdot w(t)^{-1}\cdot V(t+1)\right]$.

$h(t)\in R, H(t)\in R^{m-1}$ is composed of kernel function elements and vectors that delete samples during the sliding window movement, and $f(t+1)\in R$, $V(t+1)\in R^{m-1}$. Observing

formula (12), we can see that if we want to calculate it $A_{t+1}^{-1}$, we need to find it first $w(t)^{-1}$, and formula (13) can be deduced according to formula (11).

$$A_t^{-1}=\begin{bmatrix}0 & 0\\ 0 & w(t)^{-1}\end{bmatrix}+\begin{bmatrix}\dfrac{1}{Z(t)} & \dfrac{-H(t)^T w(t)^{-1}}{Z(t)}\\ \dfrac{-w(t)^{-1}\cdot H(t)}{Z(t)} & \dfrac{w(t)^{-1}\cdot H(t)\cdot H(t)^T\cdot w(t)^{-1}}{Z(t)}\end{bmatrix} \tag{13}$$

When it $A_t^{-1}$ is known, it can be obtained by its (1,1) position element , and then $\frac{1}{Z(t)}$ the row vector can be obtained $Q_1$ by using the first row element $A_t^{-1}$. The calculation method is shown in formula (14).

$$Q_1=\frac{-H(t)^T w(t)^{-1}}{Z(t)} \tag{14}$$

Then use $w(t)$ the first column vector element to calculate the column vector $Q_2$, and the calculation method is shown in formula (15).

$$Q_2=\frac{-w(t)^{-1}\cdot H(t)}{Z(t)} \tag{15}$$

Finally, combining equations (14) and (15), we can obtain

$$\begin{aligned}Q_3&=\frac{w(t)^{-1}\cdot H(t)\cdot H(t)^T\cdot w(t)^{-1}}{Z(t)}\\ &=Q_1(t)\cdot Q_2(t)\cdot Z(t)\end{aligned} \tag{16}$$

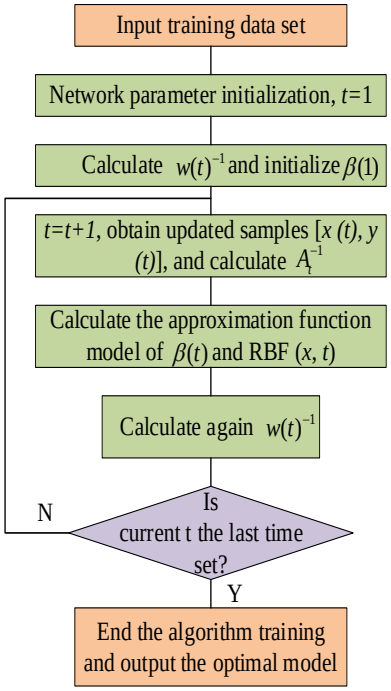


Fig. 3. The Training Flow Chart of the Improved RBF Neural Network Algorithm based on Sliding Window.

From the above content, it can be seen that the $w(t)^{-1}$ numerical value is equivalent to $\phi(t)^{-1}$ the matrix subtraction after removing the first row and the first column $Q_3(t)$, and then combining the formula (10) and formula (12), it can be obtained , that is, the $A_{t+1}^{-1}$ design of the recursive method from $A_t^{-1}$ to to is formed. $A_{t+1}^{-1}$. So far, the improved RBF online learning neural network algorithm based on sliding window is designed, and its calculation flow is shown in Fig. 3.

## IV. Improve the Performance Verification of RBF Algorithm

### A. Algorithm Performance Verification Experimental Design and Data Set Processing

This research plans to design two verification experiments. One is to use the Duffing equation example, which is widely used to verify the training effect of deep learning algorithms, to train and compare the performance of each algorithm. In this study, the parameters in the Duffing equation were set as $p_1=-1.1$, $p_2=0.4$, $\Delta t=0.01$, , $w=1.7$ and $q=1.5$, respectively, and the experimental plans of 1000, 3000, 5000, 10000, and 100,000 samples generated by the Duffing equation were set respectively . In order to avoid the influence of errors caused by random factors, each experimental scheme was repeated 10 times. In the experiment, the super parameter acquisition method of each algorithm is to use dichotomy to repeatedly run 10 times within the range of conventional parameter adjustment, and the parameter of the best run result is the final parameter. The second experiment scenario is a typical application scenario of big data mining algorithms, namely the Taobao online store sales forecast experiment. The data in the experiment is the data of "Taobao Search Glasses Sales Top 100 Single Items" provided by a domestic data supplier. The purpose of the experiment is to input this data into the algorithm to predict the sales volume of the glass single product in the store. A total of 5,500 sample data of Taobao e-commerce research and sales TOP100 single products were obtained, which contained a total of 18 variables. Except for the two characteristics of "commercial name" and "credit", which cannot be numerically processed, the others are numerical characteristics, which are respectively praised. rate, complaint rate only in 30 days, return rate in the past 30 days, sales volume in the past 30 days, original price, discount, postage, sales price, collection volume, days on the shelf, main credit ratio, customer evaluation, seller service points, goods Average refund speed, shipping score, description score. It is necessary to convert the "credit" feature that affects the prediction result in the data set. The value of this feature includes "other", "blue crown", "gold crown", "diamond", "heart", and "Tmall". Each equivalence contains several different secondary classifications. Statistics show that items with different credit ratings correspond to different frequencies of appearance, which means that there is a certain relationship between credit ratings and product sales. Therefore, the credit rating is assigned with reference to Taobao's credit regulations. The single product credit rating, frequency of occurrence, and numerical ranking results are shown in Table I.

TABLE I. THE STATISTICS OF THE "CREDIT" VARIABLE OF THE TOP100 SINGLE PRODUCT DATA SET OF TAOBAO SEARCH GLASSES SALES

| Grade 1 and 2 credit rating | The number of occurrences | Numerical mapping | Grade 1 and 2 credit rating | The number of occurrences | Numerical mapping |
|---|---|---|---|---|---|
| Tmall | 1125 | Twenty-one | 1 gold | 152 | 8 |
| 1 heart | 11 | 17 | 2 gold | 20 | 6 |
| 2 hearts | 15 | 16 | 3 gold | 33 | 7 |
| 3 hearts | 40 | 18 | 4 gold | 46 | 9 |
| 4 hearts | 46 | 20 | 1 blue | 319 | 10 |
| 5 hearts | 45 | 19 | 2 blue | 289 | 1 |
| 1 drill | 144 | 12 | 3 blue | 346 | 2 |
| 2 drills | 168 | 11 | 4 blue | 184 | 3 |
| 3 drills | 249 | 13 | 5 blue | 122 | 5 |
| 4 drills | 417 | 15 | other | 5 | 4 |
| 5 drills | 272 | 14 | / | / | / |

In order to improve the model training effect, the data set also needs to be processed by data dimensionality reduction. In order to compare the performance and application value of the algorithm, this study chose Support Vector Machine (SVM) and ResNet50 to build a comparative analysis model.

### B. Analysis of Validation Experiment Results

After the calculation of all experimental schemes under the two experiments is completed, the experimental data are counted. The following first analyzes the performance test experiment based on the Duffing equation example. Under the premise of five sample sizes, the statistical results of the mean square error of each algorithm are shown in Table II.

In order to verify whether there is a significant difference in the calculation results of each algorithm under each total experimental scheme, an F test was performed, and the significance level of the difference was set to 0.05. Note that IOL and SWOL in Table II represent the improved RBF algorithm based on incremental learning and the modified RBF algorithm incorporating sliding windows designed in this study, respectively. It can be seen from Table II that the root mean square error data P values of the output test set of each group of algorithms under various experimental schemes are all less than the significance level of 0.05, and the difference is considered significant. Specifically, under the experimental schemes of various sample numbers, the mean square error of the SVM model is the largest, followed by the ResNet50 model. The mean square error of the test set of the IOL and SWOL algorithms designed in this research is relatively small, and the mean square error of the SWOL algorithm is relatively small. minimum. For example, when the total sample is 100000, the mean square errors of the IOL, SWOL, SVM, and ResNet50 algorithms are 1.86e-07, 1.59e-07, 3.37e-07, and 2.67e-07, respectively. Then analyze the training time of each algorithm, and the statistical results are shown in Table III.

TABLE II. MEAN SQUARE ERROR STATISTICS OF EACH ALGORITHM ON THE DUFFING EQUATION EXAMPLE SAMPLE TEST SET

| Algorithm | Total samples 1000; test set: training set = 300:700 | Total samples 3000; test set: training set = 900:2100 | Total samples 5000; test set: training set = 1500:3500 | Total samples 10000; test set: training set = 3000:7000 | Total samples 100000; test set: training set=30000:70000 |
|---|---|---|---|---|---|
| IOL | 3.25e-04 | 4.36e-06 | 5.64e-07 | 3.28e-07 | 1.86e-07 |
| SWOL | 3.06e-04 | 4.12e-06 | 5.38e-07 | 3.02e-07 | 1.59e-07 |
| SVM | 4.57e-04 | 5.98e-06 | 6.74e-07 | 5.35e-07 | 3.37e-07 |
| ResNet50 | 3.59e-04 | 4.41e-06 | 5.92e-07 | 4.81e-07 | 2.67e-07 |
| *F* value | 1.651 | 1.548 | 2.154 | 5.833 | 9.645 |
| *P* value | 0.034 | 0.039 | 0.022 | 0.007 | 0.003 |

TABLE III. STATISTICS ON THE TRAINING TIME OF EACH ALGORITHM ON THE DUFFING EQUATION EXAMPLE SAMPLE (UNIT: SECOND)

| Algorithm | Total samples 1000; test set: training set = 300:700 | Total samples 3000; test set: training set = 900:2100 | Total samples 5000; test set: training set = 1500:3500 | Total samples 10000; test set: training set = 3000:7000 | Total samples 100000; test set: training set=30000:70000 |
|---|---|---|---|---|---|
| IOL | 3.2 | 73.8 | 389.4 | 3564.4 | 3.79e+06 |
| SWOL | 1.9 | 30.5 | 175.2 | 1442.0 | 1.48e+06 |
| SVM | 5.9 | 86.2 | 833.6 | 7957.9 | 8.25e+06 |
| ResNet50 | 4.2 | 74.2 | 759.1 | 6553.1 | 6.73e+06 |
| *F* value | 1.954 | 2.694 | 2.545 | 2.844 | 3.177 |
| *P* value | 0.031 | 0.021 | 0.020 | 0.016 | 0.013 |

The numbers in the cells of Table III represent the average model training time of the corresponding algorithm under the corresponding experimental scheme. It can be seen from Table III that the P values of the training time data of each group of algorithms under each experimental scheme are all less than the significance level of 0.05, and the difference is considered to be different. significant. Specifically, under the experimental schemes of various sample numbers, the training time of the SVM model is the largest, followed by the ResNet50 model. The training time of the IOL and SWOL algorithms in this study is relatively small, and the training time of the SWOL algorithm is the smallest. For example, when the total number of samples is 100,000, the training time of the IOL, SWOL, SVM, and ResNet50 algorithms are 3.79e+06, 1.48e+06, 8.25e+06, and 6.73e+06, respectively. The following is an analysis of the performance of each algorithm in the second experiment. First, the change of the loss function value of each algorithm during the training process is counted, as shown in Fig. 4.

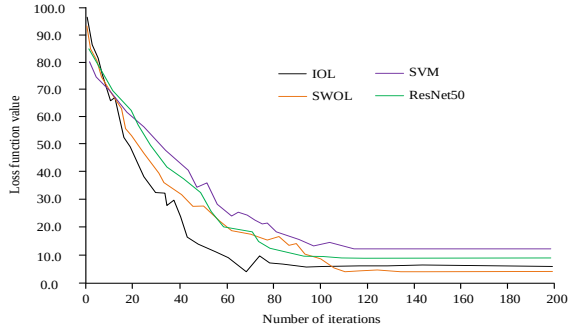


Fig. 4. Changes in the Loss Function Value of Each Algorithm during the Training Process of the e-commerce Dataset.

Observing Fig. 4, it can be seen that with the increase of the number of iterations, the calculation time of each algorithm shows an overall decreasing law, and finally converges. From the point of view of convergence speed, the IOL algorithm has the fastest convergence speed, and the algorithm training is roughly completed around the 85th time, while the SVM algorithm has the slowest convergence speed, which roughly completes the convergence around 117 times. After convergence, the loss function of the SWOL algorithm is the smallest, which is 4.2. The calculation time-consuming situation of various algorithms is counted, as shown in Fig. 5.

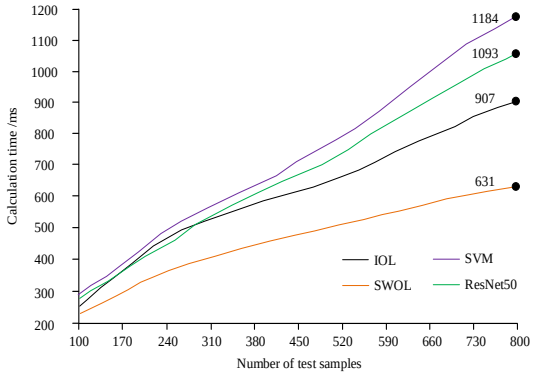


Fig. 5. The Calculation Time Change of Each Algorithm in the e-commerce Data Test Set.

Observing Fig. 5, it can be seen that with the increase of the number of test samples, the time consumption of each algorithm to process the test samples shows an overall increasing trend, but the overall calculation time of the SVM algorithm is the largest, and the overall calculation time of the SWOL algorithm is the smallest. When the number of samples in the test set is 800, the training time of the IOL, SWOL, SVM, and ResNet50 algorithms are 1184ms, 1093ms, 907ms, and 631ms, respectively. The processing performance of the algorithm is analyzed again, and the root mean square error variation of each algorithm on the test set is shown in Fig. 6.

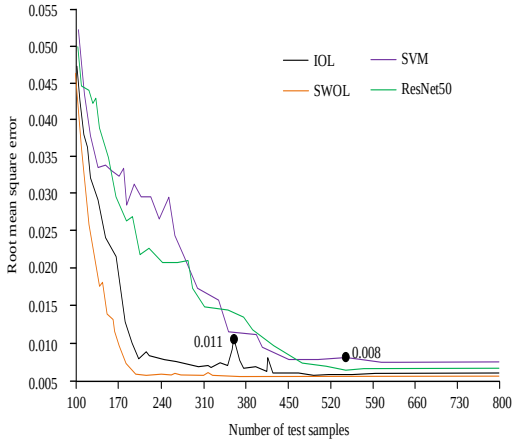


Fig. 6. Variation of the Root Mean Square Error of Each Algorithm in the e-commerce Data Test Set.

Observing Fig. 6, we can see that the root mean square error performance of the IOL and SWOL algorithms designed in this study is better under different test set samples, but the stability of the former is slightly worse than that of the latter, when the number of iterations is about 362 times, the root mean square error of the IOL algorithm has a large rebound. When there are 800 training samples, the root mean square errors of the IOL, SWOL, SVM, and ResNet50 algorithms are 0.0060, 0.0056, 0.0069, and 0.0073, respectively. In order to ensure the reliability of the research results, the change of the mean absolute error of each algorithm on the test set is finally analyzed, as shown in Fig. 7.

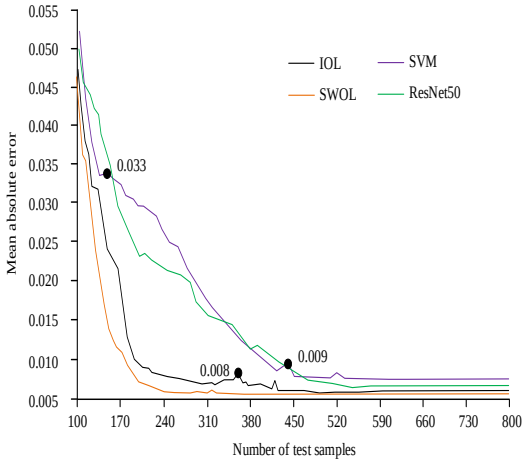


Fig. 7. Absolute Error Variation of Each Algorithm in the e-commerce Data Test Set.

Observing Fig. 7, it can be seen that the overall change of the mean absolute error of each algorithm with the increase of the number of test samples is basically the same as that of the root mean square error, but the variation range of the mean absolute error of each algorithm is reduced, especially the IOL and SWOL algorithms. However, there are still some data rebound phenomena in IOL. For example, when the number of training samples is about 360, the average absolute error of the IOL algorithm briefly rises to 0.008.

Finally, in order to further verify the application effect of the algorithm proposed in this study, 20 online sales store owners were selected from China and invited to use the customer in store consumption prediction system based on this algorithm and other comparison algorithms. The input data of this system are the customer's stay time in the store, the length and content of communication with customer service, the number of page clicks and the number of comparisons of similar products. The mean square error of the prediction results and the satisfaction of the shopkeepers (five point evaluation, the higher the score, the more satisfied) are used as indicators to measure the application effect of the algorithm. Since the data obtained from the experimental results are few and simple, the experimental results are described and analyzed in text. The experimental results show that the mean square error of the prediction results of the customer's in store consumption prediction system based on the IOL, SWOL, SVM and ResNet50 algorithms is 42.7 ￥, 29.6 ￥, 52.3 ￥ and 44.5 ￥ respectively, and the average of the shopkeeper satisfaction scores obtained are 3.85, 4.16, 4.03 and 3.61 respectively. It can be seen that the system designed based on this study has the most accurate consumption prediction and the highest user satisfaction.

## V. Conclusion

With the increase in the scale of Internet data, the use of big data technology to deal with huge data needs in e-commerce, Internet of Things and other industries is increasing. This research attempts to use the incremental learning idea to construct an improved RBF neural network, and combines the sliding window technology to design another improved RBF neural network. Using a variety of algorithms to predict the results of the Duffing equation example, the results show that when the total sample is 100,000, the mean square errors of the IOL, SWOL, SVM, and ResNet50 algorithms are 1.86e-07, 1.59e-07, and 3.37e, respectively. -07, 2.67e-07, the training time is 3.79e+06, 1.48e+06, 8.25e+06, 6.73e+06, respectively. The simulation experiment results of each algorithm in the "Taobao Search Glasses Sales TOP100 Single Product" data set show that when the number of samples in the test set is 800, the training time of the IOL, SWOL, SVM, and ResNet50 algorithms are 1184ms, 1093ms, 907ms, and 907ms, respectively. 631ms, the root mean square error is 0.0060, 0.0056, 0.0069, 0.0073 respectively. The two experimental results show that the two improved RBF algorithms designed in this study have faster training speed and higher data processing accuracy, and the improved RBF algorithm based on sliding window has the fastest speed in processing big data among the comparison algorithms.

## Reference

[1] F. Xu, X. Zhang, X. Song, et al. "Composite control of RBF neural network and PD for nonlinear dynamic plants using U-model," Journal of Intelligent & Fuzzy Systems, vol. 35(1), pp.565-575, 2018.

[2] T. Muling, T. Muqin, Y. Jieming et al. "Optimization of RBFneural network used in state recognition of coal flotation," Journal of Intelligent & Fuzzy Systems, vol. 34(2), pp.1193-1204, 2018.

[3] Y. Li, X. Chu, Z. Fu, et al. "Shelf-life prediction model of postharvest table grape using optimized radial basis function (RBF) neural network," British Food Journal, vol. 121(11), pp.2919-2936, 2019.

[4] Y. Tian, Y.L., He, Q.X. Zhu, "Soft sensor development using improved whale optimization and regularization-based functional link neural network," Industrial & Engineering Chemistry Research, vol. 59(43), pp.19361-19369, 2020.

[5] P. Valerio, P. Antonio, P. Antonio, et al. "Benchmarking big data architectures for social networks data processing using public cloud platforms," Future Generation Computer Systems, vol. 89(DEC.), pp.98-109, 2018.

[6] R. Habeeb, F. Nasaruddin, A. Gani, et al. "Real-time big data processing for anomaly detection: A Survey," International Journal of Information Management, vol. 45(4), pp.289-307, 2019.

[7] L. Zhang, K. Wang, D. Xuan, et al. "Optimal task allocation in near-far computing enhanced C-RAN for wireless Big Data processing," IEEE Wireless Communications, vol. 25(1), pp.50-55, 2018.

[8] K.K. Lwin, Y. Sekimoto, W. Takeuchi, "Development of GIS integrated Big Data research toolbox (BigGIS-RTX) for mobile CDR data processing in disasters management," Journal of Disaster Research, vol. 13(2), pp.380 -386. 2018.

[9] F. Cigna, D. Tapete, "Sentinel-1 BigData processing with P-SBAS InSAR in the geohazards exploitation platform: an experiment on coastal land subsidence and landslides in Italy," Remote Sensing, vol. 13(5), pp.885 - 910, 2021.

[10] S. Muhammad, A. Ganjar, F. Norma, et al. "Performance analysis of IoT-based sensor, big data processing, and machine learning model for real-time monitoring system in automotive manufacturing," Sensors, vol. 18(9), pp. 2946-2969, 2018.

[11] L. Wang, Y. Ma, J. Yan, et al. "PipsCloud: high performance cloud computing for remote sensing big data management and processing," Future Generation Computer Systems, vol. 78(6), pp.353-368, 2018.

[12] M. Hai, Y. Zhang, H. Li, "A performance comparison of big data processing platform based on parallel clustering algorithms," Procedia Computer Science, vol. 139(10), pp.127-135, 2018.

[13] J. Neto, A.M. Moreira, G, Vargas-Solar, et al. "A two-level formal model for Big Data processing programs," Science of Computer Programming, vol. 215(3), pp.102764-102766, 2022.

[14] S. Xie, Y. Xie, T. Huang, W. Gui, C. Yang, "Generalized predictive control for industrial processes based on neuron adaptive splitting and merging RBF neural network, IEEE Transactions on Industrial Electronics," vol. 66(2), pp.1192-1202, 2018.

[15] Y. Zhang, X. Hu, Z. Hui, et al. "Parameter interval optimization of the DBD plasma actuator based on orthogonal experiment and RBF neural network approximation model," Physics of Plasmas, vol. 28(2), pp.023504- 023505, 2021.

[16] Y. Yang, X. Lai, T. Luo, et al. "Study on the viscoelastic-viscoplastic model of layered siltstone using creep test and RBF neural network," Open Geosciences, vol. 13(1), pp.72-84, 2021.

[17] N. Shaukat, A. Ali, M.J. Iqbal, et al. "Multi-sensor fusion for underwater vehicle localization by augmentation of RBF neural network and error-state kalman filter," Sensors, vol. 21(4), pp.1149 - 1153, 2021.

[18] Q. Liu, D. Li, S.S. Ge, et al. "Adaptive bias RBF neural network control for a robotic manipulator," Neurocomputing, vol.447(4),pp.213-223, 2021.

[19] Y. Wang, X. Chen, "QSPR model for Caco-2 cell permeability prediction using a combination of HQPSO and dual-RBF neural network," RSC Advances, vol. 10(70), pp.42938-42952, 2020.

[20] A. Sh, A. Hw, T.A. Yang, et al. "Time-delay estimation based computed torque control with robust adaptive RBF neural network compensator for a rehabilitation exoskeleton," ISA Transactions, vol. 97(2), pp.171- 181, 2020.