applied sciences

MDPI

Article

Application of an Optimized PSO-BP Neural Network to the Assessment and Prediction of Underground Coal Mine Safety Risk Factors

Dorcas Muadi Mulumba [1], Jiankang Liu [1,*], Jian Hao [1], Yining Zheng [2] and Heqing Liu [1]

[1] State Key Laboratory of Strata Intelligent Control and Green Mining Co-Founded by Shandong Province and the Ministry of Science and Technology, Shandong University of Science and Technology, Qingdao 266590, China; dorcasmuadi97@gmail.com (D.M.M.)

[2] Guotun Coal Mine of Shandong Energy Group Luxi Mining, Co., Ltd., Heze 274700, China

[*] Correspondence: ljk@sdust.edu.cn

Abstract: Coal has played an important role in the economies of many countries worldwide, which has resulted in increased surface and underground mining in countries with large coal reserves, such as China and the United States. However, coal mining is subject to frequent accidents and predictable risks that have, in some instances, led to the loss of lives, disabilities, equipment damage, etc. The assessment of risk factors in underground mines is therefore considered a commendable initiative. Therefore, this research aimed to develop an efficient model for assessing and predicting safety risk factors in underground mines using existing data from the Xiaonan coal mine. A model for evaluating safety risks in underground coal mines was developed based on the optimized particle swarm optimization-backpropagation (PSO-BP) neural network. The results showed that the PSO-BP neural network model for safety risk assessment in underground coal mines was the most reliable and effective, with MSE, $MAPE$, and R^2 values of 2.0×10^{-4}, 4.3, and 0.92, respectively. Therefore, the study proposed the neural network model PSO-BP for underground coal mine safety risk assessment. The results of this study can be adopted by decision-makers for evaluating and predicting risk factors in underground coal mines.

Keywords: coal; PSO-BP neural network; underground coal mines; safety risk factors; prediction

check for updates

Citation: Mulumba, D.M.; Liu, J.; Hao, J.; Zheng, Y.; Liu, H. Application of an Optimized PSO-BP Neural Network to the Assessment and Prediction of Underground Coal Mine Safety Risk Factors. *Appl. Sci.* **2023**, *13*, 5317. https://doi.org/10.3390/app13095317

Academic Editor: José Salvador Sánchez Garreta

Received: 26 March 2023
Revised: 15 April 2023
Accepted: 21 April 2023
Published: 24 April 2023

1. Introduction

Coal mining is a vital industry that provides a significant portion of the world's energy supply. However, it is also an industry that poses significant safety risks to workers. The unique conditions and hazards in coal mines make the work challenging [1], and the potential for accidents and injuries is high [2]. Coal mines, in particular, are susceptible to several hazards, including fires, explosions, cave-ins, and exposure to toxic gases. These hazards can lead to serious injuries, illnesses, and even fatalities among mine workers [3]. There has been extensive research demonstrating that underground coal mines are prone to frequent accidents, leading to the loss of lives and property [4]. Therefore, managing safety risks in coal mines is critical to ensuring workers' health and welfare and maintaining the industry's productivity and sustainability. Effective risk management requires a comprehensive understanding of the hazards and risks associated with coal mining and the implementation of sound safety policies, procedures, and controls to minimize these risks [5]. The goal of a coal mine safety risk assessment is to protect workers, equipment, and the environment from harm caused by coal mining operations [6]. To this effect, potential sources of hazards must be identified, and the likelihood and severity of potential consequences must be assessed. It is important to conduct regular risk assessments in coal mines to effectively identify and manage hazards and ensure that workers know the risks associated with their work [7]. By taking a proactive approach to risk assessment, mining companies can minimize the likelihood of accidents and injuries, protect their workers, and ensure compliance with relevant safety regulations [8].

In recent times, machine learning has become a promising tool that has supported this process. Machine learning can be used to analyze huge volumes of data from various sources, such as sensors, equipment logs, and worker behavior, to identify patterns and anomalies that could pose a risk to miners [9]. These algorithms can also predict potential hazards in real-time and provide early warning systems for miners and operators to take preventive measures [10]. With machine learning, data from various sources, such as geospatial data [11], seismic data [12], and mine ventilation systems, can be analyzed to assess the risk of geological hazards such as rock outbursts, mine collapses, and gas explosions [13]. In addition, data from mining equipment such as conveyors, drill rigs, and loaders can be analyzed to detect abnormal behavior that may lead to failures or accidents [14]. Additionally, machine learning algorithms can analyze worker behavior to identify unsafe practices and provide feedback to improve safety [15]. By incorporating machine learning into coal mine safety risk assessment, mine operators can gain a more accurate and comprehensive understanding of the risks associated with mining operations [16]. This approach can improve the efficiency and effectiveness of risk assessment and enable mine operators to take proactive measures to mitigate risks, prevent accidents, and protect miners' health and safety [17].

Backpropagation neural networks (BPNNs) have gained popularity in recent years as a tool to predict and analyze coal mine safety risks [18]. BP neural networks are artificial neural networks that use a supervised learning algorithm to train the network on a data set to learn and make predictions based on the input data [19]. This iterative forward and backward propagation process continues until the error is minimized and the network can accurately predict outputs for new inputs [20]. The BP neural network has been used in various applications related to coal mine safety, such as gas prediction [21], rock burst prediction [22], and personnel safety risk assessment [23]. Artificial neural networks (ANNs) are gaining popularity for solving complex real-world problems [24]. However, the traditional backpropagation neural network (BPNN) has some limitations that affect its performance and efficiency [25]. One of the most important limitations of the BP neural network is that it can easily get stuck in local optima, meaning it may not find the global optimal solution. This limitation affects the accuracy and reliability of the results obtained with the BPNN model [26]. It may be prone to overfitting, which means it performs well on training data but poorly on new, unseen data. Additionally, it requires a large amount of training data and can take a long time to converge [27].

The assessment and prediction of risk factors in underground coal mines are therefore considered important mechanisms for reducing or preventing mine accidents. In this study, we aimed to develop an efficient model for assessing and predicting safety risk factors in underground coal mines using existing data from the Xiaonan coal mine. Currently, we are not aware of any study in which the PSO-BP neural network model was proposed for the assessment and prediction of safety risk factors in underground coal mines. To overcome the limitations of the traditional BP neural network, an improved hybrid PSO-BP neural network was developed for the first time for the evaluation and prediction of safety risk factors in underground coal mines. The empirical analysis showed that the PSO-BP neural network model was the most reliable and effective method for the assessment and prediction of safety risks in underground coal mines.

2. Material and Methods

2.1. The BP Neural Network Model

A backpropagation neural network (BP) is a type of artificial neural network that uses a supervised learning algorithm for training [28]. It consists of multiple layers of interconnected processing units called neurons that work together to process information and make predictions [29]. The BP neural network uses a feedforward architecture where information flows in one direction from the input layer to the output layer through the hidden layers. During the training process, the network receives input data and produces an output based on the weights assigned to each neuron [30]. The difference between the predicted and actual output is measured by a loss function [31], and the weights are adjusted in the opposite direction of the gradient of the loss function using a technique called backpropagation. The backpropagation algorithm is used to update the weights of the neurons in the hidden layers and the output layer, and it uses the chain rule of calculus to propagate the error from the output layer back to the input layer [32].

This allows the network to learn from its errors and improve its predictions over time. BP neural networks are widely used for pattern recognition [33], classification [34], and regression tasks in fields such as image processing [35], natural language processing [36], and finance [37].

Suppose that X_1, X_2, X_3 ... , X_n are the independent variables in the BP neural network, which represent the faction that influences safety risk in the underground coal mine. Θ_1 is the output corresponding to the building model for predicting the safety risk in an underground coal mine. In addition, T_1 is the actual value of the safety risk for the corresponding training data; w_{ij} is the weight for each node in the hidden layer; and w_{jk} is the weight for each node in the output layer. The number of input nodes in the neural network is n, and g is the number of hidden nodes. The number of output nodes is S, and the threshold of each node is c.

a. In feed-forward propagation in the BP neural network, the output of the hidden layer is described as follows:

$$O_j = f\left(\sum_{i=1}^{n} w_{ij} x_i - c_j\right) \quad j = 1, 2, \ldots, g \tag{1}$$

The output of the output layer is as follows:

$$T_k = f\left(\sum_{j=1}^{g} O_j w_{jk} - c_k\right) \quad k = 1, 2, \ldots, s \tag{2}$$

Currently, the performance function in most BP network modeling toolboxes employs the Mean Squared Error (*MSE*) between the actual output and desired output. The learning approach of the BP network is to quickly reduce the weights and thresholds in the direction of the performance function [38]. The function is defined as follows:

$$E_k = \frac{1}{2}\sum_k (T_1 - \Theta_1)^2 \tag{3}$$

b. BP neural network Error

By substituting Equations (1) and (2) into (3). The error performance function is obtained as follows:

$$E_k = \frac{1}{2}\sum k\left(T_k\left(T_k - f\left(\sum_{j=1}^{g} W_{jk} f\left(\sum_{i=1}^{n} w_{ij} - c_j\right) - c_k\right)\right)\right) \tag{4}$$

By deriving the error function from the weight and threshold of the output, we have the following:

$$\frac{\partial E_k}{\partial w_{jk}} = -(T_k - \Theta_k) f'\left(\sum_{j=1}^{g} O_j w_{jk} - c_k\right) O_j \tag{5}$$

$$\frac{\partial E_k}{\partial c_k} = (T_k - \Theta_k) f'\left(\sum_{j=1}^{g} O_j w_{jk} - c_k\right) \tag{6}$$

The error of the output node is obtained as follows:

$$\delta_k = (T_k - \Theta_k) f'\left(\sum_{j=1}^{g} O_j w_{jk} - c_k\right) \tag{7}$$

By substituting Equation (7) into Equations (5) and (6), the following equations are described as follows:

$$\frac{\partial E_k}{\partial w_{jk}} = -\delta_k O_j \tag{8}$$

$$\frac{\partial E_k}{\partial c_k} = \delta_k \tag{9}$$

The weight and threshold adjustment formulas are described as follows:

$$w_{jk}(e+1) = w_{jk}(e) + \Delta w_{jk} = w_{jk}(e) + \eta \delta O_j \tag{10}$$

$$c_k(e+1) = c_k(e) + \eta \delta_k \tag{11}$$

In the hidden layer nodes, the weight and threshold are described as follows:

$$w_{ij}(e+1) = w_{ij}(e) + \Delta w_{ij} = w_{ij}(e) + \eta \delta_j x_i \tag{12}$$

$$\omega_j(e+1) = \omega_j(e) + \eta \delta_j \tag{13}$$

η is the learning rate of the BP neural network in Equations (10), (11), and (13).

Simulation

The evaluation object of the BP neural network is the overall safety of the coal mine; the number of neurons in the input layer is the same as the number of indicators in the index system for evaluating the safety risk in the underground coal mine, and a total of 46 secondary evaluation indicators were used as nodes in the input layer. In developing the predictive models, the input data were normalized before training the network to ensure the accuracy of the predicted results. This was performed to minimize the magnitude effect on the prediction results. The data sets were normalized to the range 0–1 using the following equations:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{14}$$

where X represents the original data, X_{norm} represents normalized data, and X_{max} and X_{min} are the maximum and minimum values, respectively, before normalization. The transformed data provides the risk evaluation value corresponding to the index, which satisfies the model's requirements. Of the data sets, 329 were selected for the training and testing data sets. Illingworth et al. suggested that 70–80% of the whole data set should be used as a training set. Therefore, 264 and 65 data sets were used in this study to develop the underground coal mine safety risk assessment prediction models.

1. Activation function selection: The transfer function selection between the layers of the BP neural network is an important part of the network. The coal safety risk assessment depends on various factors, such as the problem's complexity, the data set's size, and the desired output. In this study, the hyperbolic tangent (tanh) and sigmoid functions were adopted for the hidden and output layers, respectively.
2. Training function: The training function in the BP neural network is responsible for adjusting the network weights during the training process. The goal of the training function is to minimize the difference between the network's output and the desired output for a given input. This study used the Levenberg–Marquardt algorithm as the best and optimal training function to adjust the connection weight and reduce the mean square error.
3. Determination of the number of hidden layer nodes: the hidden layer node is determined based on the complexity of the problem being solved, the amount and quality of the available data, and the desired level of accuracy. Therefore, in this study, a trial-and-error approach was used. A different number of nodes was adopted, and

the performances of the networks were compared. The performance was measured in terms of the mean square error (MSE). The number that performed best on the validation set was optimal, as shown in Figure 1.

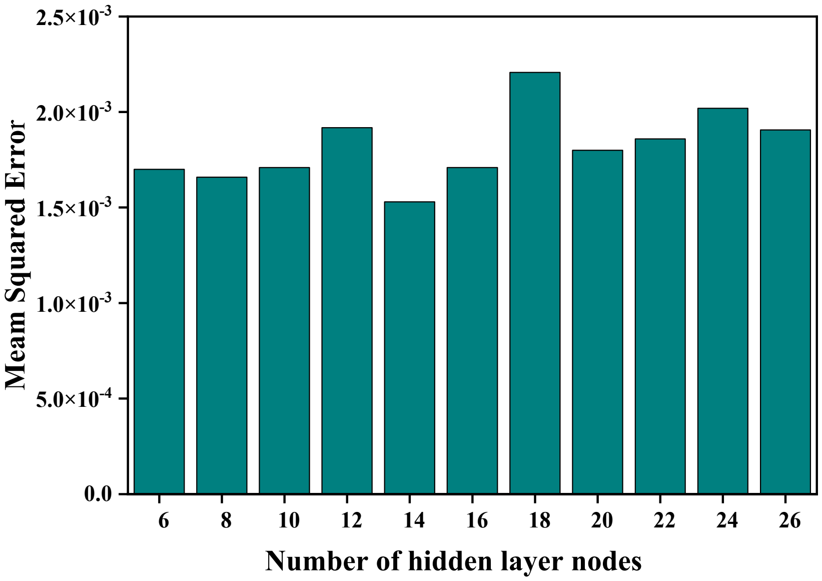

Figure 1. The number of hidden layer nodes.

To make the network training results more in line with the ideal results, this paper determines the number of hidden layer neurons by selecting the number of neurons in different hidden layers and comparing the network error when the number of hidden layer neurons is different. Therefore, 14 was selected as the optimal number for the hidden layer nodes.

4. Learning rate and lower momentum factor: The choice of the parameters in the BP neural network plays a critical role in the training process. A high learning rate can help the model converge quickly, but it may also cause the optimization algorithm to overshoot the optimal weight values and result in poor performance. A low learning rate, on the other hand, may cause the model to converge slowly or get stuck in local minima. The momentum factor can address these issues by helping the optimizer move more smoothly through the weight space and avoid getting stuck in local minima. A higher momentum factor can help the optimizer overcome local minima and reach the global minimum more quickly. In comparison, a lower momentum factor can help prevent overshooting and oscillations in the weight updates. In this study, to choose the best values for η and α, several BP neural network models were developed with η values of 0.02, 0.04, 0.06, 0.08, 0.01, and 0.2, respectively, and α values of 0.1, 0.2, 0.3, 0.4, 0.7, and 0.9, respectively. The *MSE* evaluation chose the optimal η and α values as 0.01 and 0.9, respectively.

2.2. Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) is a population-based metaheuristic optimization algorithm inspired by the social behavior of birds flocking or fish schooling. Kennedy and Eberhart first proposed the algorithm in 1995 [39]. The main aim of the PSO algorithm is to maintain a group of particles in a search space, where each particle represents a potential solution. A position vector and velocity vector characterize each particle. The position vector represents the potential solution in the search space, and the velocity vector determines the direction and speed of the particle's movement in the search space [40,41]. The best-known position of the swarms has been widely used to solve various optimization problems in recent years due to its simplicity, effectiveness, and ability to handle non-linear, non-convex, and multi-modal problems [42]. Recent research in PSO has focused on improving its performance, scalability, and applicability to various optimization problems. Hybrid algorithms combine the strengths of multiple algorithms to overcome their limitations. For example, PSO has been combined with the genetic algorithm (GA) to form a hybrid algorithm called PSO-GA [43], which has been shown to improve the performance of both algorithms. Similarly, PSO has been hybridized with Differential Evolution (DE) [44], which has been applied to solve complex optimization problems. [45] proposed an improved PSO algorithm that used a novel local search strategy to enhance its search performance for large-scale optimization. The proposed algorithm employs a guide search strategy that combines PSO with a local search method to accelerate the convergence and improve the diversity of the swarm. Ref. [46] presented a PSO algorithm that used a surrogate-assisted framework to solve multi-objective optimization. The proposed algorithm employs a surrogate model to approximate the objective functions, which can significantly reduce the computational cost and enhance the search performance. PSO has also been extended to solve multi-objective optimization problems using multi-objective PSO algorithms. Recent work has focused on improving the performance of multi-objective PSO algorithms by incorporating adaptive strategies, using Pareto dominance, and combining multi-objective PSO with other optimization algorithms [47]. Moreover, parallel PSO algorithms have been developed to take advantage of modern parallel computing architectures. One approach is to use parallel evaluation to speed up the evaluation of fitness functions [48]. Another is to use parallel population evaluation to accelerate the selection process.

The PSO algorithm starts by initializing a population of particles randomly in the search space. Each particle evaluates its fitness value based on the objective function to be optimized. A fitness function is defined to evaluate the fitness of each particle based on its position in the search space. The fitness function measures how well the particle's position solves the optimization problem and is used to determine the quality of the candidate solutions [49]. The particles update their position and velocity vectors based on their own best solution (personal best) and the best solution found by any particle in the swarm (global best). The velocity vector determines the direction and speed of the particle's movement, while the position vector determines the particle's new position in the search space [50]. In PSO, a swarm of particles is used to explore a search space in search of an optimal solution to an optimization problem. Each particle in the swarm represents a candidate solution to the problem and is located in a D-dimensional search space, where D is the number of problem variables [51]. The particles are initialized randomly within the search space, and their positions and velocities are updated iteratively based on their own best position and the best position of their neighbors in the swarm [50]. During the optimization process, the swarm of particles moves through the search space, with each particle adjusting its position and velocity based on its own experience and the experiences of its neighbors. This collective behavior allows the swarm to effectively explore the search space and converge towards an optimal solution [52]. The velocity V_i^k and the position X_i^k of the ith particle are updated as follows [50]:

$$V_i^k \leftarrow V_i^k + C1 * rand1_i^k * \left(pbest_i^k - X_i^k\right) + C2 * rand2_i^k \left(gbest^k - X_i^k\right) \tag{15}$$

$$X_i^k \leftarrow X_i^k + V_i^k \tag{16}$$

where X_i is the position of the ith particle, V_i is the velocity of the particle i, *pbest i* represents the best position yielding the best fitness value of the ith particle, and gbest is the position discovered by the whole population. *C1* and *C2* are acceleration coefficients that control the influence of *pbest* and *gbest* on the particle's movement. *rand1* and *rand2* are used to introduce randomness into the particle's movement. The values of *rand1* and *rand2* are multiplied by the acceleration coefficients *C1* and *C2*, respectively [53]. Since its debut by Kennedy and Eberhart in 1995, PSO has attracted significant attention [54]. Numerous researchers have sought to improve its performance in a variety of ways, resulting in the development of many intriguing varieties. One of the variants [55] introduces an inertia weight parameter to the basic PSO algorithms as follows:

$$V_i^k \leftarrow w * V_i^k + C1 * rand1_i^k * \left(pbest_i^k - X_i^k\right) + C2 * rand2_i^k * \left(gbest^k - X_i^k\right) \tag{17}$$

$$W = w_{max} - iter \cdot (w_{max} - w_{min}) / iter_{max} \tag{18}$$

where *iter* and $iter_{max}$ are, respectively, the current iteration and maximum iteration, and w_{min} to w_{max} is the range of inertia weight [0.4, 0.9], commonly. If w is too high, particles will move too quickly and may overshoot the optimal solutions. On the other hand, if w is too low, particles will move too slowly and may get trapped in local optima [56]. A range of values for w between 0.4 and 0.9 has been found to work well in many applications [39,57–59]. The lower bound of this range, 0.4, corresponds to a high degree of exploration, meaning that particles are more likely to explore new regions of the search space. The upper bound of the range, 0.9, corresponds to a high degree of exploitation, meaning that particles are more likely to converge towards the best solutions found. This is useful in the later stages of the search process, when the algorithm has already explored much of the search space and has a good idea of the problem landscape [60,61].

At each iteration, the particles adjust their position based on a combination of their best-known position and the best-known position of the swarm. The algorithm continues to iterate until a stopping criterion is met, such as a maximum number of iterations or a satisfactory solution is found. PSO has been used to solve a variety of optimization problems, including function optimization [62], neural network training, and image segmentation [63]. One advantage of PSO is its simplicity and ease of implementation, making it a popular choice for researchers and practitioners.

3. Network Optimization of the Coal Mine Safety Risk Assessment

3.1. Modeling

3.1.1. The GA-BP Neural Network

The traditional BP neural network can easily get stuck in local minima and converge slowly, especially for large networks or complex problems. To overcome this limitation, this section introduces the neural network GA-BP, which can accelerate convergence through a more efficient optimization method. Figure 2 shows that the GA-BP neural network consists of three components: BP neural network structure determination, genetic algorithm optimization, and BP neural network prediction. The BP neural network structure determination section is based on the number of input and output parameters of the fitting function and the length of each one. The fitness function based on the prediction error (*MSE*) of the BP neural network is used to evaluate the performance of each individual. For the individual fitness value, the genetic algorithm finds the optimal fitness value that matches the individual through selection, crossover, and mutation operations. The BP neural network prediction uses a genetic algorithm to select the optimal individual and assign the initial weights and thresholds. The network is trained to predict the output of the function [64].

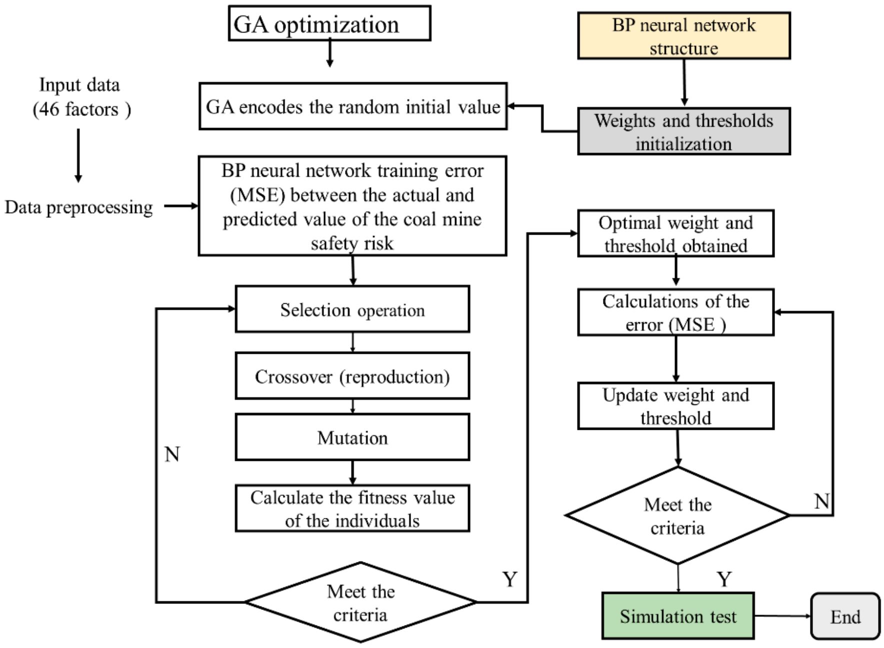
GA optimization
BP neural network structure
Input data (46 factors)
GA encodes the random initial value
Weights and thresholds initialization
Data preprocessing
BP neural network training error (MSE) between the actual and predicted value of the coal mine safety risk
Optimal weight and threshold obtained
Selection operation
Calculations of the error (MSE)
Crossover (reproduction)
Update weight and threshold
Mutation
N
Calculate the fitness value of the individuals
Meet the criteria
N
Y
Meet the criteria
Y
Simulation test
End

Figure 2. GA-BP neural network algorithm process.

Taking the *MSE* value of the prediction error of the training set as the individual fitness value, the smaller the fitness value, the better the individual at each iteration.

3.1.2. The PSO-BP Neural Network

In the BP neural network, the network learns by adjusting the weights and biases of its neurons in response to the training data [65]. The objective is to minimize a cost function that measures the error between the network's predicted and actual outputs. However, a common problem with BP neural networks is that they can get stuck at local minima. Local minima are points in the weighting space where the cost function has a lower value than in surrounding areas but may not be the global minimum, which is the best set of weights for the network [66]. If the network is stuck at a local minimum, it may not be able to improve its performance further, even if there is a better set of weights that could lead to a lower error. The PSO-BP neural network algorithm is robust to noisy and incomplete data [67]. This is because the PSO algorithm can effectively deal with noisy data by exploring the entire search space and finding the optimal set of weights that minimizes the error even in the presence of noise [68]. To develop the neural network PSO-BP, the PSO algorithm was used to find the optimal initial weights and thresholds for the BP neural network. The particles in the swarm represent different sets of weights, and their positions are updated based on their own and the global best positions. The fitness of each particle was evaluated based on the error between the predicted and actual outputs of the neural network with the corresponding weights. Using the optimal structure of 46 × 14 × 1, the coal mine safety risk evaluation model based on the PSO-BP neural network structure is shown in Figure 3.

The initialization of weights and thresholds was randomly selected. A total of 1673 weights were used to determine the length of particles in the initial population. The fitness function of the particles was evaluated at each iteration by training the BP neural network with its weights and calculating its performance on the training datasets. The velocity of each particle is updated by iterating its *pbest* and *gbest*. After the PSO algorithm updated the weights and thresholds of each particle, the BP algorithm was used to train the network with the updated weights and thresholds. The BP algorithm uses the error between the predicted output and the actual output to adjust the weights and thresholds of the neural network. The PSO-BP algorithm terminates when a termination criterion is met, such as reaching a maximum number of iterations or a certain accuracy level. In this study, *MSE* was used to evaluate the fitness function of the BP neural network on the training dataset for the i-th particle. The smaller the fitness value of a particle, the better the particle. The flowchart of the learning algorithm is shown in Figure 4.

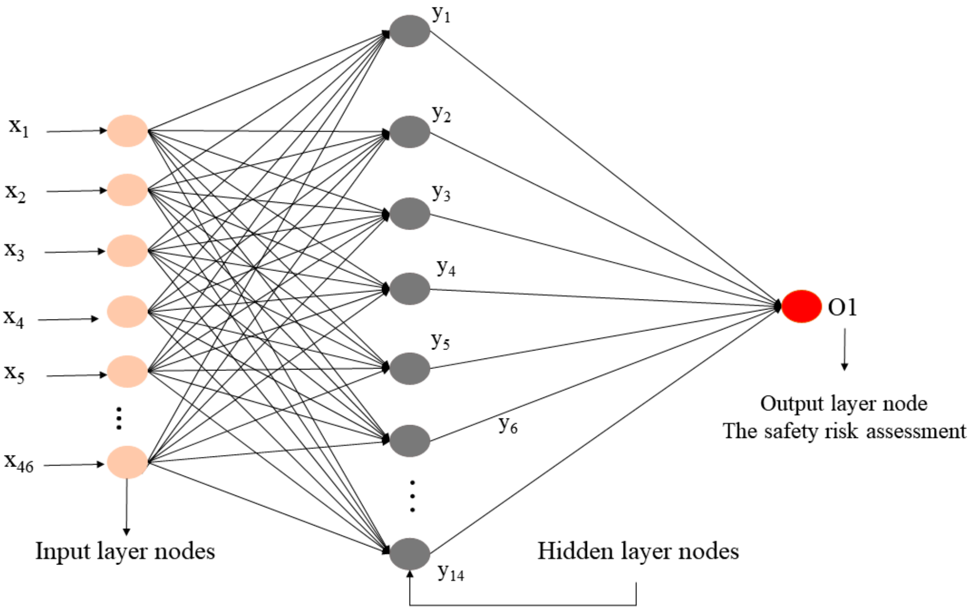

Figure 3. Coal safety risk assessment based on the PSO-BP neural network's optimal structure.

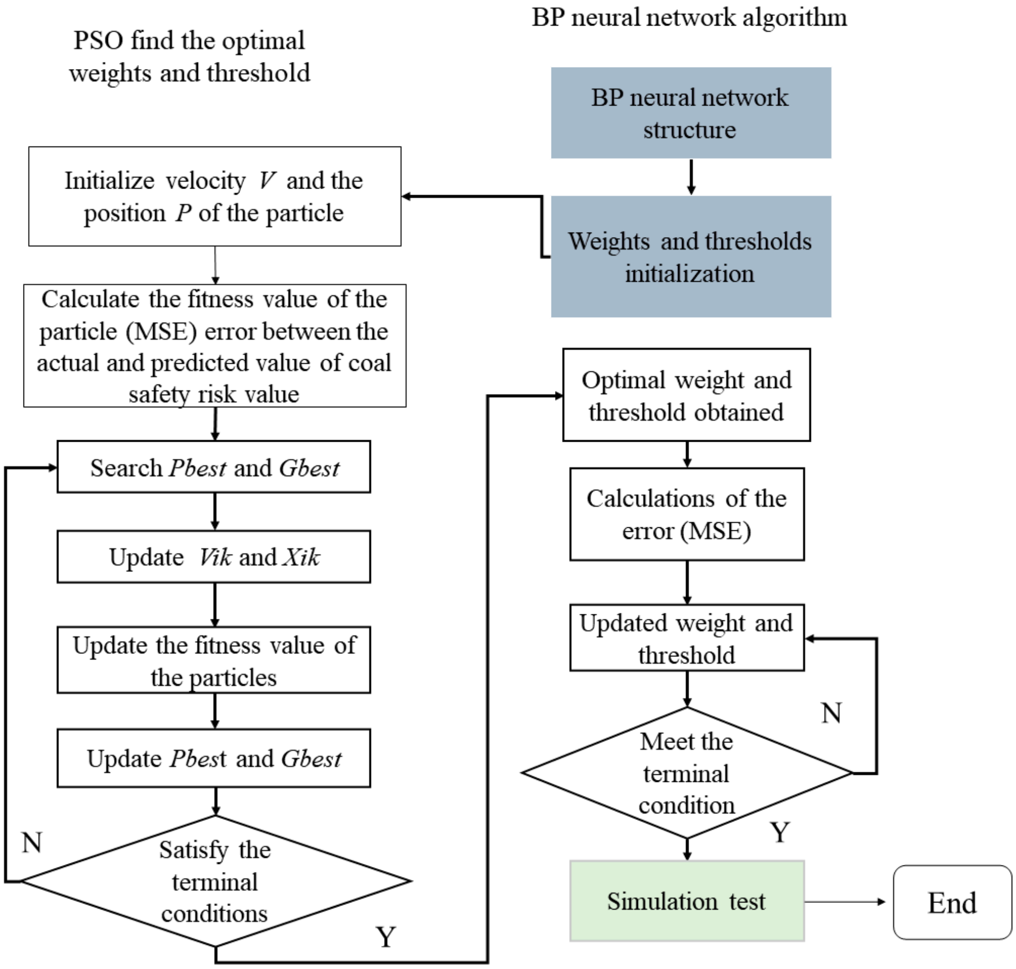
PSO find the optimal weights and threshold
BP neural network algorithm
BP neural network structure
Weights and thresholds initialization
Initialize velocity *V* and the position *P* of the particle
Calculate the fitness value of the particle (MSE) error between the actual and predicted value of coal safety risk value
Search *Pbest* and *Gbest*
Update *Vik* and *Xik*
Update the fitness value of the particles
Update *Pbest* and *Gbest*
Satisfy the terminal conditions
N
Y
Optimal weight and threshold obtained
Calculations of the error (MSE)
Updated weight and threshold
Meet the terminal condition
N
Y
Simulation test
End

Figure 4. PSO-BP neural algorithm process.

The Optimized Parameters of the Network Model

In order to verify the optimization methodology and the accuracy of the observed values, the process was performed based on the results provided in the developed MATLAB (R2019a) code. The initialization of the PSO algorithm involves setting various parameters that govern the behavior of the swarm. These parameters include the number of particles in the population, the number of iterations, the inertia weight, and the parameters that control the update of the velocity and position vectors. As shown in Table 1, the initialization parameters were described as follows.

Table 1. PSO initial optimization parameters.

Optimization Parameters	Values
The number of particles in the population (SwarmSize)	50
The maximum number of iterations	500
Inertia weight (W)	0.60
The inertia weight damping ratio	0.40
The personal learning coefficient (*C1*)	2.5
The global learning coefficient (*C2*)	2.5

Number of Particles in the Population

The number of particles in the PSO population is an important hyperparameter that can affect the performance and convergence speed of the algorithm. Generally, a larger population size can increase the diversity of the search and reduce the chance of getting stuck in local optima, but it also requires more computational resources and may lead to slower convergence [69]. Selecting the number of particles in the PSO population can be a challenging task that requires careful consideration of the problem characteristics, the available resources, and the desired performance [70]. A good practice is to perform a sensitivity analysis to evaluate the effect of different population sizes on the optimization results and to choose a reasonable value based on the trade-off between exploration and exploitation [71].

The Number of Iterations

Iterations refer to the number of times the population of particles is updated before the algorithm terminates. The more iterations an algorithm has, the more thoroughly the search space can be explored, but it also requires more computing resources and may result in slower convergence [72]. Similar to the selection of the population size, there are different approaches to determining the number of iterations in PSO. One common approach is to set a fixed number of iterations based on prior experience or a rule of thumb [73]. For example, a common rule of thumb is to run PSO for 500–2000 iterations for most optimization problems.

Inertia Weight

The role of the inertia weight is to balance the global and local search behaviors of the particles. A higher inertia weight favors global exploration, allowing the particles to move faster and cover a larger search space [74]. A lower inertia weight favors local exploitation, allowing the particles to converge more tightly around the best solutions found. In this study, the adaptive approach has been used. The equation is described as follows:

$$v_i^{(t+1)} = phi * \left[v_i^{(t)} + C1 * r1 * \left(pbest_i - x_i^{(t)}\right) + C2 * r2 * \left(gbest - x_i^{(t)}\right) \right] \quad (19)$$

where *phi* is the sum of the cognitive and social learning factors.

The Inertia Weight Damping Ratio

The inertia weight damping ratio is a parameter used in some adaptive approaches for setting the inertia weight in PSO. The damping ratio is used to adjust the rate at which the inertia weight changes over iterations in order to balance exploration and exploitation and improve convergence [75]. According to the formula below, the damping ratio is the ratio of the current and maximum inertia weights.

$$d = \frac{w^{(t)}}{w_{\max}} \tag{20}$$

where $w^{(t)}$ is the current inertia weight at iteration t, and $W_{\max}$ is the maximum inertia weight allowed in the optimization.

Acceleration Coefficients

The acceleration coefficients *C1* and *C2* are typically constant during the optimization process and can be set to different values depending on the problem characteristics and the desired convergence behavior [76,77]. For example, setting *C1* = 2.5 and *C2* = 2.5 gives equal importance to the personal best and global best positions, while setting *C1* = 1 and *C2* = 2 gives more weight to the global best position.

3.2. Model Evaluation Indicators

For a comprehensive assessment of safety risk in underground mining, three statistical metrics (coefficient of determination R^2, mean square error *MSE*, and mean absolute percentage error *MAPE*) were used to determine the average performance of the optimal model and the accuracy of the prediction. The coefficient of determination (R^2) is the percentage of variance of the dependent variable that can be predicted using the independent variables [78]. The coefficient of determination (R^2) was used to measure the effectiveness of the neural network model, and therefore the optimal model was determined based on this principle. The coefficient of determination (R^2) can have values between 0 and 1. An R^2 value closer to 1 indicates higher correlation and better agreement between the predicted results and the target values [79]. *MSE* is the mean or average square of the difference between actual and estimated values. This means that the *MSE* is obtained by dividing the square of the difference between the predicted and actual target variables by the number of data points. Positive values are always preferable, while values close to zero are optimal. The smaller the values of *MAPE* and *MSE*, the better the accuracy of the prediction model [80]. The equations for the three statistical metrics are given below.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (X_i - Y_i)^2}{\sum_{i=1}^{n} (\bar{Y} - Y_i)^2} \tag{21}$$

$$MAPE = \frac{\sum_{i=1}^{n} |X_i - Y_i|}{n} * 100 \tag{22}$$

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (X_i - Y_i)^2 \tag{23}$$

where n is the total number of the data points, X_i is the actual value of the ith sample, Y_i is the predicted value of the ith sample, and $\bar{Y}$ is the mean value of the actual values.

4. Result and Analysis

Primary and secondary data were used in carrying out this report. A questionnaire survey was used to collect the primary data from the Xiaonan coal mine company. MATLAB software (64-bit (win64), R2019a) was used to train and test the network. Of the data sets, 329 were used to train and test the performance of the optimal model, with an optimal structure of 46 × 14 × 1. In order to show the prediction accuracy of the neural network models, a comparative analysis of the error between the predicted values of the traditional BP neural network, the GA-BP neural network, and the PSO-BP neural network is implemented in Figure 7. The comparison curves between the predicted and actual values of the coal mine safety risk for each model are shown in Figure 8.

4.1. Results Analysis

Predictive accuracy ensures timely intervention and the possible prevention of accidents. Therefore, developing precision models is paramount to promoting miner and equipment safety in underground coal mines worldwide. As mentioned earlier, the BP neural network can get stuck at a local minimum when trying to find the global minimum during training. This happens because the gradient descent algorithm used in training the BP neural network tries to find the direction of the steepest descent to minimize the error function. However, it may get stuck at a local minimum instead of reaching the global minimum. In addition, the convergence speed of BPNN decreases with large data, and the real-time performance of its inhibition becomes poor. In this research, the BP neural network was used and then optimized to build a more efficient and accurate prediction model for the prediction of safety risk factors in the Xiaonan coal mine. The BP neural network was optimized with the GA neural network to form the GA-BP neural network. The BP neural network was also optimized with the PSO neural network to form the PSO-BP neural network. Therefore, the particle swarm optimization (PSO) algorithm, with its simple implementation and high convergence speed, was investigated to optimize the BP neural network. The PSO technique does not require complex encoding and decoding processes like genetic algorithms do. The real number denotes a particle whose internal velocity is updated for the optimal solution [79]. Therefore, the PSO algorithm was combined with the BP neural network to build an efficient model for predicting safety risk factors in underground coal mines. The assessment indicators for the PSO-BP neural network are shown in Table 2.

Table 2. Assessment indicators of the PSO-BP neural network.

Model No.	***MSE*** **[$\times 10^{-4}$]**		***MAPE*** **(%)**		***R*^{2}**	
	Train	Test	Train	Test	Train	Test
1	2.9	3.7	4.1	4.7	0.84	0.83
2	5.4	5.5	5.0	5.1	0.83	0.80
3	3.5	2.6	3.6	4.9	0.86	0.84
4	3.5	3.0	4.3	4.2	0.85	0.86
5	1.1	2.0	1.2	2.6	0.94	0.92
6	3.7	3.8	3.6	4.8	0.87	0.91
7	3.5	4.3	4.3	4.2	0.88	0.87
8	2.1	2.3	4.0	3.7	0.85	0.81
9	2.4	3.9	3.7	4.1	0.87	0.83
10	2.2	2.3	3.7	4.7	0.83	0.85

Ten repeated optimization models were run for the PSO-BP neural network to determine the best fitness (see Table 2). In each iteration, a new population was continuously generated, and the optimal weights and biases of the BP neural network were determined. Figure 5 shows the best global fitness of the ten models. *MSE, MAPE*, and R^2 were used as the main evaluation indicators, as shown in Table 2.

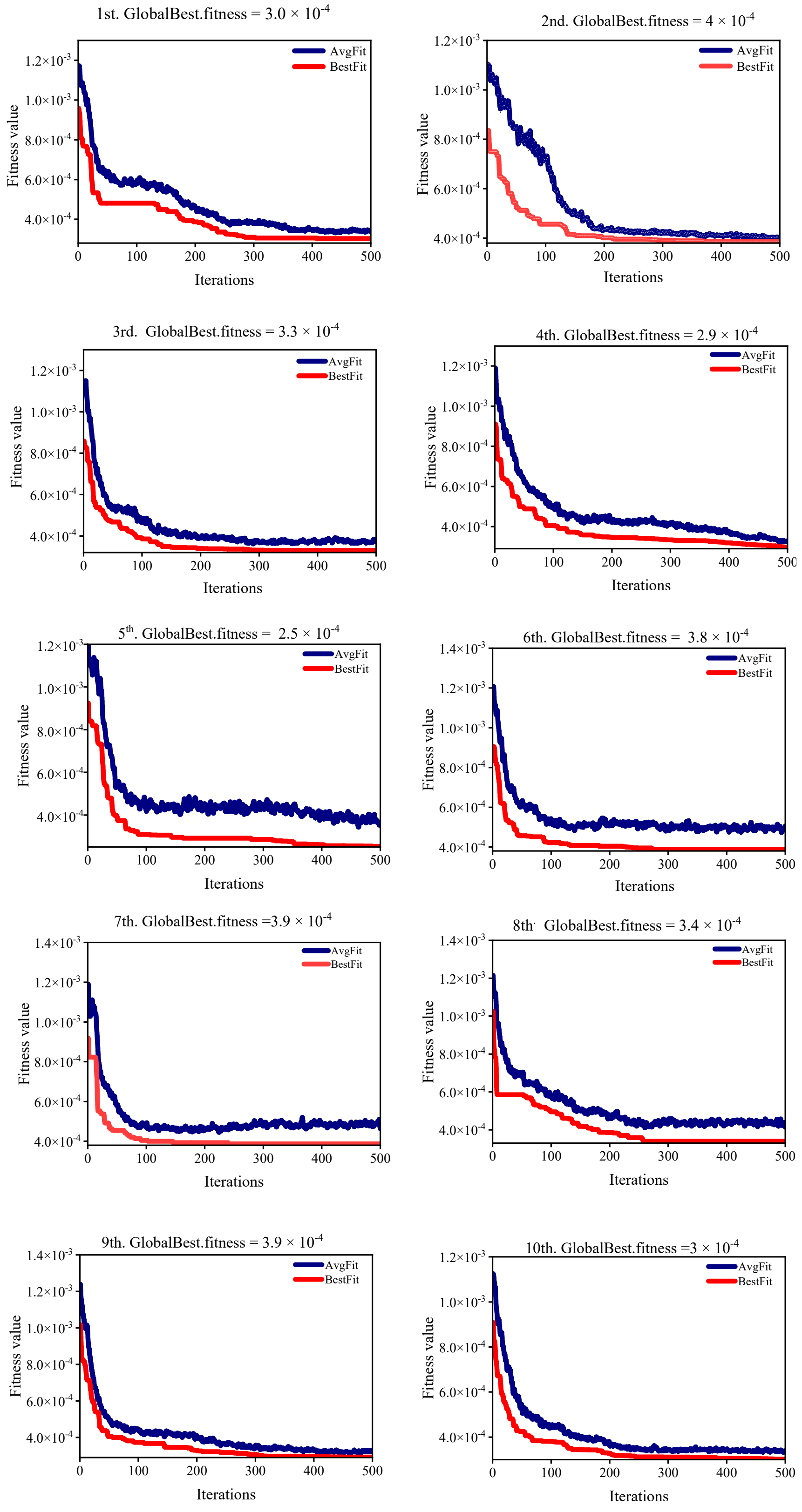

Figure 5. The evolution of the global best's fitness. Best and average fitness based on the error (MSE) of the BP neural network models 1–10.

It can be seen that the fifth model had the lowest value of 2.5 × 10^{-4}, while the second model had the highest value of 4.0 × 10^{-4} among all the ten models. This shows that the 5th model had the best predictive accuracy compared to the 2nd model, which had the highest global fitness value. This is because a lower global fitness value indicates that the model has the best prediction accuracy. In addition, it can be seen from Table 2 that the error values when testing the 5th model were 2.0 × 10^{-4}. Additionally, the mean absolute percentage value of the fifth model for testing was 2.6, which is the lowest value compared to the results obtained by the other models. On the other hand, the second model, which had the worst values for prediction accuracy, achieved an error value of 5.5 × 10^{-4} and a mean absolute percentage value of 5.1 for the testing of the data sets. Accordingly, low *MSE* and *MAPE* values indicate a high-performing model [80]. Apart from this, the results of the regression analysis for the training and testing of the fifth and second models are shown in Figure 6. The R^2 value of the testing for the fifth model was 0.92 compared to the second model, which achieved a predictive accuracy of 0.80. This shows that the fifth model has a good correlation with the actual value and is therefore favorable and reliable. Therefore, the fifth model was considered the neural network model (PSO-BP) for further analysis.

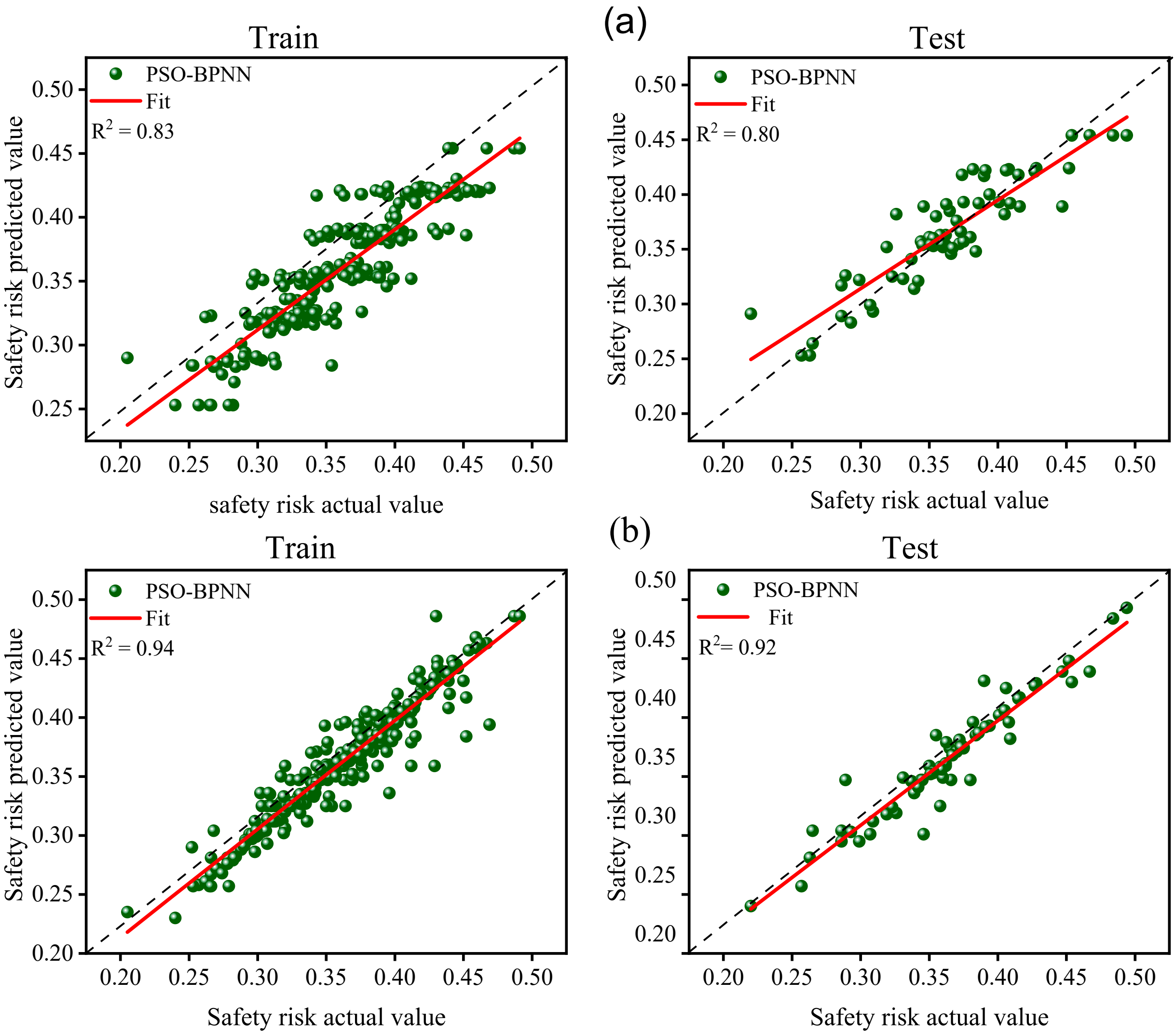

Figure 6. The regression plot and the fitted line between the actual and predicted values of the PSO-BP neural network model. (**a**) the smallest accuracy; (**b**) highest accuracy.

4.2. Comparative Analysis of Models

Three indicators were used to evaluate these models to determine their predictive accuracy. A comparison of the three models was made to determine the best model for predicting the safety risk factors in the Xiaonan coal mine. The *MSE*, *MAPE*, and R^2 values were used to compare the performance of the optimized models with that of the BP neural network, as shown in Table 3.

Table 3. Performance of each of the models.

Model	***MSE***		***MAPE* (%)**		***R*²**	
	Train	Test	Train	Test	Train	Test
BPNN	1.3×10^{-3}	1.5×10^{-3}	6	9.7	0.64	0.50
GA-BPNN	3.2×10^{-4}	4.2×10^{-4}	4.2	5.1	0.82	0.78
PSO-BPNN	1.1×10^{-4}	2.0×10^{-4}	3.1	4.3	0.94	0.92

Comparing the GA-BP neural network with the BP neural network, it can be seen that the *MSE* and *MAPE* values for the tests of the GA-BP neural network were 4.2×10^{-4} and 5.1, respectively, compared to the testing of the BP neural network, which achieved 1.5×10^{-3} and 9.7, respectively, as shown in Table 3. Lower *MSE* and *MAPE* values show more efficient prediction accuracy of the models, so the GA-BP neural network had better prediction accuracy than the BP neural network. More importantly, the R^2 value of the model GA-BP for the tests was 0.78 compared to the BP neural network model, which was 0.50. This shows that the predicted values of the GA-BP neural network model had a good correlation with the actual values compared to the BP neural network model. This makes the GA-BP neural network model more favorable and reliable than the BP neural network model [81]. Furthermore, comparing the PSO-BP neural network model with the BP neural network model, the PSO-BP neural network model turns out to be better. For example, the *MSE* and *MAPE* values for testing the PSO-BP neural network model were 2.0×10^{-4} and 4.3, respectively. The testing values of the *MSE* and *MAPE* for the BP neural network model were 1.5×10^{-3} and 9.7, respectively. In comparison, the PSO-BP neural network model had lower *MSE* and *MAPE* values, indicating improved prediction accuracy. In addition, the R^2 value of the PSO-BP neural network model for the testing was 0.92 compared with the BP neural network model, which had 0.50. Higher R^2 values indicate that the predictive accuracy of the PSO-BP neural network model had a good correlation with the actual values compared to the BP neural network model [82,83]. This shows that the PSO-BP neural network model is more favorable and reliable than the BP neural network model. Figure 7 shows the error comparison of the three models for training and testing data sets.

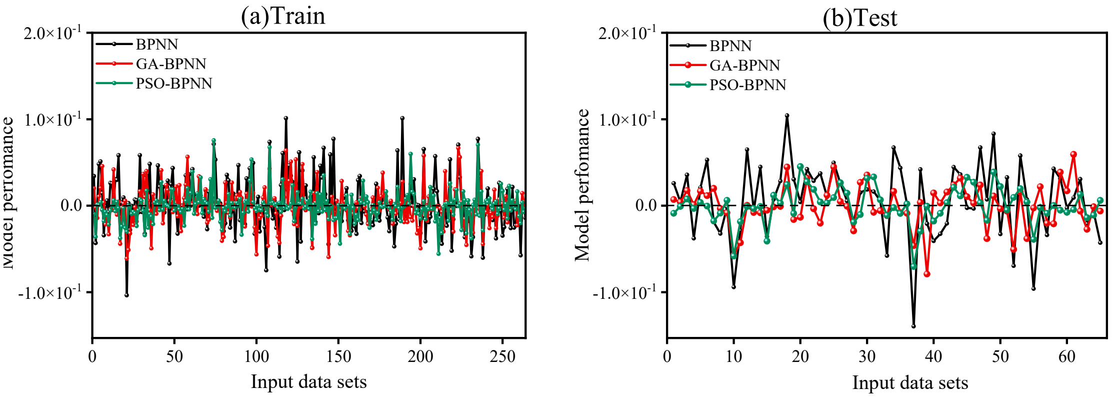

Figure 7. Error comparison between the predicted values of the BP neural network, the GA-BP neural, and the PSO-BP neural network. (**a**) Training data sets (**b**) Testing data sets.

Again, a comparison was made between the GA-BP neural network model and the PSO-BP neural network model to determine the most efficient and reliable model. The *MSE* of the testing for the GA-BP neural network model was 4.2×10^{-4}, whereas the PSO-BP neural network model was 2.0×10^{-4}. The *MAPE* value of the testing for the GA-BP neural network model was 5.1, whereas the PSO-BP neural network model was 4.3%. Smaller *MSE* and *MAPE* values indicate a better-performing model, so the PSO-BP neural network model has higher prediction accuracy than the GA-BP neural network model. In addition, the R^2 value of the testing for the GA-BP neural network model reached a value of 0.78 compared to the PSO-BP neural network, which was 0.92. This also indicates that the output of the PSO-BP neural network model has a good correlation with the actual values compared to the GA-BP neural network model, as a higher R^2 value indicates a model with high precision. The prediction improvement of the PSO-BP neural network model over the BP neural network model was 85.2% compared to the GA-BP neural network model, which was 65.7%, indicates that the PSO-BP neural network model has a better prediction improvement in comparison, as indicated in Table 4.

Table 4. Models' prediction improvement over the traditional BP neural network.

Model Prediction Improvement		
	Train	Test
GA-BPNN	58.9%	65.7%
PSO-BPNN	89.3%	85.2%

Figure 8 shows a comparison curve between the predicted values and the actual values of the BP, GA-BP, and PSO-BP neural network models. From Figure 8c, it can be seen that the predicted values of the PSO-BP neural network almost match the actual values, making it the best prediction model. The analysis of the *MSE*, *MAPE*, R^2, and global best fitness shows that the PSO-BP neural network model has the highest prediction accuracy. Therefore, in this paper, the PSO-BP neural network model is proposed as an improved model for the prediction and assessment of safety risk factors in underground mines. This study is in line with other studies that have also proposed the PSO-BP neural network model for the assessment and prediction of various issues. The PSO-BP neural network model has better robustness and accuracy in the assessment and prediction of risk factors compared to the other models [84]. Furthermore, in addition to having a simple algorithm, the PSO-BP neural network model is highly evaluative and intelligent [85]. For example, Deng et al. proposed a model for the prediction of the number of coliform bacteria in Dai's Special Snacks, Sapie, based on the PSO-BP neural network model. Ma et al. also proposed a thermal error model of the spindle system based on the PSO-BP neural network.

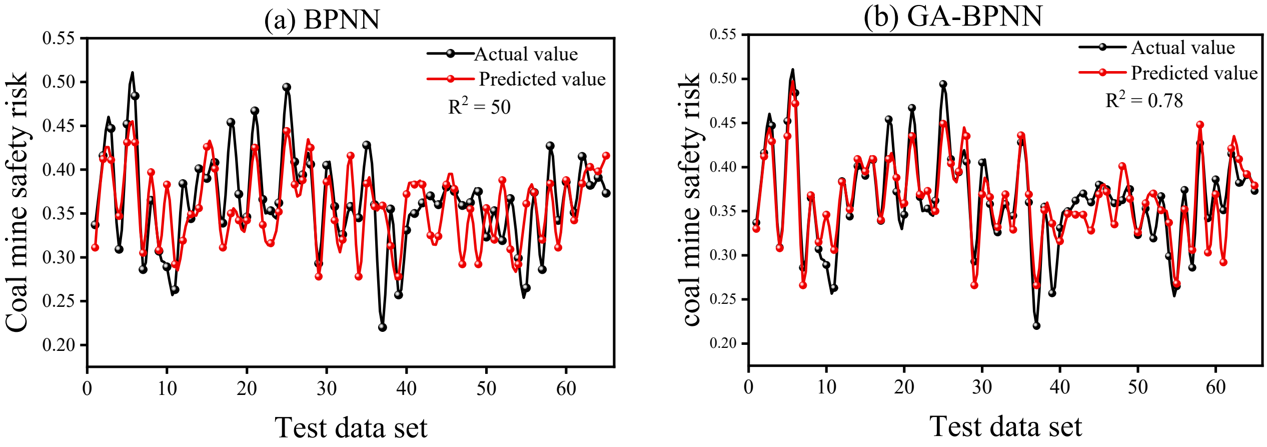

Figure 8. *Cont.*

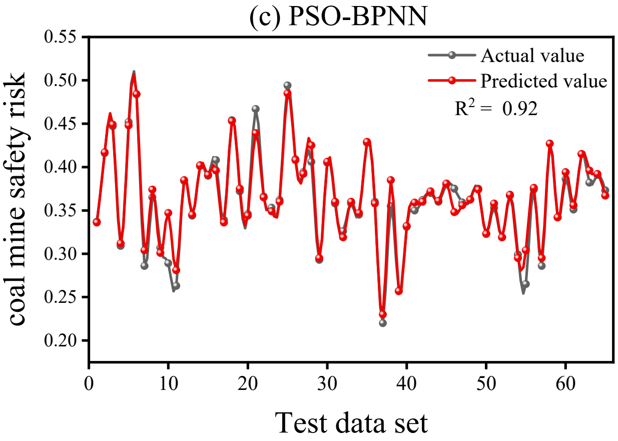

Figure 8. Comparison curves between the predicted and actual values of each model. (**a**) BP neural network, (**b**) GA-BP neural network and (**c**) PSO-BP neural network.

4.3. Limitations of the Study

It is important to note that this study, like any other, has some limitations. For example, it was difficult to collect the data for this study because the responses to the questionnaires took a long time to come in. Furthermore, compiling the data was a tedious process that took a lot of time. We believe that minimal errors could have occurred at any of these stages that would have been reflected in the results. Again, we believe that the parameters used in the evaluation of the neural network proposed in this study for PSO-BP did not produce the best result due to these minimal errors. Nonetheless, we believe that these errors are minor and do not affect the overall conclusions of this study.

5. Conclusions

Coal has played an important role in supplying energy to the economies of many countries, but it also poses various risk factors to miners, equipment, and others. Risk assessment and identification are very important, as unrecognized risks can lead to accidents resulting in minor to severe injuries or even fatalities. Accurate risk assessment provides reliable information for taking timely safety measures. To this end, a total of 329 datasets were gathered from the Xiaonan coal mine to aid in the development of a novel and efficient model for the assessment and prediction of risk factors. A novel approach to underground coal mine assessment (PSO-BP) that combines particle swarm optimization (PSO) and BP neural networks has been proposed for evaluating and predicting safety risk factors in underground coal mines. Three indicators including, the *MSE*, *MAPE*, and R^2, were used to evaluate the models, and a comparison was made between the PSO-BP, BP, and GA-BP models. The values of the *MSE*, *MAPE*, and R^2 of the testing for the PSO-BP neural network were 2×10^{-4}, 4.3, and 0.92, respectively, which were the best-performing values compared to the other models. Therefore, the study proposed the PSO-BP neural network model as an effective model to evaluate the safety risk of underground coal mining.

Author Contributions: Conceptualization, J.L. and D.M.M.; methodology, J.L.; resources, J.H.; data curation, D.M.M.; writing—original draft preparation, D.M.M.; writing—review and editing, J.L. and J.H.; visualization, D.M.M.; supervision, J.L.; Y.Z.: Writing—review and editing, Project administration; H.L.: Writing—original draft, Project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of China (Grant No.52204099 and Grant No.52174121) and the Natural Science Foundation of Shandong Province, China (Grant No. ZR2022QE203).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The corresponding author can provide data supporting the findings of this study upon request. Due to ethical or privacy concerns, the data are not publicly available.

Acknowledgments: Thanks to Jiankang Liu and Jian Hao, who provided the main ideas and financial support for the research. Dorcas Muadi Mulumba is responsible for compiling and writing the articles. Thanks to Yining Zheng and Heqing Liu from the Shandong University of Science and Technology.

Conflicts of Interest: The authors declare they have no known competing financial interest or personal relationship that could have appeared to influence the work reported in this paper.

Abbreviations

O_K	The output vector of the network in the k-ith layer
net_k	The summation weighted at the output layer k
W_{jk}	The weight of hidden layer j and output layer k
y_j	The output of the hidden layer j
x_i	The input at the nodes in layer i
v_{ij}	The weight of the input layer and hidden layer
net_j	the summation of the weighted input
w_{jk}	The transfer function in the jth layer node
E	The neural network error
$-\eta$	The learning constant
δ_k^O	The error signal for the output layer O and hidden layer k
Δw_{jk}	Deviation error of the weight in the hidden layer j and output layer k
Δv_{iij}	Derivation error of the weight in input layer I and hidden layer j
X_i	The position of the ith particle
V_i	The velocity of the particle
$C1$	The personal learning coefficient
$C2$	The global learning coefficient
W	The inertia weight parameter in the PSO algorithm
X_{norm}	Normalized data
R^2	Coefficient of determination
$MAPE$	Mean absolute percentage error
MSE	Mean squared error

References

1. Paul, P.S.; Maiti, J. The role of behavioral factors on safety management in underground mines. *Saf. Sci.* **2007**, *45*, 449–471. [CrossRef]
2. Senapati, A.; Bhattacherjee, A.; Chatterjee, S. Causal relationship of some personal and impersonal variates to occupational injuries at continuous miner worksites in underground coal mines. *Saf. Sci.* **2022**, *146*, 105562. [CrossRef]
3. Petsonk, E.L.; Rose, C.; Cohen, R. Coal mine dust lung disease. New lessons from an old exposure. *Am. J. Respir. Crit. Care Med.* **2013**, *187*, 1178–1185. [CrossRef]
4. Sovacool, B.K. The costs of failure: A preliminary assessment of major energy accidents, 1907–2007. *Energy Policy* **2008**, *36*, 1802–1820. [CrossRef]
5. Li, S.; You, M.; Li, D.; Liu, J. Identifying coal mine safety production risk factors by employing text mining and Bayesian network techniques. *Process Saf. Environ. Prot.* **2022**, *162*, 1067–1081. [CrossRef]
6. Wang, D.; Sui, W.; Ranville, J.F. Hazard identification and risk assessment of groundwater inrush from a coal mine: A review. *Bull. Eng. Geol. Environ.* **2022**, *81*, 421. [CrossRef]
7. Tong, R.; Yang, Y.; Ma, X.; Zhang, Y.; Li, S.; Yang, H. Risk assessment of Miners' unsafe behaviors: A case study of gas explosion accidents in coal mine, china. *Int. J. Environ. Res. Public Health* **2019**, *16*, 1765. [CrossRef] [PubMed]
8. Kharzi, R.; Chaib, R.; Verzea, I.; Akni, A. A Safe and Sustainable Development in a Hygiene and Healthy Company Using Decision Matrix Risk Assessment Technique: A case study. *J. Min. Environ.* **2020**, *11*, 363–373.
9. Hassanien, A.E.; Darwish, A.; Abdelghafar, S. Machine learning in telemetry data mining of space mission: Basics, challenging and future directions. *Artif. Intell. Rev.* **2020**, *53*, 3201–3230. [CrossRef]
10. Ayvaz, S.; Alpay, K. Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. *Expert Syst. Appl.* **2021**, *173*, 114598. [CrossRef]
11. Kashyap, R. Geospatial Big Data, Analytics and IoT: Challenges, Applications and Potential. In *Cloud Computing for Geospatial Big Data Analytics: Intelligent Edge, Fog and Mist Computing*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 191–213.

12. Manzoor, U.; Ehsan, M.; Radwan, A.E.; Hussain, M.; Iftikhar, M.K.; Arshad, F. Seismic driven reservoir classification using advanced machine learning algorithms: A case study from the lower Ranikot/Khadro sandstone gas reservoir, Kirthar fold belt, lower Indus Basin, Pakistan. *Geoenergy Sci. Eng.* **2023**, *222*, 211451. [CrossRef]
13. Sahu, A.; Mishra, D.P. Coal mine explosions in India: Management failure, safety lapses and mitigative measures. *Extr. Ind. Soc.* **2023**, *14*, 101233. [CrossRef]
14. Zheng, Z.; Wang, F.; Gong, G.; Yang, H.; Han, D. Intelligent technologies for construction machinery using data-driven methods. *Autom. Constr.* **2023**, *147*, 104711. [CrossRef]
15. Kudashkina, K.; Corradini, M.G.; Thirunathan, P.; Yada, R.Y.; Fraser, E.D. Artificial Intelligence technology in food safety: A behavioral approach. *Trends Food Sci. Technol.* **2022**, *123*, 36–38. [CrossRef]
16. Sadeghi, S.; Soltanmohammadlou, N.; Nasirzadeh, F. Applications of wireless sensor networks to improve occupational safety and health in underground mines. *J. Saf. Res.* **2022**, *83*, 8–25. [CrossRef] [PubMed]
17. Ali, M.H.; Al-Azzawi, W.K.; Jaber, M.; Abd, S.K.; Alkhayyat, A.; Rasool, Z.I. Improving coal mine safety with internet of things (IoT) based Dynamic Sensor Information Control System. *Phys. Chem. Earth Parts A/B/C* **2022**, *128*, 103225. [CrossRef]
18. Bai, G.; Xu, T. Coal mine safety evaluation based on machine learning: A BP neural network model. *Comput. Intell. Neurosci.* **2022**, *2022*, 5233845. [CrossRef]
19. Cruz, I.A.; Chuenchart, W.; Long, F.; Surendra, K.; Andrade, L.R.S.; Bilal, M.; Liu, H.; Figueiredo, R.T.; Khanal, S.K.; Ferreira, L.F.R. Application of machine learning in anaerobic digestion: Perspectives and challenges. *Bioresour. Technol.* **2022**, *345*, 126433. [CrossRef]
20. Agatonovic-Kustrin, S.; Beresford, R. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *J. Pharm. Biomed. Anal.* **2000**, *22*, 717–727. [CrossRef]
21. Wu, Y.; Gao, R.; Yang, J. Prediction of coal and gas outburst: A method based on the BP neural network optimized by GASA. *Process Saf. Environ. Prot.* **2020**, *133*, 64–72. [CrossRef]
22. Zheng, Y.; Zhong, H.; Fang, Y.; Zhang, W.; Liu, K.; Fang, J. Rockburst prediction model based on entropy weight integrated with grey relational BP neural network. *Adv. Civ. Eng.* **2019**, *2019*, 34. [CrossRef]
23. Qi, S.; Jin, K.; Li, B.; Qian, Y. The exploration of internet finance by using neural network. *J. Comput. Appl. Math.* **2020**, *369*, 112630. [CrossRef]
24. Chong, H.Y.; Yap, H.J.; Tan, S.C.; Yap, K.S.; Wong, S.Y. Advances of metaheuristic algorithms in training neural networks for industrial applications. *Soft Comput.* **2021**, *25*, 11209–11233. [CrossRef]
25. Zhong, K.; Wang, Y.; Pei, J.; Tang, S.; Han, Z. Super efficiency SBM-DEA and neural network for performance evaluation. *Inf. Process. Manag.* **2021**, *58*, 102728. [CrossRef]
26. Yang, L.; Birhane, G.E.; Zhu, J.; Geng, J. Mining employees safety and the application of information technology in coal mining. *Front. Public Health* **2021**, *9*, 709987. [CrossRef]
27. Chen, J.; Huang, S. Evaluation model of green supply chain cooperation credit based on BP neural network. *Neural Comput. Appl.* **2021**, *33*, 1007–1015. [CrossRef]
28. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning Representations by Backpropagating Errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
29. López-Monroy, A.P.; García-Salinas, J.S. Neural networks and deep learning. In *Biosignal Processing and Classification Using Computational Learning and Intelligence*; Academic Press: Cambridge, MA, USA, 2022; pp. 177–196.
30. Jana, D.K.; Bhunia, P.; Adhikary, S.D.; Bej, B. Optimization of effluents using artificial neural network and support vector regression in detergent industrial wastewater treatment. *Clean. Chem. Eng.* **2022**, *3*, 100039. [CrossRef]
31. Shen, S.L.; Elbaz, K.; Shaban, W.M.; Zhou, A. Real-time prediction of shield moving trajectory during tunnelling. *Acta Geotech.* **2022**, *17*, 1533–1549. [CrossRef]
32. Hosseini, V.R.; Mehrizi, A.A.; Gungor, A.; Afrouzi, H.H. Application of a physics-informed neural network to solve the steady-state Bratu equation arising from solid biofuel combustion theory. *Fuel* **2023**, *332*, 125908. [CrossRef]
33. Saeed, A.; Li, C.; Gan, Z.; Xie, Y.; Liu, F. A simple approach for short-term wind speed interval prediction based on independently recurrent neural networks and error probability distribution. *Energy* **2022**, *238*, 122012. [CrossRef]
34. Zhang, S.Z.; Chen, S.; Jiang, H. A back propagation neural network model for accurately predicting the removal efficiency of ammonia nitrogen in wastewater treatment plants using different biological processes. *Water Res.* **2022**, *222*, 118908. [CrossRef]
35. Feng, C.; Zhang, J.; Zhang, W.; Hodge, B.M. Convolutional neural networks for intra-hour solar forecasting based on sky image sequences. *Appl. Energy* **2022**, *310*, 118438. [CrossRef]
36. Rodzin, S.; Bova, V.; Kravchenko, Y.; Rodzina, L. Deep Learning Techniques for Natural Language Processing. In *Artificial Intelligence Trends in Systems, Proceedings of the 11th Computer Science On-line Conference, July 2022*; Springer International Publishing: Cham, Switzerland, 2022; Volume 2, pp. 121–130.
37. Wen, T.; Xiao, Y.; Wang, A.; Wang, H. A novel hybrid feature fusion model for detecting phishing scam on Ethereum using deep neural network. *Expert Syst. Appl.* **2023**, *211*, 118463. [CrossRef]
38. Rajawat, A.S.; Jain, S. Fusion deep learning based on back propagation neural network for personalization. In Proceedings of the 2nd International Conference on Data, Engineering and Applications (IDEA), Bhopal, India, 28–29 February 2020; IEEE: New York, NY, USA, 2010; pp. 1–7.

39. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
40. Pontani, M.; Conway, B.A. Particle swarm optimization applied to space trajectories. *J. Guid. Control Dyn.* **2010**, *33*, 1429–1441. [CrossRef]
41. Jain, M.; Saihjpal, V.; Singh, N.; Singh, S.B. An Overview of Variants and Advancements of PSO Algorithm. *Appl. Sci.* **2022**, *12*, 8392. [CrossRef]
42. Fallahi, S.; Taghadosi, M. Quantum-behaved particle swarm optimization based on solitons. *Sci. Rep.* **2022**, *12*, 13977. [CrossRef]
43. Arrif, T.; Hassani, S.; Guermoui, M.; Sánchez-González, A.; Taylor, R.A.; Belaid, A. GA-GOA hybrid algorithm and comparative study of different metaheuristic population-based algorithms for solar tower heliostat field design. *Renew. Energy* **2022**, *192*, 745–758. [CrossRef]
44. Punyakum, V.; Sethanan, K.; Nitisiri, K.; Pitakaso, R.; Gen, M. Hybrid differential evolution and particle swarm optimization for Multi-visit and Multi-period workforce scheduling and routing problems. *Comput. Electron. Agric.* **2022**, *197*, 106929. [CrossRef]
45. Mousavirad, S.J.; Rahnamayan, S. CenPSO: A novel center-based particle swarm optimization algorithm for large-scale optimization. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020; pp. 2066–2071.
46. Lv, Z.; Wang, L.; Han, Z.; Zhao, J.; Wang, W. Surrogate-assisted particle swarm optimization algorithm with Pareto active learning for expensive multi-objective optimization. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 838–849. [CrossRef]
47. Premkumar, M.; Jangir, P.; Sowmya, R.; Alhelou, H.H.; Heidari, A.A.; Chen, H. MOSMA: Multi-objective slime mould algorithm based on elitist non-dominated sorting. *IEEE Access* **2020**, *9*, 3229–3248. [CrossRef]
48. Naji, H.R.; Shadravan, S.; Jafarabadi, H.M.; Momeni, H. Accelerating sailfish optimization applied to unconstrained optimization problems on graphical processing unit. *Eng. Sci. Technol. Int. J.* **2022**, *32*, 101077.
49. Robinson, E.; Sutin, A.R.; Daly, M.; Jones, A. A systematic review and meta-analysis of longitudinal cohort studies comparing mental health before versus during the COVID-19 pandemic in 2020. *J. Affect. Disord.* **2022**, *296*, 567–576. [CrossRef] [PubMed]
50. Marini, F.; Walczak, B. Particle swarm optimization (PSO). A tutorial. *Chemom. Intell. Lab. Syst.* **2015**, *149*, 153–165. [CrossRef]
51. Rahnamayan, S.; Wang, G.G. Toward effective initialization for large-scale search spaces. *Trans Syst.* **2009**, *8*, 355–367.
52. Khare, A.; Rangnekar, S. A review of particle swarm optimization and its applications in solar photovoltaic system. *Appl. Soft Comput.* **2013**, *13*, 2997–3006. [CrossRef]
53. Sigarchian, S.G.; Orosz, M.S.; Hemond, H.F.; Malmquist, A. Optimum design of a hybrid PV–CSP–LPG microgrid with Particle Swarm Optimization technique. *Appl. Therm. Eng.* **2016**, *109*, 1031–1036. [CrossRef]
54. Han, W.; Yang, P.; Ren, H.; Sun, J. Comparison study of several kinds of inertia weights for PSO. In Proceedings of the 2010 IEEE International Conference on Progress in Informatics and Computing, Shanghai, China, 1–12 December 2010; IEEE: New York, NY, USA, 2010; Volume 1, pp. 280–284.
55. Liang, J.J.; Qin, A.K.; Suganthan, P.N.; Baskar, S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evol. Comput.* **2006**, *10*, 281–295. [CrossRef]
56. Jiao, B.; Lian, Z.; Gu, X. A dynamic inertia weight particle swarm optimization algorithm. *Chaos Solitons Fractals* **2008**, *37*, 698–705. [CrossRef]
57. Eberhart, R.C.; Shi, Y. Comparison between genetic algorithms and particle swarm optimization. In Proceedings of the Evolutionary Programming VII: 7th International Conference, EP98, San Diego, CA, USA, 25–27 March 1998; Springer: Berlin/Heidelberg, Germany, 1998; pp. 611–616.
58. Parsopoulos, K.E.; Vrahatis, M.N. Recent approaches to global optimization problems through particle swarm optimization. *Nat. Comput.* **2002**, *1*, 235–306. [CrossRef]
59. Clerc, M.; Kennedy, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73. [CrossRef]
60. Deng, G.-F.; Lin, W.-T.; Lo, C.-C. Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization. *Expert Syst. Appl.* **2012**, *39*, 4558–4566. [CrossRef]
61. Trivedi, V.; Varshney, P.; Ramteke, M. A simplified multi-objective particle swarm optimization algorithm. *Swarm Intell.* **2020**, *14*, 83–116. [CrossRef]
62. Singh, N.; Singh, S.B.; Houssein, E.H. Hybridizing salp swarm algorithm with particle swarm optimization algorithm for recent optimization functions. *Evol. Intell.* **2022**, *15*, 1–34. [CrossRef]
63. Zhang, M.; Liu, D.; Wang, Q.; Zhao, B.; Bai, O.; Sun, J. Detection of alertness-related EEG signals based on decision fused BP neural network. *Biomed. Signal Process. Control* **2022**, *74*, 103479. [CrossRef]
64. Yu, F.; Xu, X. A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network. *Appl. Energy* **2014**, *134*, 102–113. [CrossRef]
65. Ghaffari, A.; Abdollahi, H.; Khoshayand, M.R.; Bozchalooi, I.S.; Dadgar, A.; Rafiee-Tehrani, M. Performance comparison of neural network training algorithms in modeling of bimodal drug delivery. *Int. J. Pharm.* **2006**, *327*, 126–138. [CrossRef]
66. Rodger, J.A. A fuzzy nearest neighbor neural network statistical model for predicting demand for natural gas and energy cost savings in public buildings. *Expert Syst. Appl.* **2014**, *41*, 1813–1829. [CrossRef]
67. Ren, C.; An, N.; Wang, J.; Li, L.; Hu, B.; Shang, D. Optimal parameters selection for BP neural network based on particle swarm optimization: A case study of wind speed forecasting. *Knowl. Based Syst.* **2014**, *56*, 226–239. [CrossRef]

68. Singh, P.; Dwivedi, P. Integration of new evolutionary approach with artificial neural network for solving short term load forecast problem. *Appl. Energy* **2018**, *217*, 537–549. [CrossRef]
69. Lage, P.L.C. An analytical solution to the population balance equation with coalescence and breakage-the special case with constant number of particles. *Chem. Eng. Sci.* **2002**, *53*, 599–601.
70. Shi, X.H.; Liang, Y.C.; Lee, H.P.; Lu, C.; Wang, L.M. An improved GA and a novel PSO-GA-based hybrid algorithm. *Inf. Process. Lett.* **2005**, *93*, 255–261. [CrossRef]
71. Adam, P.P.; Napiorkowski, J.J.; Piotrowska, A.E. Population size in particle swarm optimization. *Swarm Evol. Comput.* **2020**, *58*, 100718.
72. Zhang, H.; Li, H.; Tam, C.M. Particle swarm optimization for resource-constrained project scheduling. *Int. J. Proj. Manag.* **2006**, *24*, 83–92. [CrossRef]
73. Lobo, F.G.; Lima, C.F. Adaptive Population Sizing Schemes in Genetic Algorithms. *Parameter Setting Evol. Algorithms* **2007**, *54*, 185–204.
74. Kentzoglanakis, K.; Poole, M. Particle swarm optimization with an oscillating inertia weight. In Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, Montreal, QC, Canada, 8–12 July 2009; pp. 1749–1750.
75. He, M.; Liu, M.; Wang, R.; Jiang, X.; Liu, B.; Zhou, H. Particle swarm optimization with damping factor and cooperative mechanism. *Appl. Soft Comput.* **2019**, *76*, 45–52. [CrossRef]
76. Rao, R.V.; Pawar, P.J.; Shankar, R. Multi-objective optimization of electrochemical machining process parameters using a particle swarm optimization algorithm. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2008**, *222*, 949–958. [CrossRef]
77. Song, Y.; Chen, Z.; Yuan, Z. New chaotic PSO-based neural network predictive control for nonlinear process. *IEEE Trans. Neural Netw.* **2007**, *18*, 595–601. [CrossRef]
78. Gogtay, N.J.; Thatte, U.M. Principles of correlation analysis. *J. Assoc. Physicians India* **2017**, *65*, 78–81. [PubMed]
79. Wang, Z.; Zhang, J.; Wang, J.; He, X.; Fu, L.; Tian, F.; Liu, X.; Zhao, Y. A Back Propagation neural network based optimizing model of space-based large mirror structure. *Optik* **2019**, *179*, 780–786. [CrossRef]
80. Davide, C.; Warrens, M.J.; Jurman, G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Comput. Sci.* **2021**, *7*, e623.
81. Zhu, C.; Zhang, J.; Liu, Y.; Ma, D.; Li, M.; Xiang, B. Comparison of GA-BP and PSO-BP neural network models with initial BP model for rainfall-induced landslides risk assessment in regional scale: A case study in Sichuan, China. *Nat. Hazards* **2020**, *100*, 173–204. [CrossRef]
82. Ma, C.; Zhao, L.; Mei, X.; Shi, H.; Yang, J. Thermal error compensation of high-speed spindle system based on a modified BP neural network. *Int. J. Adv. Manuf. Technol.* **2017**, *89*, 3071–3085. [CrossRef]
83. Deng, Y.; Xiao, H.; Xu, J.; Wang, H. Prediction model of PSO-BP neural network on coliform amount in special food. *Saudi J. Biol. Sci.* **2019**, *26*, 1154–1160. [CrossRef] [PubMed]
84. Lin, S.W.; Chen, S.C.; Wu, W.J.; Chen, C.H. Parameter determination and feature selection for back-propagation network by particle swarm optimization. *Knowl. Inf. Syst.* **2009**, *21*, 249–266. [CrossRef]
85. Jiang, L.; Wang, X. Optimization of online teaching quality evaluation model based on hierarchical PSO-BP neural network. *Complexity* **2020**, *7*, 1–12. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.