Yarn unevenness prediction using generalized regression neural network under various optimization algorithms

Hao Jiang, Jiuxiang Song, Baowei Zhang and Yonghua Wang

Abstract
Unevenness is one of the important parameters for evaluating yarn quality, but the current prediction accuracy of yarn unevenness is low. One of the important reasons is that there are few sample datasets for yarn unevenness prediction. For this problem, this paper applies a generalized regression neural network to predict the unevenness of the yarn. Then, the generalized regression neural network is optimized by using particle swarm optimization, fruit fly optimization algorithm, and gray wolf optimizer, respectively. Finally, the optimized models were experimentally validated for their effectiveness. The experimental results show that the generalized regression neural network optimized by gray wolf optimizer has the best effect and the fastest optimization speed; the generalized regression neural network optimized by particle swarm optimization algorithm has the middle optimization speed; the generalized regression neural network optimized by fruit fly optimization algorithm has the worst effect and the slowest optimization speed.

Keywords
Yarn unevenness prediction, few sample dataset, generalized regression neural network, particle swarm optimization, fruit fly optimization algorithm, gray wolf optimizer

Date received: 25 January 2022; accepted: 23 March 2022

Introduction
The prediction of yarn quality metrics is always a key research topic for researchers in the textile field. The quality metrics of yarns include: strength, neps, unevenness, hairiness, and strength coefficient of variation. However, it is obvious from the existing researches that many scholars have focused on the strength of yarn, but there are few studies on the prediction of yarn unevenness. Unevenness is also one of the important yarn parameters compared to the yarn strength metric. It is defined as the irregularity of the yarn linear density in the direction of its length and it has a very important influence on the quality of yarn and fabric.

The prediction of unevenness of yarn is a complex nonlinear problem. For nonlinear problems, better results are often obtained using neural network, and many scholars have applied neural networks to the prediction of yarn unevenness. Üreyen and Gürkan used a neural network to predict the unevenness of ring spun yarn and concluded that the prediction of yarn unevenness using a neural network was better when compared with a linear regression method. Majumdar et al. used the method of combining neural network and fuzzy neural system to predict yarn unevenness. Demiryürek and Koç used a...
neural network and a traditional statistical model to predict the unevenness of polyester/viscose blended open-end rotor spun yarns. Malik et al.4 designed experiments and demonstrated the effectiveness of neural network with multiple linear regression on the index of unevenness of yarn for predicting the unevenness of yarn. Mwasiagi et al.5 proposed a hybrid algorithm for the problem that the back propagation (BP) algorithm in artificial neural network is easy to fall into local minima in the process of parameter search. The hybrid algorithm combines Differential Evolution algorithm and BP algorithm, and it is verified that the hybrid algorithm has good results in the prediction of yarn unevenness through experiments. Wu et al. 6 proposed another hybrid algorithm that combines Mind Evolutionary Algorithm and BP algorithm to accomplish the updating of weights and thresholds in the BP neural network, and the experiment proves that this hybrid algorithm works better compared to the unoptimized BP algorithm. Ghanmi et al. 7 proposed a fuzzy artificial neural network that combines expert experience which provides a direction for the optimization of artificial neural network to obtain better results by using expert experience. There are many parameters that affect the unevenness of yarn. Singh et al.8 researched the effect of several key process parameters including: spring stiffness, conveying speed, and coil position on yarn unevenness and showed that the position of the coil and the spring stiffness have a significant effect on the unevenness. Based on the idea of feature nodes and enhancement nodes, Jiang et al.9 proposed a method for yarn unevenness prediction based on broad multilayer neural networks.

Although artificial neural network has good results in predicting the unevenness of yarn, its prediction results still fail to reach the prediction target. A major reason for this problem is that artificial neural network often requires a certain amount of data to achieve the expected goal, and it is often difficult to collect enough clean data for artificial neural network training in textile mills, so the results obtained do not meet the expectations. To address this problem, Wang et al.10 developed a fuzzy integrated evaluation method to predict yarn unevenness. However, there is often a large gap between the predicted and actual values of their models. Whereas, generalized regression neural network (GRNN) proposed by Specht11 will converge to an optimized regression surface with a large accumulation of sample size and can have good prediction results when the data samples are small. Therefore, GRNN has been applied in many fields. Islam et al.12 used multiple GRNN to combine the operating conditions of the various subsystems of the transformer to form a quantitative transformer health index. Cheng et al.13 used GRNN to predict photovoltaic power generation. Benabdessalam et al.14 used GRNN to predict the basic characteristics of hydraulic jumps in a straight rectangular compound channel: (i) the sequent depths ratios, (ii) the relative energy losses, and (iii) the relative lengths. Yu et al.15 used GRNN to predict the shelf life of damaged Korla fragrant pears. Wu and Chen16 used GRNN to predict electric vehicle sales. However, its application in the field of textile prediction is still relatively small.

Although GRNN has achieved good results in many fields, there is still a key problem in GRNN: GRNN has a smoothing parameter, and the selection of the smoothing parameter determines the effect of the final algorithm. Therefore, it is necessary to optimize the selection of smoothing parameter in GRNN. For this aspect, many scholars have studied it, and there are three main optimization algorithms, which are particle swarm optimization (PSO), fruit fly optimization algorithm (FOA), and gray wolf optimizer (GWO). Among them, the GRNN optimized using PSO algorithm is called particle swarm optimization-generalized regression neural network (PSO-GRNN); the GRNN optimized using FOA is called fruit fly optimization algorithm-generalized regression neural network (FOA-GRNN); and the GRNN optimized by GWO is called gray wolf optimizer-generalized regression neural network (GWO-GRNN). The PSO-GRNN algorithm has been widely used in various fields. Gao and Chen17 used PSO-GRNN to predict sea clutter. Zhao et al.18 used PSO-GRNN to predict the occurrence of urban logistics public accidents. Zhang19 used PSO-GRNN to predict air quality. Zhang et al.20 used PSO-GRNN to predict landslide risk. Jia et al.21 used PSO-GRNN for target distance prediction in deep sea environment. The FOA-GRNN algorithm has also been widely used in various fields. Zhang et al.22 used FOA-GRNN to predict the performance of switched reluctance motor. Qiao et al.23 used FOA-GRNN to predict interlamellar spacing and mechanical properties of high carbon pearlitic steel. And Song et al.24 used FOA-GRNN to predict the air quality index. Similarly, the GWO-GRNN algorithm has been widely used in various fields. Ge et al.25 used GWO-GRNN for high-accuracy day-ahead short-term photovoltaic output prediction. Ge et al.26 used GWO-GRNN for short-term load forecasting of regional distribution network. Cao and Zhang27 proposed a model combining an adaptive dynamic GWO and GRNN for a complex problem and validated it on a complex industrial dataset.

In recent years, with the continuous research on metaheuristic algorithms, a variety of fresh and effective algorithms have emerged. Wang et al.28 proposed a new nature-inspired metaheuristic algorithm called monarch butterfly optimization by simplifying and idealizing the migration of monarch butterflies. The superior performance of monarch butterfly optimization is demonstrated by a comparative study of 38 benchmark problems with five other metaheuristic algorithms. Inspired by elephant herding behavior, Wang et al.29 proposed a new population-based metaheuristic search method, called Elephant Herding Optimization, for solving optimization tasks.
Mirjalili and Lewis\textsuperscript{30} proposed a novel nature-inspired metaheuristic optimization algorithm, called Whale Optimization Algorithm, which mimics the social behavior of humpback whales. Wang\textsuperscript{31} developed a new metaheuristic algorithm, called the moth search algorithm, inspired by moth phototropism and Lévy flight. Arora and Singh\textsuperscript{32} proposed a new nature-inspired algorithm, the butterfly optimization algorithm, that mimics the food search and mating behavior of butterflies to solve global optimization problems. Heidari et al.\textsuperscript{33} proposed a novel population-based, nature-inspired optimization paradigm called Harris Hawks Optimizer. The main inspiration for Harris Hawks Optimizer is the cooperative behavior and chasing style of Harris Hawks in nature, called surprise raids. Li et al.\textsuperscript{34} proposed a new stochastic optimizer, called the slime moth algorithm, based on the oscillation pattern of slime moths in nature. Abdollahzadeh et al.\textsuperscript{35} proposed a new heuristic algorithm inspired by the social intelligence of gorilla troops in nature called the Artificial Gorilla Troop Optimizer. Abdollahzadeh et al.\textsuperscript{36} proposed a new metaheuristic algorithm inspired by the lifestyle of the African condor. The algorithm, named African Condor Optimization Algorithm, simulates the foraging and navigation behavior of African condors. Although these algorithms have superior performance, these optimization algorithms are more commonly used to solve complex problems optimally, whereas the optimization of the smoothing factor in GRNN is a simple optimization problem, that is, using a simple optimization algorithm does not increase the complexity of the model and yields better results. Therefore, for yarn unevenness prediction problems with small samples, the use of PSO-GRNN, FOA-GRNN, and GWO-GRNN are the most superior choices.

A summary of the above shows that GRNN can still achieve good results with small data sets; PSO, FOA, and GWO are three commonly used algorithms for optimization of smoothing factors in GRNN, and all of them can achieve good results. Therefore, this paper studies GRNN and the corresponding optimization algorithms for the prediction of yarn unevenness.

**Principle and method**

**Generalized regression neural network**

GRNN is a radial basis function network based on mathematical statistics, and its theoretical basis is nonlinear regression analysis. GRNN has strong nonlinear mapping ability and learning speed, and has stronger advantages than Radial Basis Function Neural network. The network finally converges to the optimization regression with more sample size set, and the prediction effect is good when the sample data is small. The network can also handle unstable data. GRNN generally consists of four layers, which are input layer, mode layer, summation layer, and output layer. The network results are shown in Figure 1. Where, $(X_1, X_2, \ldots, X_n)$ are the input parameters, $(Y_1, Y_2, \ldots, Y_n)$ are the output parameters.

**Particle swarm optimization**

The basic concept of PSO algorithm originated from the study of foraging behavior of bird flocks. In the PSO algorithm, a particle is used to simulate the above-mentioned individual bird and each particle can be regarded as a search individual in the N-dimensional search space. The current position of the particle is a candidate solution of the corresponding optimization problem, and the flight process of the particle is the search process of the individual. The flight speed of particles can be dynamically adjusted according to the historical optimal position of particles and the historical optimal position of population. Particles have only two properties: speed and position. Speed represents the speed of movement and position represents the direction of movement. The optimal solution searched by each particle individually is called individual extremum, and the best individual extremum in the particle population is regarded as the current global optimal solution. Iterate continuously, update the speed and position through formulas (1) and (2), and finally get the optimal solution that meets the termination condition.

$$V_{id} = \omega V_{id} + C_1 \text{random}(0,1)(P_{gd} - X_{id}) + C_2 \text{random}(0,1)(P_{gd} - X_{id})$$

$$x_{id} = x_{id} + V_{id}$$

Where $x_{id}$ denotes the position of the $i$-th particle in the d-dimension, $V_{id}$ denotes the speed of the $i$-th particle in the d-dimension. $C_1$ and $C_2$ denote acceleration constants, random (0, 1) denotes a random number on the interval [0, 1], $P_{gd}$ denotes the d-th dimension of the individual extreme value of the $i$-th variable, and $P_{gd}$ denotes the d-th dimension of the global optimal solution. $\omega$ is called the inertia factor, whose value is non-negative.

**Fruit fly optimization algorithm**

FOA is an optimization algorithm based on the fruit fly foraging behavior. By simulating the foraging process of fruit flies using their keen sense of smell and vision, FOA achieves iterative search of the solution space for the population. Its specific optimization process is summarized as follows.

1) Initialization parameters: the size of the fruit fly population, the maximum number of iterations and population position, etc.
2) Nose search: random initialization the direction and position coordinate of fly individual searches food.

\[ X_i = X_0 + \text{Random} \]  \hspace{1cm} (3)
\[ Y_i = Y_0 + \text{Random} \]  \hspace{1cm} (4)

Where, \( X_i \) and \( Y_i \) denote the \( i \)-th position of the fruit fly (\( i = 0, 1, 2, 3, \ldots, n \)), \( n \) is the maximum number of iterations, and \( \text{Random} \) denotes the random number.

3) Calculation the smell density: calculates the distance between the origin and the fruit fly individuals, and determines the smell density value, which is the reciprocal of the distance.

\[ D_i = \sqrt{X_i^2 + Y_i^2} \]  \hspace{1cm} (5)
\[ S_i = 1 / D_i \]  \hspace{1cm} (6)

Where \( D_i \) denotes the distance of the fruit fly from the origin in the \( i \)-th iteration, and \( S_i \) denotes the value of taste concentration determination in the \( i \)-th iteration.

4) Fitness calculation: calculates the smell density (fitness value) of each individual position.

\[ \text{Smell}_i = f \left( S_i \right) \]  \hspace{1cm} (7)

Where \( \text{Smell}_i \) denotes the fitness value of the \( i \)-th iteration and \( f \) denotes the fitness function.

5) Finds the individual with the highest smell density in this fruit fly population.

6) Retains the best value of smell density and individual coordinate, other individuals within the population fly to this location.

7) Iterative optimization: repeats the steps 2)–5), and compared with the smell density of previous generation, if it is better, then goes to step 6).

**Gray wolf optimizer**

GWO is an optimization search method inspired by the prey hunting activity of the gray wolf, which has strong convergence performance, few parameters, and easy implementation, etc. The GWO optimization process contains the steps of social hierarchy stratification, tracking,
encircling, and attacking prey of the gray wolf, and the details of its steps are shown below.

1) Social hierarchy stratification. A social stratification model of gray wolves was constructed. The fitness of each individual in the population was calculated, and the three gray wolves with the best fitness in the population were labeled as $\alpha$, $\beta$, and $\delta$, while the rest were labeled as $\sigma$.

2) Encircling prey. The mathematical model of the behavior of a gray wolf gradually approaching its prey and encircling it is shown below.

$$D = C \cdot X_p(t) - X(t)$$ (8)

$$X(t + 1) = X_p(t) - AD$$ (9)

$$A = 2a_n - a$$ (10)

$$C = 2r_2$$ (11)

Where $t$ is the current number of iterations; $\cdot$ denotes Hadamard product operation; $A$ and $C$ are coefficient vectors; $X_p(t)$ denotes the position vector of the prey; $X(t)$ denotes the position vector of the current gray wolf; $a$ decreases linearly from 2 to 0 throughout the iteration; $r_1$ and $r_2$ are random vectors in $[0, 1]$.

3) Hunting. Wolves $\alpha$, $\beta$, and $\delta$ are assumed to be the first wolves closest to the prey in the wolf pack. The positions of the remaining wolves $\sigma$ are updated by the following equation.

$$D_\alpha = C_\alpha \cdot X_\alpha - X$$ (12)

$$D_\beta = C_\beta \cdot X_\beta - X$$ (13)

$$D_\delta = C_\delta \cdot X_\delta - X$$ (14)

$$X_1 = X_\alpha - A_1 \cdot D_\alpha$$ (15)

$$X_2 = X_\beta - A_2 \cdot D_\beta$$ (16)

$$X_3 = X_\delta - A_3 \cdot D_\delta$$ (17)

$$X(t + 1) = \frac{X_1 + X_2 + X_3}{3}$$ (18)

Where $X_\alpha$, $X_\beta$, and $X_\delta$ are the current positions of wolves $\alpha$, $\beta$, and $\delta$, respectively; $C_\alpha$, $C_\beta$, and $C_\delta$ are the coefficient vectors of wolves $\alpha$, $\beta$, and $\delta$, respectively; $A_1$, $A_2$, and $A_3$ are the coefficient vectors; and $D_\alpha$, $D_\beta$, and $D_\delta$ are the distances between the individual wolves $\alpha$, $\beta$, $\delta$, and the head wolf in the remaining wolves $\sigma$, respectively.

Data preparation

The data in this article comes from the production data of a textile mill in Anhui, China. The type of yarn produced is combed compact spinning spun yarn (100 counts).

Parameter setting

The parameters in the data used in this paper include input and output parameters. Among them, the input parameters are subdivided into three main parts: raw cotton parameters (before production), raw cotton parameters (detected during production), and parameters during machine processing, the parameters of each part and their corresponding units and abbreviations are shown in Tables 1 to 3. The output parameter is the unevenness (%) of the produced yarn, denoted as $y$.

### Table 1. Raw cotton parameters (before production).

| Parameter                              | Abbreviation |
|----------------------------------------|--------------|
| Short fiber percentage (%)             | $x_1$        |
| Neps (Neps/g)                          | $x_2$        |
| Micronaire value                       | $x_3$        |
| Strength (cN/tex)                      | $x_4$        |
| The quality length (mm)               | $x_5$        |
| Length uniformity (%)                  | $x_6$        |
| Impurity rate (%)                      | $x_7$        |

Division of training set and testing set

In this paper, the train test split function of the sklearn library package is used to split the dataset. The train_test_split function can divide the dataset into a training set and a testing set of different sizes and different sample distributions according to different random_seed and different test_size. In this paper, the data from a textile factory in Anhui, China, with 61 data, is divided into 54 training data and 7 testing data by using the train_test_split function.

Case study

In this case study, the focus is to verify the accuracy and robustness of GRNN and the optimized GRNN in predicting yarn unevenness. The algorithms used in the experiments were implemented in Python (3.7), TensorFlow (1.13.2), sklearn (0.0), pandas (1.0.1), neupy (0.8.2), matplotlib (3.1.3), NumPy (1.18.1), and sko (0.5.9) environments.

In this paper, the evaluation metrics of the model include mean absolute error (MAE), mean square error (MSE), root mean square error (RMSE), and Coefficient
Table 2. Raw cotton parameters (detected during production).

| Parameter | Abbreviation |
|-----------|--------------|
| The neps detected by AFIS before the cotton passes through the carding machine (Neps/g) | \( x_8 \) |
| The neps detected by AFIS after the cotton passes through the carding machine (Neps/g) | \( x_9 \) |
| The short fiber percentage detected by AFIS before the cotton passes through the carding machine (%) | \( x_{10} \) |
| The short fiber percentage detected by AFIS after the cotton passes through the carding machine (%) | \( x_{11} \) |
| The percentage of the quality of cotton lost during the process of cotton passing through the carding machine to the quality of the input cotton (%) | \( x_{12} \) |
| The neps detected by AFIS before the cotton passes through the combing (Neps/g) | \( x_{13} \) |
| The neps detected by AFIS after the cotton passes through the combing (Neps/g) | \( x_{14} \) |
| The short fiber percentage detected by AFIS before the cotton passes through the combing (%) | \( x_{15} \) |
| The short fiber percentage detected by AFIS after the cotton passes through the combing (%) | \( x_{16} \) |
| The Percentage Of The Quality Of Cotton Lost During The Process Of Cotton passing through the combing to the quality of the input cotton (%) | \( x_{17} \) |

Table 3. Parameters during machine operation.

| Parameter | Abbreviation |
|-----------|--------------|
| Carding linear speed (m/min) | \( x_{18} \) |
| Pre-drawing linear speed in the drawing process (m/min) | \( x_{19} \) |
| The working speed of combing machine (nips/min) | \( x_{20} \) |
| Roving twist rate (%) | \( x_{21} \) |
| Spun yarn twist rate (%) | \( x_{22} \) |
| The spindle speed of spinning frame (r/min) | \( x_{23} \) |
| Spun yarn draw ratio | \( x_{24} \) |
| Yarn count (tex) | \( x_{25} \) |

The parameters are compared with those obtained from the five-layer neural network with scale of 25, 32, 32, 32, 1. Artifical neural network is a common algorithm used to solve nonlinear problems, which has the advantages of simple structure and high accuracy. In this section, GRNN and Artificial Neural Network (ANN) are used to predict yarn unevenness. Among them, the ANN is a five-layer neural network with scale of 25 \( \times \) 32 \( \times \) 32 \( \times \) 32 \( \times \) 1 (25, 32, 32, 32, 1 are the number of neurons in the input layer, the first hidden layer, the second hidden layer, the third hidden layer, and output layer respectively), in which the activation function of the first three layers is the relu function and the activation function of the output layer is the sigmoid function. Since the performance of the GRNN network depends on the smoothing factor, the iterative method is used to select the smoothing factor. The initial value is 0 and 1000 iterations are performed in increments of 0.001, while the best result is selected as the experimental result of GRNN. Using the number of iterations of the smoothing factor as the horizontal coordinate and the MSE evaluation metric as the vertical coordinate, the relationship between the MSE evaluation metric of GRNN and the number of iterations of the smoothing factor is obtained as shown in Figure 2.

As can be seen in Figure 2, the error of the GRNN model under the MSE metric decreases and then increases with the number of iterations, which is close to a smooth curve. Therefore, in the problem of yarn unevenness prediction for small-scale data, the smoothing factor of GRNN is a simple optimization-seeking problem, and the optimization using PSO, FOA, and GWO algorithms is sufficient. Meanwhile, it is easy to conclude that the MSE value is minimized when the number of iterations is 367, which is when the smoothing factor takes the value of 0.367. The smoothing factor was chosen to be 0.367, and the obtained MSE metrics, RMSE metrics, MAE metrics, and \( R^2 \) metrics were compared with those obtained from the five-layer neural network to obtain the results in Table 4.

As can be seen in Table 4, although the five-layer neural network has good accuracy, the GRNN obtained by the

\[
\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\text{actual}_i - \text{predicted}_i)^2} \tag{21}
\]

\[
R^2 = 1 - \frac{\sum_{i=0}^{N} (\text{actual}_i - \text{predicted}_i)^2}{\sum_{i=0}^{N} \left( \text{actual}_i - \frac{1}{N} \left( \sum_{i=0}^{N} \text{predicted}_i \right) \right)^2} \tag{22}
\]

Where \( \text{actual}_i \) denotes the \( i \)-th true value, \( \text{predicted}_i \) denotes the \( i \)-th predicted value, and \( N \) denotes the total number of test sets.

Generalized regression neural network

Artificial neural network is a common algorithm used to solve nonlinear problems, which has the advantages of simple structure and high accuracy. In this section, GRNN and Artificial Neural Network (ANN) are used to predict yarn unevenness. Among them, the ANN is a five-layer neural network with scale of 25 \( \times \) 32 \( \times \) 32 \( \times \) 32 \( \times \) 1 (25, 32, 32, 32, 1 are the number of neurons in the input layer, the first hidden layer, the second hidden layer, the third hidden layer, and output layer respectively), in which the activation function of the first three layers is the relu function and the activation function of the output layer is the sigmoid function. Since the performance of the GRNN network depends on the smoothing factor, the iterative method is used to select the smoothing factor. The initial value is 0 and 1000 iterations are performed in increments of 0.001, while the best result is selected as the experimental result of GRNN. Using the number of iterations of the smoothing factor as the horizontal coordinate and the MSE evaluation metric as the vertical coordinate, the relationship between the MSE evaluation metric of GRNN and the number of iterations of the smoothing factor is obtained as shown in Figure 2.

As can be seen in Figure 2, the error of the GRNN model under the MSE metric decreases and then increases with the number of iterations, which is close to a smooth curve. Therefore, in the problem of yarn unevenness prediction for small-scale data, the smoothing factor of GRNN is a simple optimization-seeking problem, and the optimization using PSO, FOA, and GWO algorithms is sufficient. Meanwhile, it is easy to conclude that the MSE value is minimized when the number of iterations is 367, which is when the smoothing factor takes the value of 0.367. The smoothing factor was chosen to be 0.367, and the obtained MSE metrics, RMSE metrics, MAE metrics, and \( R^2 \) metrics were compared with those obtained from the five-layer neural network to obtain the results in Table 4.

As can be seen in Table 4, although the five-layer neural network has good accuracy, the GRNN obtained by the
iterative method has higher accuracy. The GRNN has low error than the five-layer neural network under MSE metric, RMSE metric, and MAE metric. Meanwhile, GRNN gets much higher $R^2$ value than the five-layer neural network, that is, GRNN has better regression characteristics. The comparative analysis of the two models shows that the neural network has some advantages for nonlinear problems (yarn unevenness prediction is a nonlinear problem), but it is difficult to get particularly excellent results when the data volume is small; while the GRNN can still get better results when the data volume is small, which proves the good effect of GRNN on small-scale yarn unevenness prediction problem.

**Optimized generalized regression neural network**

Although GRNN has achieved better results with the iterative method, the iterative method has limited effect on finding the best. To address this, this section studied the effectiveness of GRNN under various optimization algorithms (PSO, FOA, GWO) on the problem of predicting the unevenness of yarn. The parameters of PSO are set as follows: the number of populations is 10, the maximum number of iterations is 1000, and the weight is 0.8, and $C_1 = C_2 = 2$. The parameters about FOA are set as follows: the fruit fly population size is 10, the maximum number of iterations is 1000, and the flight radius of the fruit fly is 0.001. The parameters about GWO are set as follows: the number of value-seeking wolves is 10, the maximum number of iterations is 1000, and the lower limit of the smoothing factor searched is 0 and the upper limit is 1. Using the above parameters, the MSE metric of the test set is used as the fitness value to find the smoothing factor parameters of the GRNN algorithm. And the relationship between the MSE evaluation metrics of the GRNN optimized by PSO, FOA, and GWO algorithms and the number of iterations of the smoothing factor is plotted using the number of iterations as the horizontal coordinate and the MSE metrics of the test set as the vertical coordinate, as shown in Figures 3 to 5, respectively.

**Table 4.** Five-layer neural network compared with GRNN.

| Metric | Five-layer neural network | GRNN |
|--------|---------------------------|------|
| MSE    | 0.040210727               | 0.023465507 |
| RMSE   | 0.188692436               | 0.124845967 |
| MAE    | 0.200526125               | 0.153184551 |
| $R^2$  | -0.48479699               | 0.48882548  |

**Figure 2.** Variation curve of MSE value with the number of iterations (GRNN).

**Figure 3.** Variation curve of MSE value with the number of iterations (PSO-GRNN).

**Figure 4.** Variation curve of MSE value with the number of iterations (FOA-GRNN).
From the above figures, it is obvious that the PSO algorithm and GWO algorithm have the fastest optimization speed and the FOA algorithm has the slowest optimization speed. Analysis of the optimization process of their algorithms shows that the slow optimization speed of the FOA algorithm is due to the fact that the FOA algorithm performs a wide range of search through olfaction and sends food odor information to the surrounding fruit flies during foraging. In contrast, when a fly is found to have a higher concentration, the individual fly will rely on the visual function to fly to that location. It is precisely because of such population characteristics that the diversity of the population is reduced and have the defect of easily falling into local optimization, which leads to the premature maturation problem.

At the same time, the MSE metrics, RMSE metrics, MAE metrics, and $R^2$ metrics corresponding to the optimal results obtained by the three optimization algorithms are compared to obtain the results in Table 5.

From Table 5, it can be seen that the difference in the smoothing factors of the GRNN algorithms found by the PSO, FOA, and GWO algorithms is not large; under the MSE evaluation metric, the PSO-GRNN algorithm has the smallest error and the FOA-GRNN algorithm has the largest error; under the RMSE evaluation metric, the GWO-GRNN algorithm has the smallest error and the FOA-GRNN algorithm has the largest error; under the MAE evaluation metric, the GWO-GRNN and PSO-GRNN algorithms have the smallest error and the FOA-GRNN algorithm has the largest error; under the $R^2$ evaluation metric, the PSO-GRNN has the largest regression coefficient and the FOA-GRNN algorithm has the smallest regression coefficient. Although the difference between the results of the three algorithms is not large, it can be seen from the obtained data that the GWO-GRNN algorithm is the most stable and the best; the FOA-GRNN algorithm is the most unstable and the least effective.

To demonstrate the effectiveness of the optimized algorithm, running time comparison experiments were conducted using the optimized GRNN and the unoptimized GRNN, and the results obtained are shown in Table 6.

As can be seen in Table 6, both PSO-GRNN and GWO-GRNN have less running time than GRNN, while FOA-GRNN has more running time than GRNN. The analysis shows that the swarm intelligence optimization algorithm has a faster optimization speed compared to the traditional iterative method, but there is also a tendency to fall into a local optimum leading to a significant increase in the running time.

Finally, the $F$-test was used to prove the experimental results, and the results obtained are shown in Table 7.

Results and discussion

First, the above experimental contents are summarized. Among them, comparing the experimental results of the five-layer neural network and GRNN shows that: although the five-layer neural network can achieve good results in yarn unevenness prediction problem, GRNN has better results. Meanwhile, the $R^2$ metric of the five-layer neural network is negative, which reflects the unreliability of the results. That is, although the five-layer neural network can achieve good accuracy on the small-scale yarn unevenness prediction problem, the results are unreliable. In contrast, GRNN not only has high accuracy on yarn unevenness prediction, but also has a $R^2$ metric, which proves its effectiveness on small-scale data prediction. Comparing GRNN, PSO-GRNN, FOA-GRNN, and GWO-GRNN, it can be seen that GWO-GRNN not only has a very high accuracy but also has the least running time; FOA-GRNN has the worst results, even less effective than using the iterative method to find the smoothing factor in GRNN.

Next, the five-layer neural network and GRNN are compared and analyzed. Because of the small yarn unevenness data and the possible existence of erroneous data in the original data, good results can be obtained using the five-layer neural network, but the final results may have produced overfitting phenomena because of its model fitting based on the data only. Also, since GRNN has the ability to eventually converge to samples with more clusters, it is resistant to erroneous data, so it has better results with less data.

Finally, the GRNN optimized by iterative method, PSO-GRNN, FOA-GRNN, and GWO-GRNN are analyzed. The GRNN optimized by iterative method finds the optimal solution of the model by continuously iterating, and the accuracy of the model can be determined by adjusting the step size of each iteration. The accuracy of the model is proportional to the running time, and it does not
fall into the problem of local optimal solutions because each solution is traversed. PSO-GRNN optimizes the smoothing factor of GRNN by particle swarm algorithm, and it can get better results in less time than iterative method. FOA-GRNN takes longer time and runs worse than GRNN optimized by iterative method. GWO-GRNN is the best among the four models, which not only has the least running time but also has excellent running results.

In general, in practical production, the results of using PSO-GRNN and GWO-GRNN are excellent for small-scale yarn unevenness prediction problem. Meanwhile, GRNN optimized by iterative method is suitable for the case of simple models with low accuracy requirements. And FOA-GRNN is suitable for the case with low requirement of running time (because it is easy to fall into local optimum).

### Conclusion and future work

In the field of engineering, the data generally has the problem of small data size and the existence of unrecognized abnormal data. In response to this problem, some scholars have applied GRNN to the engineering field and has been widely used. In this paper, GRNN is proposed to be applied to the aspect of yarn unevenness prediction in practical engineering, and various optimization algorithms (PSO, FOA, GWO) are used to find the smoothing factor of GRNN with little yarn unevenness data and undifferentiated error data. The experimental results show that GRNN has a better result than the five-layer neural network. Meanwhile, GRNN optimized by PSO and GRNN optimized by GWO have better and very stable results in general, and GRNN optimized by FOA has poorer results in general.

Based on this, future research can be conducted in the following two aspects.

1) All the optimization algorithms proposed in the paper need to choose the appropriate parameters to get better results. How to choose the appropriate parameters is a major focus of the optimization algorithm.

2) Optimization algorithms can be optimized. The optimization algorithm can be chosen to optimize the optimization process.

### Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

### Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by the Science and Technology Research Program of Henan Province, China (202102210181).

### ORCID iD

Jiuxiang Song [https://orcid.org/0000-0002-0173-6824](https://orcid.org/0000-0002-0173-6824)

### References

1. Üreyen ME and Gürkan P. Comparison of artificial neural network and linear regression models for prediction of ring spun yarn properties. II. Prediction of yarn hairiness and unevenness. *Fibers Polym* 2008; 9(1): 92–96.

2. Majumdar A, Ciocoiu M and Blaga M. Modelling of ring yarn unevenness by soft computing approach. *Fibers Polym* 2008; 9(2): 210–216.
3. Demiryürek O and Koç E. Predicting the unevenness of polyester/viscose blended open-end rotor spun yarns using artificial neural network and statistical models. *Fibers Polym* 2009; 10(2): 237–245.

4. Malik SA, Farooq A, Gereke T, et al. Prediction of blended yarn evenness and tensile properties by using artificial neural network and multiple linear regression. *Autex Res J* 2016; 16(2): 43–50.

5. Mwasiagi JI, Huang X and Wang X. The use of hybrid algorithms to improve the performance of yarn parameters prediction models. *Fibers Polym* 2012; 13(9): 1201–1208.

6. Wu ZG, Zhang KJ and Yu CW. Cotton yarn quality prediction based on mind evolutionary neural network. *DEStech Transactions on Engineering and Technology Research, (MCEE)*, 2017.

7. Ghanmi H, Ghith A and Benameur T. Prediction of rotor-spun yarn quality using hybrid artificial neural network-fuzzy expert system model. *Indian J Fibre Text Res* 2019; 44(1): 31–38.

8. Singh S, Bhowmick N and Vaz A. Influence of can-spring stiffness, delivery speed and sliver coils position on unevenness. *J Text Apparel Technol Manage* 2019; 11(1): 1–12.

9. Jiang H, Song J, Zhang B, et al. Prediction of yarn unevenness based on BMNN. *J Eng Fiber Fabr* 2021; 16: 1–15.

10. Wang Y, Song J, Fan T, et al. Study of yarn quality prediction model based on fuzzy comprehensive evaluation. *J Inst Eng (India) Ser E* 2022; 2022: 1–9.

11. Specht DF. A general regression neural network. *IEEE Trans Neural Netw* 1991; 2(6): 568–576.

12. Islam M, Lee G, Hettiwatte SN, et al. Calculating a health index for power transformers using a subsystem-based GRNN approach. *IEEE Trans Power Deliv* 2018; 33(4): 1903–1912.

13. Cheng X, Gao C, Hu C, et al. Temperature & humidity-weighted-modified GRNN based prediction model of photovoltaic power generation. In: 2021 6th international conference on inventive computation technologies (ICICT), Coimbatore, India, 20–22 January 2021, pp.251–256. New York, NY: IEEE.

14. Benabdesselam A, Houichi L and Achour B. GRNN-based models for hydraulic jumps in a straight rectangular compound channel. *Model Earth Syst Environ* 2021; 2021: 1–12.

15. Yu S, Lan H, Li X, et al. Prediction method of shelf life of damaged Korla fragrant pears. *J Food Process Eng* 2021; 44(12): e13902.

16. Wu M and Chen W. Forecast of electric vehicle sales in the world and China based on PCA-GRNN. *Sustainability* 2022; 14(4): 2206.

17. Gao Z and Chen L. Sea clutter sequences regression prediction based on PSO-GRNN method. In: 2013 8th international symposium on computational intelligence and design (ISCID), Hangzhou, China, 12–13 December 2015, pp.72–75. New York, NY: IEEE.

18. Zhao M, Ji S and Wei Z. Risk prediction and risk factor analysis of urban logistics to public security based on PSO-GRNN algorithm. *PLoS One* 2020; 15(10): e0238443.

19. Zhang Y. Seasonal disparity in the effect of meteorological conditions on air quality in China based on Artificial Intelligence. *Atmos 2021*; 12(12): 1670.

20. Zhang D, Li Q, Ge R, et al. Prediction of landslide risk based on modified generalized regression neural network algorithm. In: Han Q, McLoone S, Peng C, et al. (eds) Intelligent equipment, robots, and vehicles. Singapore: Springer, 2021, pp.299–308.

21. Jia Y, Mo Y, Wang W, et al. Deep-sea source ranging method using modified general regression neural network. In: 2021 *OES China Ocean Acoustics (COA)*, Harbin, China, 14–17 July 2021, pp.1032–1037. New York, NY: IEEE.

22. Zhang Z, Rao S and Zhang X. Performance prediction of switched reluctance motor using improved generalized regression neural networks for design optimization. *CES Trans Electr Mach Syst* 2018; 2(4): 371–376.

23. Qiao L, Zhu J and Wang Y. Coupling physics in machine learning to predict interlamellar spacing and mechanical properties of high carbon pearlitic steel. *Mater Lett* 2021; 293: 129645.

24. Song C, Wang L, Hou J, et al. The optimized GRNN based on the FDS-FOA under the hesitant fuzzy environment and its application in air quality index prediction. *Appl Intell* 2021; 51(11): 8365–8376.

25. Ge L, Xian Y, Yan J, et al. A hybrid model for short-term PV output forecasting based on PCA-GWO-GRNN. *J Mod Power Syst Clean Energy* 2020; 8(6): 1268–1275.

26. Ge L, Xian Y, Wang Z, et al. A GWO-GRNN based model for short-term load forecasting of regional distribution network. *CSEE J Power Energy Syst* 2020; 7(5): 1093–1101.

27. Cao W and Zhang C. An effective parallel integrated neural network system for industrial data prediction. *Appl Soft Comput* 2021; 107: 107397.

28. Wang GG, Deb S and Cui Z. Monarch butterfly optimization. *Neural Comput Appl* 2019; 31(7): 1995–2014.

29. Wang GG, Deb S and Coelho LDS. Elephant herding optimization. In: 2015 3rd international symposium on computational and business intelligence (ISCBI), Bali, Indonesia, 7–9 December 2015, pp.1–5. New York, NY: IEEE.

30. Mirjalili S and Lewis A. The whale optimization algorithm. *Adv Eng Softw* 2016; 95: 51–67.

31. Wang GG. Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Comput* 2018; 10(2): 151–164.

32. Arora S and Singh S. Butterfly optimization algorithm: a novel approach for global optimization. *Soft Comput* 2019; 23(3): 715–734.

33. Heidari AA, Mirjalili S, Faris H, et al. Harris hawks optimization: algorithm and applications. *Future Gener Comput Syst* 2019; 97: 849–872.

34. Li S, Chen H, Wang M, et al. Slime mould algorithm: a new method for stochastic optimization. *Future Gener Comput Syst* 2020; 111: 300–323.

35. Abdollahzadeh B, Soleimanian Gharehchopogh F and Mirjalili S. Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. *Int J Intell Syst* 2021; 36(10): 5887–5938.

36. Abdollahzadeh B, Gharehchopogh FS and Mirjalili S. African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. *Comput Ind Eng* 2021; 158: 107408.