applied sciences

MDPI

Article

Soil Heavy-Metal Pollution Prediction Methods Based on Two Improved Neural Network Models

Zhangang Wang [1,2,3,*], Wenshuai Zhang [1] and Yunshan He [1]

[1] School of Information and Communication Engineering, Beijing Information Science and Technology University, Beijing 100101, China
[2] Key Laboratory of Information and Communication Systems, Ministry of Information Industry, Beijing Information Science and Technology University, Beijing 100101, China
[3] Key Laboratory of the Ministry of Education for Optoelectronic Measurement Technology and Instrument, Beijing Information Science and Technology University, Beijing 100101, China
* Correspondence: wangzg@bistu.edu.cn

Abstract: Current soil pollution prediction methods need improvement, especially with regard to accuracy in supplementing missing heavy-metal values in soil, and the accuracy and slow convergence speed of methods for predicting heavy-metal content at unknown points. To reduce costs and improve prediction accuracy, this study used two neural network models (SA-FOA-BP and SE-GCN) to supplement missing heavy-metal values and efficiently predict heavy-metal content in soil. The SA-FOA-BP model combines simulated annealing and fruit fly algorithms to optimize the parameter search method in traditional BP neural networks and improve prediction of missing heavy-metal values in soil. A spatial information fusion graph convolutional network prediction model (SE-GCN) constructs a spatial information encoder that can perceive spatial context information, and embeds it with spatial autocorrelation used for auxiliary learning to predict the heavy-metal content in soil. From the experimental results, the SE-GCN model demonstrates improved performance in terms of evaluation indicators compared with other models. Application analysis of the two improved neural network models was conducted; application scenarios and suitability were analyzed, showing that these models have practical application value for soil pollution prediction.

Keywords: BP neural network; graph convolutional neural network; heavy metal in soil; missing value prediction; content prediction

check for updates

Citation: Wang, Z.; Zhang, W.; He, Y. Soil Heavy-Metal Pollution Prediction Methods Based on Two Improved Neural Network Models. *Appl. Sci.* **2023**, *13*, 11647. https://doi.org/10.3390/app132111647

Academic Editor: Douglas O'Shaughnessy

Received: 24 September 2023
Revised: 18 October 2023
Accepted: 23 October 2023
Published: 25 October 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Heavy-metal pollution in soil threatens human health and severely damages the human ecological environment [1]. The regional soil heavy-metal pollution detection system in China is constantly improving, developing toward "digitization", "precision", and "intelligence" [2]. A high-level mathematical model for the analysis of heavy-metal pollution in soil is required.

There is a relatively complete set of mathematical analysis methods for ecological environments, including correlation analysis, principal component analysis, and regression analysis in statistical analysis, and classification and clustering methods in data mining [3]. However, the premise for validating these methods is the availability of complete sample data. In soil sample collection, missing sample data can often result from outdated sampling equipment, contaminated samples, and other issues. Collecting and analyzing soil samples is costly and time-consuming, making resampling impractical. Ecological environment researchers have been seeking an effective solution for missing values [4]; however, there is currently no mature method to address missing values. Traditional missing soil value processing methods are too simple, using only mean filling, mode filling, and ignoring tuples to interpolate missing data without considering the correlation between soil sample data, with poor interpolation results for missing values.

Soil heavy-metal pollution control and restoration have long cycles and high costs; thus, current soil prevention and control measures focus mainly on prevention; restoration is auxiliary. There is limited research on soil pollution prediction; most research is based on time-series water and air pollution prediction [5,6]. Unlike water and air pollution, heavy metals have a strong sedimentary type in soil; soil heavy-metal content does not change significantly without human intervention over a long period of time. Thus, it is not feasible to predict soil heavy-metal pollution content using time-series prediction methods. There have been many studies worldwide on qualitative analysis, prevention and control, and soil restoration with regard to soil pollution; however, there have been few studies on the quantitative analysis of soil pollution. It is becoming extremely important to accurately predict the heavy-metal content in soil.

The main contributions of this study are as follows:

(1) A new SA-FOA-BP neural network model was constructed to predict missing values of heavy metals in soil. This algorithm model primarily combines the simulated annealing algorithm with the fruit fly optimization algorithm to optimize the parameters in place of traditional methods for parameter optimization in the BP neural network, thereby addressing the shortcomings of the traditional BP neural network.
(2) A spatial information fusion graph convolutional network prediction model, SE-GCN, was proposed. It establishes a spatial information encoder capable of perceiving spatial contextual information and embeds it with spatial autocorrelation, serving as auxiliary learning to predict the heavy-metal content in soil.
(3) The comparative experiments were conducted on the prediction of missing values and contents of heavy metals in soil using two neural network models, and their application scenarios and applicability were analyzed. The experiment showed that, compared to the traditional BP neural network, the SA-FOA-BP neural network model demonstrated improvements in error evaluation metrics, indicating that SA-FOA effectively optimized the parameters. The SE-GCN model achieved more accurate predictions of soil heavy-metal contents compared to existing methods, suggesting that the spatial encoder effectively extracted spatial information.

2. Literature Review

2.1. Research on Missing Value Prediction

Researchers have proposed methods to predict missing data [7]. Effective missing value prediction methods can make data more complete, which is helpful in subsequent data analysis. Statistical and machine learning methods are primarily used for missing-value predictions.

Based on statistical methods, Kayid used the maximum likelihood method and expectation maximization algorithm to predict the parameters for complete and right-censored data [8]. Bashir et al. proposed a vector autoregression model that integrated minimizing expectation and prediction error to predict missing data. However, this method did not fully consider the category of the data, leading to large differences between the predicted and actual values [9]. Gondara constructed a super-complete deep denoising autoencoder for multiple missing data imputations; it required that the proportion of missing data was not too high. Better prediction results can only be achieved with complete data [10].

Another method was based on machine learning. Najwa et al. used several machine learning models, including two support vector machines (SVM), six regression models, and artificial neural networks (ANN) to predict various water quality parameters of the Langat River in Malaysia. Based on the model performance metrics, the ANN model is superior to other models, while the SVM model exhibits overfitting characteristics [11]. Researchers have also combined denoising autoencoders with generative adversarial networks to process industrial IoT data with high missing rates; however, such models cannot produce missing values with high accuracy [12].

The problem of missing heavy-metal data in soil is serious; statistical methods are commonly used for missing-value imputation. Although statistical methods are relatively

simple, there are shortcomings and limitations to using statistical methods for missing-value imputation. Although current machine learning-based missing-value prediction methods can accurately predict missing values, feature extraction between the data is insufficient, and the prediction accuracy can be improved. Predicting missing values in soil heavy-metal data can make the data more complete, improve data utilization, and provide technical support for subsequent data analysis. Thus, an efficient and accurate method for producing the missing values in soil pollution data is necessary.

2.2. Prediction of Soil Heavy-Metal Content

In the initial stages of soil pollution prediction research, sample data were often interpolated; the prediction accuracy depended on the number of sampling points. With insufficient sampling data, the prediction accuracy is low. With a complex soil sampling environment and high cost, sampling and analysis of soil heavy-metal content often require significant financial, material, and human resources, and the amount and dimensions of the data are limited. Thus, efficient use of limited soil data is critical in soil pollution prediction research. The rapid development of machine learning has attracted widespread attention; machine learning methods are used to predict soil heavy-metal content.

Cao W used a collaborative composite neural network model using wavelet neural network (WNN) as the basic prediction model to predict the heavy-metal content in soil; at the same time, they proposed a parallel bird swarm algorithm (PBSA) to solve the parameter optimization problem of WNN. The experiment showed that the model could effectively predict the heavy-metal content [13]. Yin G proposed a method based on the genetic algorithm and neural network model that integrated soil properties and environmental factors to predict the soil heavy-metal content [14]. With the correlation analysis of heavy-metal content and pretreatment spectral band, the models using Support Vector Machine (SVM), Random Forest (RF), and Extreme Learning Machine (ELM), respectively, are constructed to predict the content of heavy metal in soil. The results show that the prediction results of machine learning methods are better than PLS [15]. Gao et al. used a regression kriging method based on natural and anthropogenic factors to predict the Cd content of heavy metals in soil at different locations and analyzed the prediction accuracy of the model [16].

Although machine learning methods can provide good ideas for predicting soil heavy-metal content, such models have problems including insufficient accuracy and slow convergence speed when predicting pollution. Insufficient consideration of spatial correlation attributes leads to less-than-ideal prediction results. More accurate and efficient prediction models are necessary to address the problems associated with heavy-metal content prediction.

3. Predicting Missing Soil Heavy-Metal Values Based on SA-FOA-BP

Incomplete or incorrect soil sample data in collection or laboratory sample analysis is problematic. Simply deleting these data can reduce the experimental accuracy. Because there is a correlation between heavy metals in soil samples, interpolating missing data through correlation algorithms can ensure the accuracy of soil heavy-metal data prediction. In this study, the simulated annealing algorithm was improved by the fruit fly optimization algorithm to optimize the parameters of the BP neural network, improve the prediction accuracy of missing soil heavy-metal values, and overcome the poor convergence and easy local optimization of the BP neural network.

3.1. Adaptive Step-Size Fruit Fly Optimization Algorithm Based on Individual Differences

The individual updating strategy in the swarm intelligence optimization algorithm balances the functions of local and global searches, making it the first choice for many researchers to optimize models and algorithms. The Fruit Fly Optimization Algorithm (FOA) has a simple structure, strong operability, and requires few parameters to be adjusted, making it a representative swarm intelligence optimization algorithm.

3.1.1. Steps and Problems of FOA

Fruit flies search for food by following a scent; their position is related to the odor concentration. Individual fruit flies fly toward areas with a higher concentration of food odor. When one fruit fly finds the best position for food, other fruit flies converge toward it and start the next search. The FOA simulates this process, iterating a certain number of times to find the global optimal solution [17]. The steps of the FOA are presented as follows.

(1) Randomly initialize the starting positions of the fruit fly population, and set the maximum number of iterations maxgen and the population size.
(2) Determine the position of each fruit fly based on its flight direction and search step length.
(3) Calculate the concentration judgement value Si, which is the reciprocal of the distance between the position and origin coordinates.
(4) Substitute Si into the fitness function, which is also the odor concentration judgement function, to obtain the odor concentration for the individual fruit fly.
(5) Integrate the fruit fly population; the individual with the highest odor concentration is selected to represent the optimal fruit fly position.
(6) Save the best odor concentration value; find the coordinates of the individual with the best odor concentration value, and let the other fruit flies fly in that direction.
(7) The model begins iterative optimization, repeating Steps 2–6 and assessing whether the odor concentration is better than that of the previous iteration. If it is better, continue to search; otherwise, maintain the position of the previous generation of fruit flies and end the algorithm.

The FOA has some shortcomings in optimization problems, indicated in Step 2. The optimization step size set by the FOA during the fly search for food is a fixed value. This leads to weak information exchange between individual flies in the algorithm and limits the search ability of the fly to some extent. This results in individuals with small search iteration steps converging slowly in the fly population, whereas those with large search iteration steps oscillate repeatedly in later iterations, leading to a decrease in the local optimization ability of the model. Thus, differences between individual flies and their positions should be fully considered to improve the search ability of the FOA.

3.1.2. FOA Combined with Simulated Annealing Algorithm

Given the differences in position and search step size of individual flies during the optimization process, we adjusted the step size using the simulated annealing algorithm (SA) to make the fixed step-size adaptable to dynamically adjust to the conditions. The SA algorithm [18,19] aims to simulate a global optimal solution in high-temperature solid-state annealing.

According to the Metropolis criterion, the probability of a particle cooling down at temperature T is

$$\mathrm{p(dE)} = \exp(-\frac{\Delta \mathrm{E}}{\mathrm{kT}}) \quad (1)$$

where E is the internal energy of the particle at temperature T; ΔE represents the energy change, and k is Boltzmann's constant.

The steps for implementing the SA algorithm are presented as follows.

(1) The objective function $f(x)$, initial temperature t_0, and minimum temperature $t_{\min}$ are set.
(2) Let the current temperature be t and the feasible solution be x. The initial solution is randomly perturbed to generate a new solution.

$$\mathrm{x}' = \mathrm{x} + \Delta \mathrm{x} \quad (2)$$

The resulting energy difference is calculated as

$$\Delta \mathrm{f} = \mathrm{f}(\mathrm{x}') - \mathrm{f}(\mathrm{x}) \quad (3)$$

(3) The energy difference is calculated. If $\Delta f < 0$, then accept the new solution and replace the old solution, and move to the next iteration; if $\Delta f > 0$, then decide whether to keep the new solution based on probability according to the Metropolis criterion.
(4) Repeat steps 2–3, gradually reducing the temperature after a certain number of iterations. When $t < t_{\min}$, we iterate and output the optimal solution.

The moving step-size formula in the FOA is

$$L = \text{Random} * \text{Value} \tag{4}$$

Random is a random number between [0, 1]; *Value* is the initial step size, and is a constant value. Thus, the moving step size L is a random number in the range [0, *Value*]. However, the FOA has some drawbacks. In the early stage of fruit fly optimization, a small step-size leads to low optimization efficiency; in the later stage of optimization, a large step-size makes it easy to miss the optimal solution [20].

To address these issues, the FOA uses a non-uniform mutation strategy to improve the moving step-size, as follows:

$$L' = n(1 - \left(\frac{g-1}{\text{maxgen}}\right)^{\lambda})\text{sgn}(r - 0.5) \tag{5}$$

where n is the initial step size; g is the current iteration number; maxgen is the maximum number of iterations; λ is the nonuniform mutation factor, set to 2 in this study; sgn is the sign function; and r is a random number between [0, 1]. Equation (5) shows that the improved moving step-size decreases with an increase in the iteration number, which improves the optimization efficiency in the early stage of the optimization process and enhances the local search capability of the algorithm in the later stage to avoid missing the optimal solution as a result of a large search step-size.

Combining the FOA with the SA, the SA-FOA algorithm is proposed to increase the diversity of the search by probabilistically eliminating the worst solutions during the iteration process, ensuring that the algorithm can still start a new search from the worst solution as a new starting point when trapped in a local optimum, and improve the local search capability of the algorithm. The improved algorithm can effectively avoid step-size and local optimum problems in the FOA.

3.2. SA-FOA-BP Model

The Back Propagation (BP) neural network is a widely used neural network with input layer (i), hidden layer (h), and output layer (o), as well as the desired output (d) [21]. The key feature of the BP neural network is its ability to utilize external input samples for stimulation. In the BP neural network, each layer calculates the error between the result and the expected output, and propagates the error backward to the previous layer. It then adjusts the weights of the previous layer to reduce the difference between the network's output and the desired output until it stabilizes. Figure 1 presents structure of the BP neural network.

The basic structure of a BP neural network consists of the following components [22].

Input Layer i: It contains the input vector $x = (x_1, x_2 \cdots, x_n)$ and the weight values ω_{ih} connecting the input layer to the hidden layer. The input layer serves as the entry point of the model and is a collection of vectors formed by the input data. The first column represents the bias value b_i, and the second column to the last column represents the feature vectors (excluding the classification label). The output layer is the exit point of the model. It is the result of the dot product of the model's input layer and weights, followed by the application of the activation function.

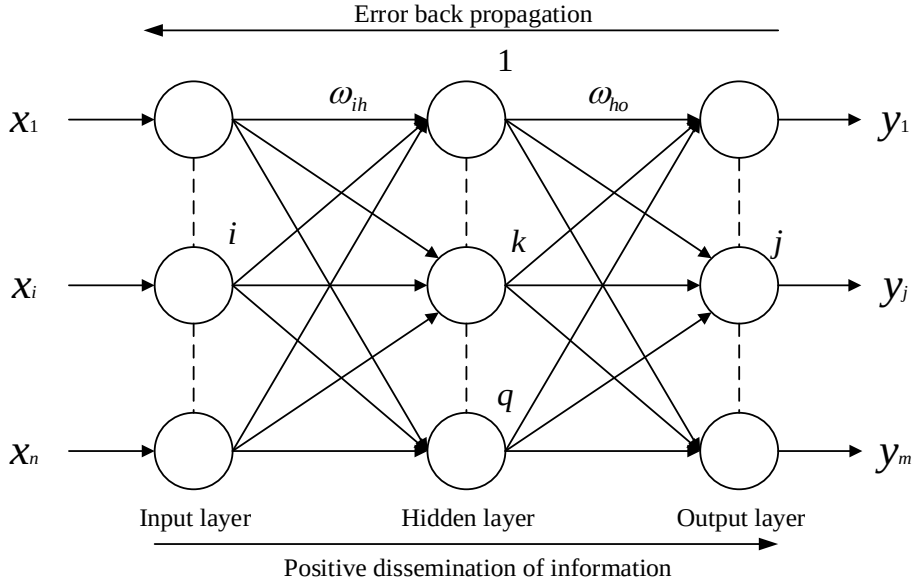


Figure 1. Structure of the BP neural network.

Hidden Layer h: It includes the input vector $yi = (yi_1, yi_2, \cdots, yi_n)$, output vector $yo = (yo_1, yo_2 \cdots, yo_n)$, bias value b_o, threshold for the connection between the hidden layer and the output layer, and the logistic activation function. In a BP neural network, there is no specific requirement for the number of hidden layers, but generally, it does not exceed 2 layers. The scalar result of the dot product between the output of the layer immediately preceding the hidden layer and the weights serves as the input for this layer. This scalar, combined with the calculation of the logistic function, forms the output of this layer.

The expected output $d_o = (d_1, d_2, \cdots, d_n)$ is the classification label vector. In a BP neural network, each layer calculates the error with respect to the expected output, propagates this error backward to the previous layer, and adjusts the weights of the previous layer accordingly.

In general, the number of neurons in the hidden layer of a BP neural network is typically calculated according to the following formula based on experience:

$$n_{hid} \leq \sqrt{n_{out} + n_{in}} + a \tag{6}$$

In Equation (6), n_{in} and n_{out} represent the number of neurons in the input and output layers, and the value of a ranges from [0, 10].

Traditional BP neural networks have limitations such as slow convergence, sensitivity to weight initialization, and easy trapping in local optima. Inspired by Boltzmann machines [23], the SA and FOA were combined with BP neural networks. The SA-FOA replaces the traditional gradient descent method used to update the weights and thresholds in BP neural networks. The FOA can find the global optimum in complex, multimodal, and non-differentiable vector spaces, and it can search for the initial weights of BP neural networks, overcoming the slow convergence of BP neural networks. The introduction of the SA reduces the possibility of the algorithm becoming trapped in a local optimum, further improving the accuracy of the BP neural network model.

The SA-FOA was used to search for the thresholds of nodes and weights of links in the BP neural network algorithm, which were assigned to the BP neural network; the BP neural network was trained to obtain the optimal parameters. Thus, the SA-FOA-BP neural network model can be divided into two parts: the first part is the SA-FOA optimization process, and the second part is the BP neural network training process.

The calculation steps of the model are presented as follows:

(1) Model initialization: Set the number of neurons in each layer of the BP neural network, including the number of neurons in the input layer m, number of neurons in the hidden layer n, and number of neurons in the output layer l. Set the initial position of the fruit fly population and set the maximum number of iterations *maxgen* and the population size.

(2) SA-FOA optimization: The process is introduced in Section 2.1 and is not repeated here. In this study, the error function e in the BP neural network is taken as the objective function, represented by concentration value in the FOA; the solution vector S is used as the comprehensive vector of connection weights and threshold values in the BP network. When the SA-FOA terminates, the weight and threshold values in the solution vector S are used to calculate the weight and threshold values of the current training sample in the BP neural network.

(3) BP neural network training: Assign the weight and threshold values obtained from the SA-FOA optimization to the BP neural network and train the BP neural network according to the preset parameters.

Figure 2 presents a flowchart of the SA-FOA-BP neural network model.

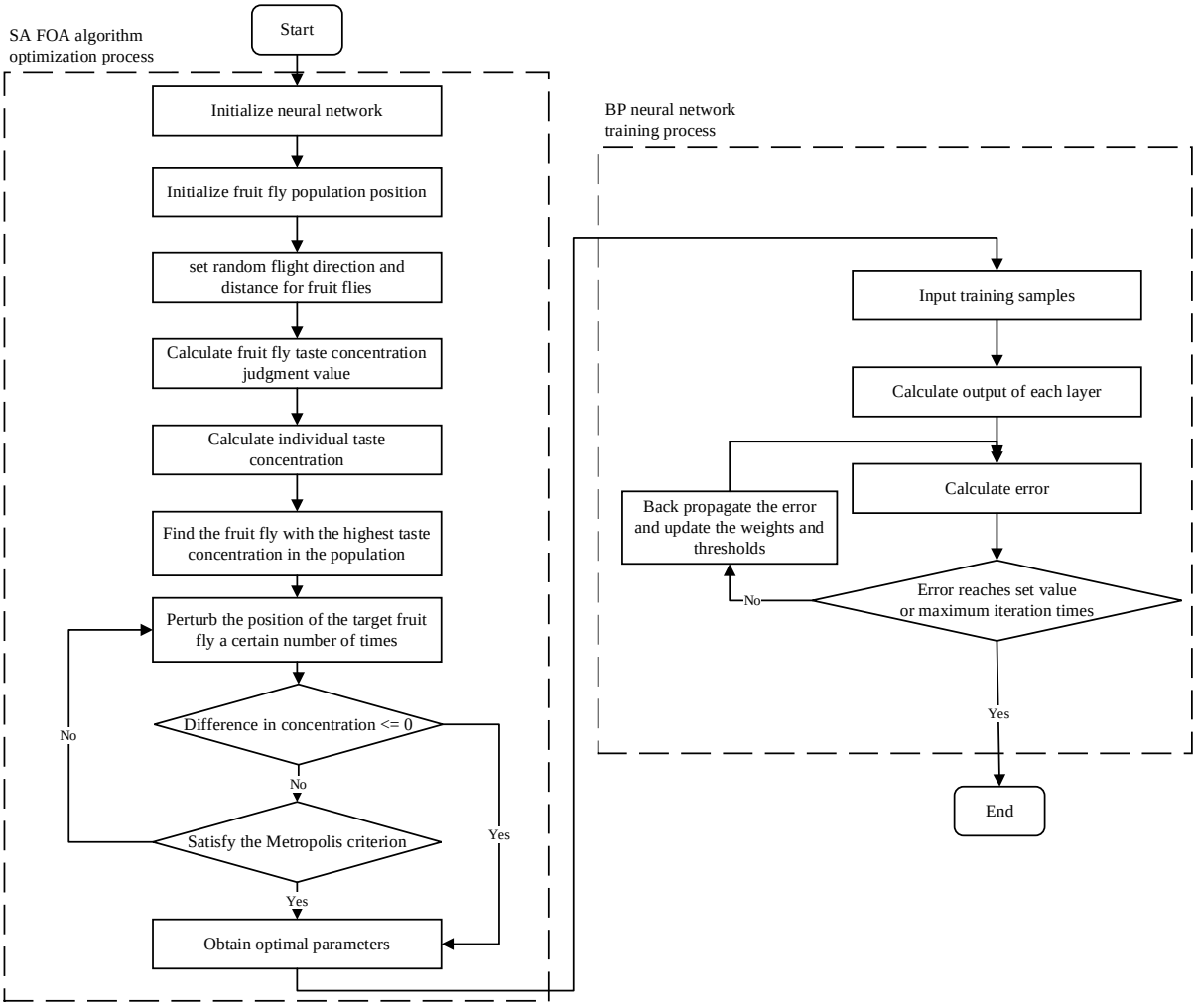


Figure 2. Flowchart of SA-FOA-BP neural network model calculation.

4. Prediction of Soil Heavy-Metal Content Using Graph Convolutional Networks Integrating Spatial Information

Spatial data contain coordinates and geometric structural features through which the location, shape, distribution, and relationships between objects can be represented. Traditional neural network methods ignore spatial modeling, whereas graph neural networks

represent spatial structures in a graph structure, providing new ideas for spatial regression prediction or interpolation.

When constructing a soil heavy-metal content prediction model, information from multiple sampling points can be combined to predict target points. To fully consider spatial features, a graph convolutional network model integrating spatial information was proposed to predict soil heavy-metal content at unknown points. The model is based on the graph convolutional neural network (GCN) and incorporates spatial information and points of interest [24] into the network.

First, we defined a data point $p_i = \{y_i, x_i, c_i\}$, where y_i is the dependent variable to be interpolated for prediction, x_i is a feature vector, and c_i is a coordinate vector. Thus, all points can be represented using $P = \{p_1, \ldots, p_n\}$. Using latitude and longitude information, all data points can be mapped in geometric space. Assuming that a certain data point is the center point, the distance d_{ij} between all data points and this point can be obtained using the Haversine formula. The Haversine formula is given by

$$d_{ij} = 2r\ argsin(\sqrt{sin^2(\frac{lat2 - lat1}{2})} + cos(lat2)cos(lat1)sin^2(\frac{lon2 - lon1}{2})) \quad (7)$$

where r is the radius of the Earth, $lon1$ and $lon2$ are the longitudes of the two points, and $lat1$ and $lat2$ are the latitudes of the two points.

The source of soil pollution directly affects heavy-metal concentrations. Thus, when predicting the heavy-metal content in soil, the features of the points of interest must be incorporated into the prediction model. In this study, the pollution source was used as the point of interest, and all pollution enterprises in the study area were regarded as a set. The interest-point matrix is defined as $\boldsymbol{V} = \{v_1, \ldots, v_n\}$, where v_i is the coordinate vector of the point of interest.

Considering the dependency between the center point and other data points, which gradually weakens with an increase in the distance between the points, the correlation also becomes weaker. Thus, the k-nearest neighbor method or setting a reasonable distance threshold can be used to improve training efficiency. Using this method, the neighborhood of each data point can be obtained and an adjacency matrix A that incorporates the fusion interest-point features can be created.

Thus, a graph $\boldsymbol{G} = (V, E)$ is constructed consisting of nodes $V = \{v_1, \ldots, v_n\}$ and edges $E = \{e_1, \ldots, e_n\}$, where the nodes and edges of the graph are provided by the adjacency matrix A. Each node has a fused interest-point information node feature x_i and y_i. Typically, the adjacency matrix A is binary. This study attempts to construct A based on different distances d_{ij} between nodes. Thus, the normalized adjacency matrix $\overline{\mathbf{A}}$ can be represented as

$$\overline{\mathbf{A}} = \mathbf{D}^{-1/2}/(\mathbf{A} + \mathbf{I})\mathbf{D}^{-1/2} \quad (8)$$

where D is the degree matrix, and I is the identity matrix of graph G. The propagation calculation process between the levels in the GCN is presented as follows.

$$\mathrm{H}^{(l+1)} = \mathrm{ReLU}(\overline{\mathrm{A}}\mathrm{H}^{(l)}\mathrm{W}) \quad (9)$$

where W is the parameter matrix of θ for the lth layer, and $H^{(l)}$ is the input matrix of graph convolution in the lth layer. The feature matrix X $=\{x_1, \ldots x_n\}$ contains all node feature vectors, and X is used as the input of $H^{(0)}$ for the first layer of the GCN. The output $\hat{y}$ of the GCN can be obtained by a parameterized GCN through the feature matrix.

4.1. Design of Spatial Information Encoders

In a traditional GCN, the connections between nodes are the only direct source of spatial context; they can be convolved similarly to the pixels in the image data. This limits the ability of the GCN to capture spatial features in multiple ways. GCN performance requires a good neighborhood structure, but a traditional GCN has arbitrary settings for

the neighborhood, such as selecting the k-nearest neighbors of each node to construct the neighborhood structure. If the underlying data cannot capture prior knowledge, setting the correct neighborhood parameters becomes difficult. In addition, a single k value may not be reasonable for all nodes, and different nodes may depend on their neighbors to different degrees. The traditional GCN structure does not include a module that converts point coordinates into different spaces. Converting point coordinates into different spaces can provide more information on the spatial structure [25].

It is difficult for a GCN to handle complex spatial dependencies, similar to spatial interpolation, which cannot be solved using other methods. Thus, a spatial position context-aware encoding method is proposed to construct a spatial encoder (SE) for encoding spatial information that can flexibly learn the features of each spatial coordinate.

Assuming that the spatial coordinate matrix is $\mathbf{C} = \{c_1, \ldots c_n\}$, c_i represents the latitude and longitude coordinates of each data point. The spatial information encoder is composed of sine and cosine functions and a fully connected network; the spatial information encoder is defined as $SE(\mathbf{C}) = NN(PE^{(g)}(C))$, where $PE^{(g)}(\mathbf{C}) = \left[PE_0^{(g)}(\mathbf{C}); \ldots; PE_{S-1}^{(g)}(\mathbf{C})\right]$, and $PE^{(g)}(\mathbf{C})$ is the series composition of multi-scale sine and cosine functions. In each scale, its encoder is $PE_s^g(\mathbf{C}) = \left[PE_{s,1}^{(g)}(\mathbf{C}); \ldots; PE_{s,2}^{(g)}(\mathbf{C})\right]$, and the spatial dimensions v of C (latitude and longitude) are processed separately such that

$$g = \lambda_{\max}/\lambda_{\min} \tag{10}$$

$$PE_{s,v}^{(g)}(\mathbf{C}) = \left[\cos\left(\frac{\mathbf{C}^{[v]}}{\lambda_{\min} g^{s/S-1}}\right); \sin\left(\frac{\mathbf{C}^{[v]}}{\lambda_{\min} g^{s/S-1}}\right)\right] \forall s \in \{0, \ldots, S-1\}, \forall v \in \{1,2\} \tag{11}$$

where $\lambda_{\min}$ are the minimum and $\lambda_{\max}$ are the maximum grid scales, the output of $PE^{(g)}(\mathbf{C}) \subseteq$ is passed through the vector space required by the fully connected neural network, and the coordinate embedding matrix $\mathbf{C}_{emb}$ is created.

4.2. Spatial Autocorrelation for Aiding Learning

According to the first law of geography, the closer two things are to physical space, the more likely they are to be related. Real-world objects are rarely randomly distributed. Data points closer to the spatial distance tend to exhibit strong spatial correlations.

Global spatial autocorrelation [26] reflects spatial clustering on a global scale. However, there is often local instability within objects, which requires analysis using local spatial autocorrelation. The local Moran's I index [27] was calculated to capture spatial heterogeneity and understand local spatial clustering characteristics. The local Moran's I index was calculated using Equation (13).

$$S^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2 \tag{12}$$

$$I_i = \frac{x_i - \overline{x}}{S^2}\sum_{j \neq i}^{n} w_{ij}(x_j - \overline{x}) \tag{13}$$

In Equation (13), $\overline{x}$ represents the mean value of all units within the area, represents the variance, x_i and x_j represent the values of units within the area, w_{ij} represents the total number of units within the area.

In this study, Moran's I index of spatial autocorrelation, denoted as I(Y), was used for auxiliary learning to jointly construct new prediction results with the prediction results Y of the GCN. The limitation of local spatial autocorrelation in the neighborhood may lead to neglecting the influence of longer distances. Thus, in the GCN training process, a small batch of points was used. For each training session, n points were randomly selected from the training data as batch B for training. The corresponding adjacency matrix $\mathbf{A}_B$ was constructed using the data points in the batch $I(\mathbf{Y}_B)$, and the corresponding local Moran's I

index B was calculated. Using different batches for training, the central point may have different neighborhoods in different training iterations. Thus, Moran's I of the central point can change continuously throughout the iteration process, reflecting more distant or closer neighborhoods, which helps overcome the scale sensitivity of Moran's I.

4.3. Construction of Graph Convolutional Networks Integrating Spatial Information

GCN (Graph Convolutional Network) is a deep learning method for handling data with graph-like properties. Its main purpose is to extract spatial features from data by leveraging the relationships between data points. Currently, graph convolution includes graph convolutional neural networks based on the spatial domain (vertex domain) and graph convolutional neural networks based on the spectral domain. Figure 3 shows the GCN structure diagram.

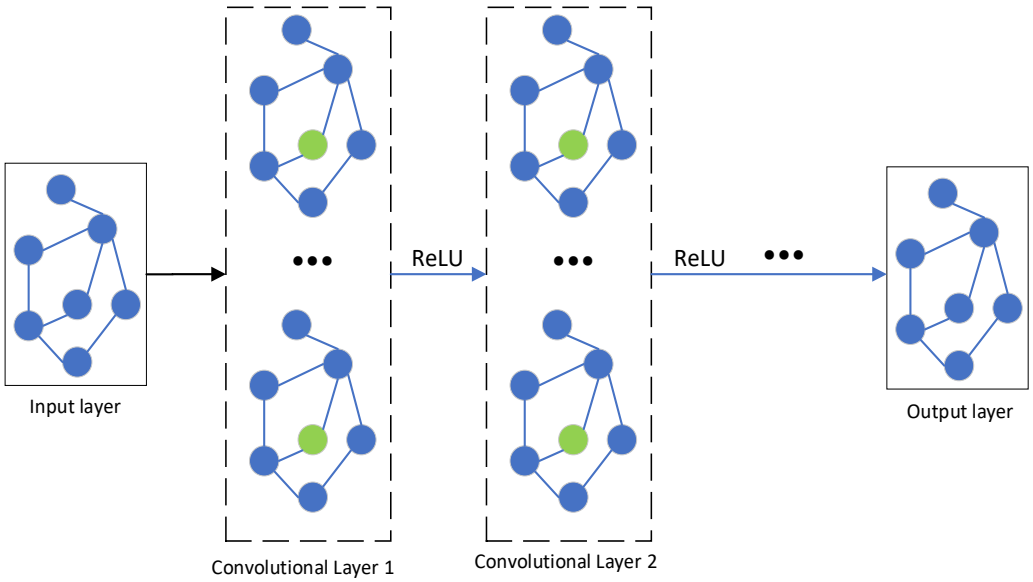


Figure 3. GCN structure diagram.

A GCN that integrates spatial information includes a spatial information encoder that is used to learn the contextual information of point coordinates throughout the GCN training process. In constructing the SE-GCN, spatial information and point-of-interest information are fused as the model input, and the SE-returned coordinates are embedded into a matrix and connected with other node features to provide training data for the GCN operator. The SE-GCN predicts the output of Moran's I index of spatial autocorrelation as an auxiliary prediction result. The Moran's I index was obtained by constructing a new training graph based on the k-nearest neighbors [28] from a randomly sampled batch of points during each training; the final result was obtained through prediction of the local Moran's I index and SE-GCN [29]. The model structure of the SE-GCN is shown in Figure 4.

We assumed a randomly sampled batch of points B, where $p_1, \ldots, p_{n_{batch}} \in B$ is one of the points in $c_1, \ldots, c_{n_{batch}}$. Using the coordinates of B, a spatial graph was constructed using the k-nearest neighbor algorithm, and the adjacency matrix $\mathbf{A}_B$ was obtained. Next, the coordinates were embedded into a latent space using a spatial information encoder consisting of trigonometric functions and a single fully connected neural network, resulting in the coordinate-embedding vector $\mathbf{C}_B^{emb} = \mathbf{c}_1^{emb}, \ldots, \mathbf{c}_{n_{batch}}^{emb}$. The output of the spatial information encoder was connected to the node features to create the input for the first layer of the SE-GCN, denoted as

$$\mathrm{H}^{(0)} = concat\left(\mathrm{X}_B, \mathrm{C}_B^{emb}\right) \tag{14}$$

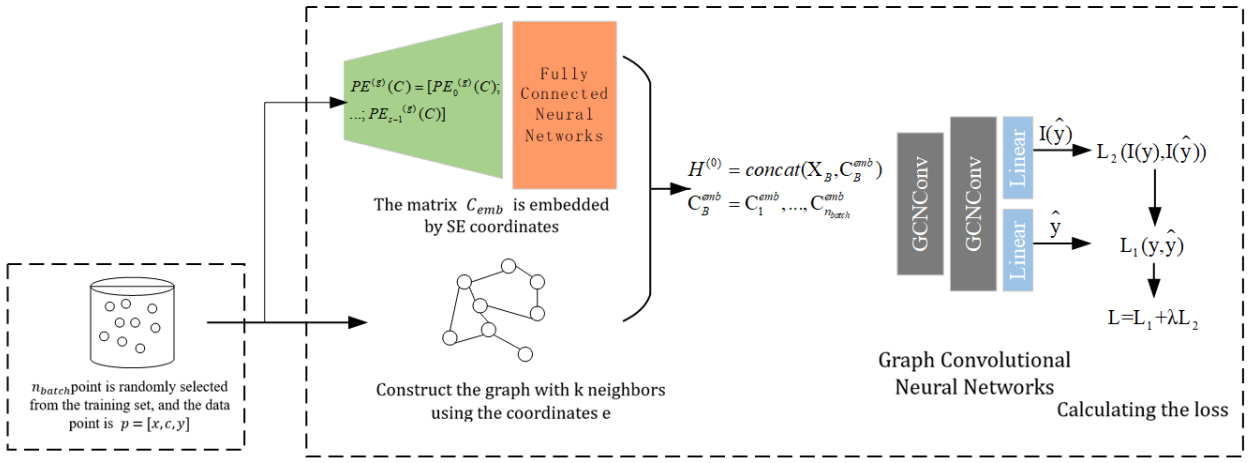


Figure 4. SE-GCN model structure.

To incorporate the local Moran's I index into the SE-GCN, the measurement $I(\mathbf{Y}_B)$ of the resulting variable $\mathbf{Y}_B$ was calculated using the formula for the local Moran's I index at the beginning of each training based on the spatial weights of the adjacency matrix $\mathbf{A}_B$. $\mathbf{Y}_B$ and $I(\mathbf{Y}_B)$ were predicted.

$$\hat{\mathrm{Y}}_B = \mathrm{H}^{(l)} \tag{15}$$

$$I(\hat{\mathrm{Y}}_B) = \mathrm{H}_{aux}^{(l)} \tag{16}$$

The loss of SE-GCN can be calculated using any regression criterion, such as the mean squared error (MSE):

$$loss = MSE(\hat{\mathrm{Y}}_B, \mathrm{Y}_B) + \lambda MSE(I(\hat{\mathrm{Y}}_B), I(\mathrm{Y}_B)) \tag{17}$$

where λ represents the weight of auxiliary learning.

The steps for using the SE-GCN model to predict soil heavy-metal content are presented as follows.

(1) Determine the graph structure; input the training dataset of soil heavy metal sampling points, including all features of the spatial dimension, the pollution source dataset, and input of sampling points to form an input vector. Integrate the characteristics of pollution sources to construct an adjacency matrix. Initialize the model parameters.
(2) The model performs spatial feature extraction. The spatial features of each sampling point are extracted from the graph convolutional neural network. The coordinate embedding matrix obtained using the spatial information encoder is connected to the node features extracted by the model to provide training data for the GCN operator.
(3) Further model training, output Moran's I index of spatial autocorrelation, and provide auxiliary learning for prediction results.
(4) Iteratively learn the SE-GCN model, calculate its loss function value, iterate continuously, and obtain an optimal SE-GCN model.
(5) The test set is input into the optimal SE-GCN model to predict the heavy-metal content in the soil and the predicted values at the required points.

The k-nearest neighbor algorithm is used in the SE-GCN model to define the spatial graph; however, the proposed spatial information encoder is not limited by the k-nearest neighbor algorithm because it does not operate on the graph convolutional network. This enables the spatial information encoder to learn context-aware information for each coordinate individually, considering the potential distance neighborhoods within different batches. Whereas spatial graphs rely on straight-line distances between points, spatial information encoders embed latitude–longitude values into a high-dimensional latent space. This feature information is passed on during the learning process. SE-GCN model

training was performed on a new graph composed of randomly sampled data points at each iteration, which helped to better generalize the spatial feature information while remembering the neighborhood structure. Different training batches can also help calculate Moran's I index, which captures the spatial autocorrelation of near and far neighbors. The local Moran's I index was calculated using Equation (13). The Moran's index can range from -1 to 1 and is used to measure spatial autocorrelation. A higher value indicates a greater degree of clustering of similar regions in space and is positive when there is positive correlation. Conversely, a lower value indicates a higher level of dispersion, and it is negative when there is negative correlation.

5. Analysis of Experimental Results

5.1. Analysis of Prediction Results of Missing Soil Heavy-Metal Values

Using soil from a certain area in North China as the experimental object, the experimental data included soil heavy-metal content, soil pH, and sampling latitude and longitude. The data dimensions differed. Before the experiment, it was necessary to normalize the data characteristics to eliminate the influences between data with different dimensions and improve the speed of the model calculation. In this experiment, the data sample size was not large, and the input and output layer dimensions were small; thus, the neural network model used a three-layer network structure to reduce the amount of mathematical calculation while ensuring accuracy. In a neural network, we attempt to ensure that the dimension of the input vector matches the number of neurons in the input layer, and the dimension of the output vector matches the number of neurons in the output layer. In the missing value prediction experiment, the number of neurons in the input layer was set to five and the number of neurons in the output layer was set to three. The data were divided into training and test sets in a ratio of 7:3.

Table 1 shows a comparison of the measured values of Hg, Cd, and Pb with the predicted results based on the SA-FOA-BP neural network model and the traditional BP neural network algorithm. Due to the particularity of soil heavy-metal data, the specific values are inconvenient for public display. The 40 predicted datasets were divided into four groups, each with 10 data points, and the average values were tabulated. The predicted results of the SA-FOA-BP neural network model and the measured results were compared; the relative error range for Hg was 6.78–23.16%, and the average error was 16.35%. Compared with the traditional BP neural network prediction, the average error value was reduced by 11.251%. The relative error range for Cd was 7.33–22.86%, and the average error value was 16.37%. Compared with the traditional BP neural network prediction, the average error value was reduced by 10.21%. The relative error range of Pb was 1.71–26.20%, and the average error value was 14.69%. Compared with the traditional BP neural network prediction, the average error value was reduced by 11.699%. Overall, the error between the value predicted by the SA-FOA-BP neural network model and the measured value ranged from 1.71 to 26.20%; the average error was 15.8033%, within the error range reported in similar research (0.93–26.67%), indicating that prediction using the SA-FOA-BP neural network model is reasonable.

Figures 5–7 show comparison charts of the measured values of the three heavy metals and the values predicted by the different models.

To validate the effectiveness of the SA-FOA algorithm, this section used the SA-FOA algorithm and the FOA algorithm to solve a univariate quadratic equation for its minimum value in MATLAB. The parameter settings were as follows: In the FOA algorithm, the maximum number of iterations was set to maxgen = 200, and the population size was size = 50. In the SA-FOA algorithm, the maximum number of iterations was also set to maxgen = 200, the population size was size = 50, and the non-uniform mutation factor λ was set to 2. The univariate quadratic equation to be solved is represented as Equation (18):

$$f(x) = -5 + x^2 \tag{18}$$

Table 1. Comparison of SA-FOA-BP neural network and traditional BP neural network prediction results.

Heavy Metal Elements	Sample Number	Average Actual Value/ (mg/kg^{-1})	SA-FOA-BP Neural Network Prediction Results			BP Neural Network Prediction Results		
			Average Predicted Value/ (mg/kg^{-1})	Mean Relative Error	Overall Mean Error	Average Predicted Value/ (mg/kg^{-1})	Mean Relative Error	Overall Mean Error
Hg	1–10	0.191	0.208	19.46%	16.35%	0.183	27.57%	27.601%
	11–20	0.168	0.184	14.25%		0.159	26.86%	
	21–30	0.172	0.167	18.36%		0.175	28.66%	
	31–40	0.171	0.175	13.28%		0.156	27.32%	
Cd	1–10	0.367	0.365	17.14%	16.37%	0.332	27.31%	26.58%
	11–20	0.197	0.192	15.95%		0.186	26.21%	
	21–30	0.274	0.257	16.44%		0.226	25.95%	
	31–40	0.163	0.174	16.74%		0.175	26.83%	
Pb	1–10	23.251	24.459	15.84%	14.69%	24.248	27.24%	26.389%
	11–20	26.658	25.909	15.16%		26.876	26.32%	
	21–30	27.685	28.440	13.39%		25.855	26.81%	
	31–40	24.429	24.138	14.35%		24.274	25.14	

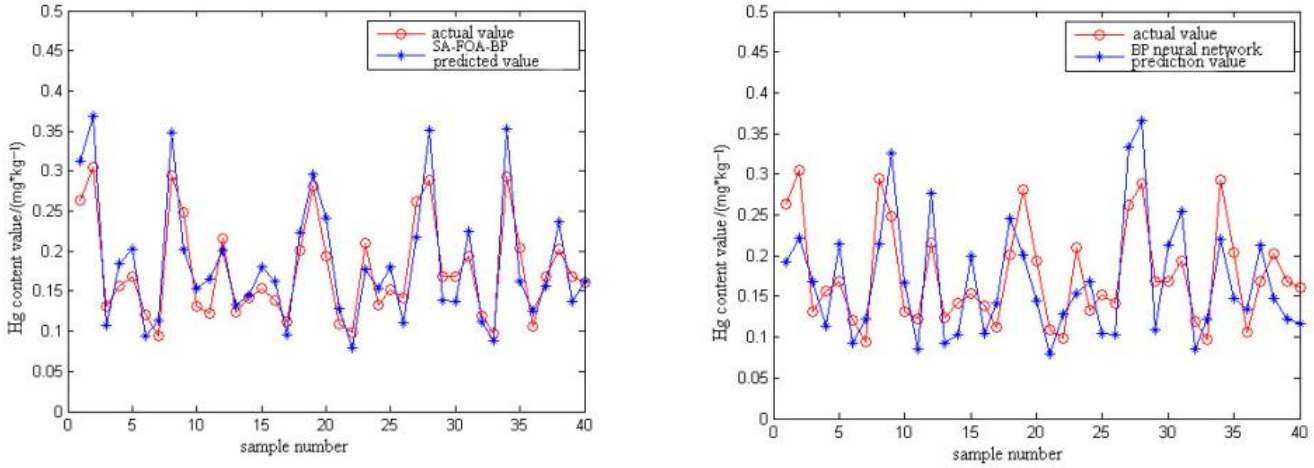


Figure 5. Comparison of measured Hg value and values predicted by different models.

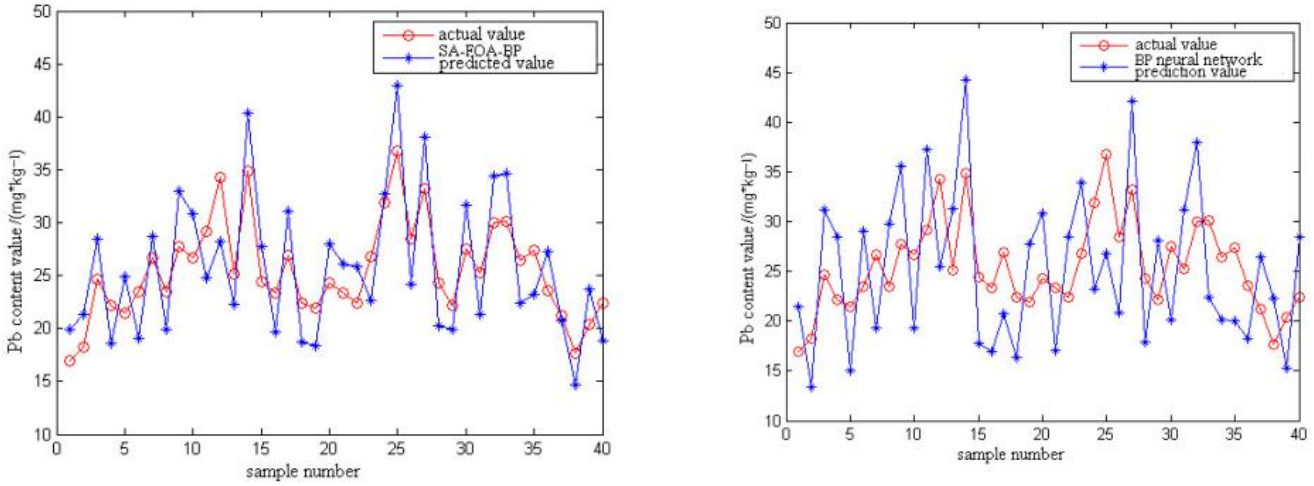


Figure 6. Comparison of measured Pb value and values predicted by different models.

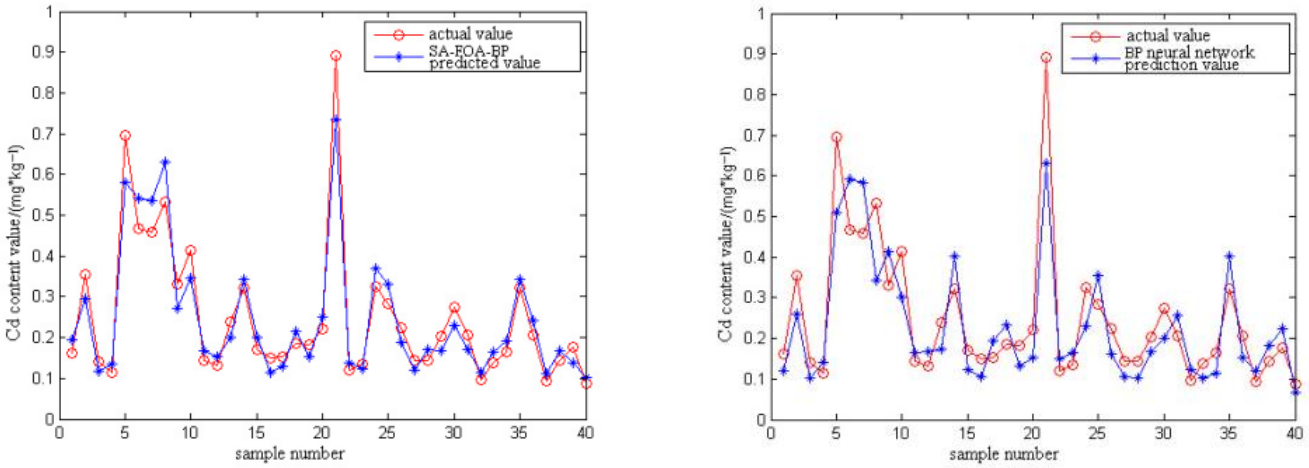


Figure 7. Comparison of measured Cd value and values predicted by different models.

By calculation, it is known that the minimum value of this equation is −5, and the iteration process is shown in Figure 8:

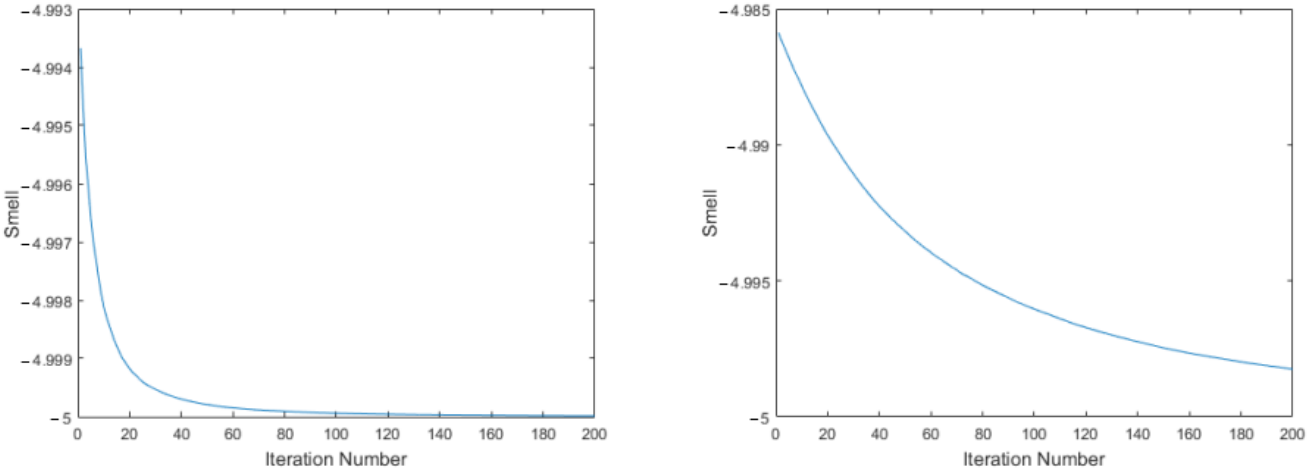


Figure 8. Iteration Process of SA-FOA and FOA Algorithms.

In Figure 8, the left image represents the iteration graph of the SA-FOA algorithm, while the right image represents the iteration graph of the traditional FOA algorithm. It can be observed that after 200 iterations, the SA-FOA algorithm has already calculated a value of −5. However, under the same number of iterations, the traditional FOA algorithm converges slower than the SA-FOA algorithm. This indicates that the SA-FOA algorithm, compared to the FOA algorithm, has a more noticeable optimization effect for solving general function problems.

5.2. Analysis of Soil Heavy-Metal Prediction Results

In the soil heavy-metal content prediction experiment, the dataset was divided into training and test sets to train and verify the performance of the model, respectively. The experimental dataset contained 322 sets of heavy-metal data, elevation information for each sampling point, and the coordinates of 224 enterprises that may produce pollution. The originally obtained data were normalized and preprocessed to produce the experimental dataset. The possible polluting enterprises were used as points of interest and combined with node characteristics. Ninety percent of the data in the experimental dataset were randomly selected as the training set; the remaining 10% of the data were used as the test set to predict the contents of Hg and Ni in the soil at unknown points. The model performance was verified using the test set.

For prediction of mercury and nickel in the soil, the RMSE, MAE, and R^2 at different λ were obtained; the calculated error indicators are presented in Table 2.

Table 2. Error indicators for each heavy metal at different λ.

Element	λ	RMSE	MAE	R^2
Hg	0	0.0265	0.0007	0.857
	0.25	0.0207	0.0004	0.912
	0.5	0.0226	0.0005	0.896
	0.75	0.0223	0.0005	0.898
Ni	0	1.095	4.973	0.837
	0.25	0.774	3.517	0.918
	0.5	0.818	3.763	0.909
	0.75	0.86	3.836	0.899

From Table 2, the SE-GCN model predicts mercury at λ 0.25; the RMSE, MAE, and R^2 are lower than other values. The MAE decreased by 20% and R^2 increased by 0.016 compared to λ 0.75. The RMSE decreased by 7.17%, the MAE decreased by 20%, and R^2 increased by 0.014 in the prediction of nickel. The RMSE, MAE, and R^2 were lower than other values. Compared to a value of 0, the RMSE decreased by 29.32%, the MAE decreased by 29.28%, and R^2 increased by 0.25. Compared with a value of 0.5, the RMSE decreased by 5.38%, the MAE decreased by 6.54%, and R^2 increased by 0.009. Compared with a value of 0.75, the RMSE decreased by 10%, the MAE decreased by 8.32%, and R^2 increased by 0.019. In predicting soil heavy metals, adding spatial autocorrelation can alleviate most of the overfitting problems in the model.

From Table 2, the SE-GCN model had lower RMSE, MAE, and R^2 values for each evaluation indicator of mercury at $\lambda = 0.25$ than the other models. Compared with $\lambda = 0$, the RMSE decreased by 21.89%, the MAE decreased by 42.86%, and R^2 increased by 0.055. Compared with $\lambda = 0.5$, the RMSE decreased by 8.41%, the MAE decreased by 20%, and R^2 increased by 0.016. Compared with $\lambda = 0.75$, the RMSE decreased by 7.17%, the MAE decreased by 20%, and R^2 increased by 0.014. At $\lambda = 0.25$, the RMSE, MAE, and R^2 for nickel were lower than for other metals. Compared with $\lambda = 0$, the RMSE decreased by 29.32%, the MAE decreased by 29.28%, and R^2 increased by 0.081. Compared with $\lambda = 0.5$, the RMSE decreased by 5.38%, the MAE decreased by 6.54%, and R^2 increased by 0.009. Compared with $\lambda = 0.75$, the RMSE decreased by 10%, the MAE decreased by 8.32%, and R^2 increased by 0.019. Thus, the SE-CGN model constructed in this study can alleviate most of the overfitting problems by adding spatial autocorrelation in predicting soil heavy metals.

Figure 9 shows the fitting curves of the predicted and actual values of Hg for different λ. The predicted Hg with $\lambda = 0.25$ fit best with the actual values.

Figure 10 shows the fitting curves of the predicted and actual values of Ni predicted by SE-GCN for different λ. The predicted Ni with $\lambda = 0.25$ fit best with the actual value.

The prediction performance of the proposed SE-GCN model was further demonstrated through comparative experiments. The SE-GCN model (for 0.25), GCN model, DCNN, Kriging, and RBF were used to predict soil heavy-metal content, and the RMSE, MAE, and R^2 indices were compared. The evaluation indices for the predicted soil heavy-metal values for each model are shown in Table 3.

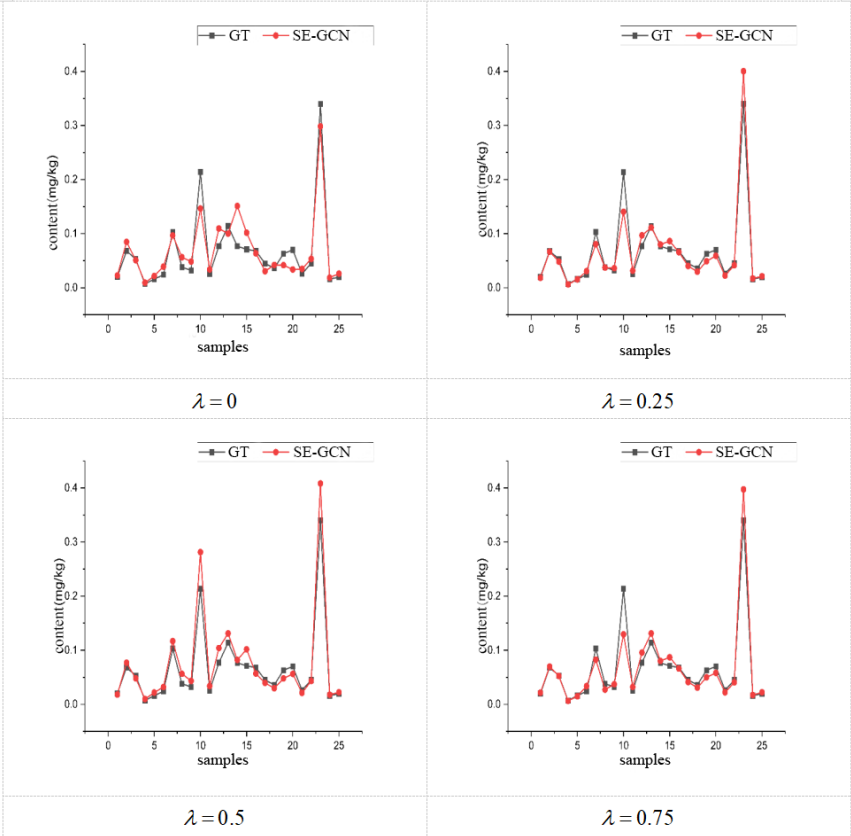


Figure 9. Fitted curves of predicted and actual values of Hg for different λ.

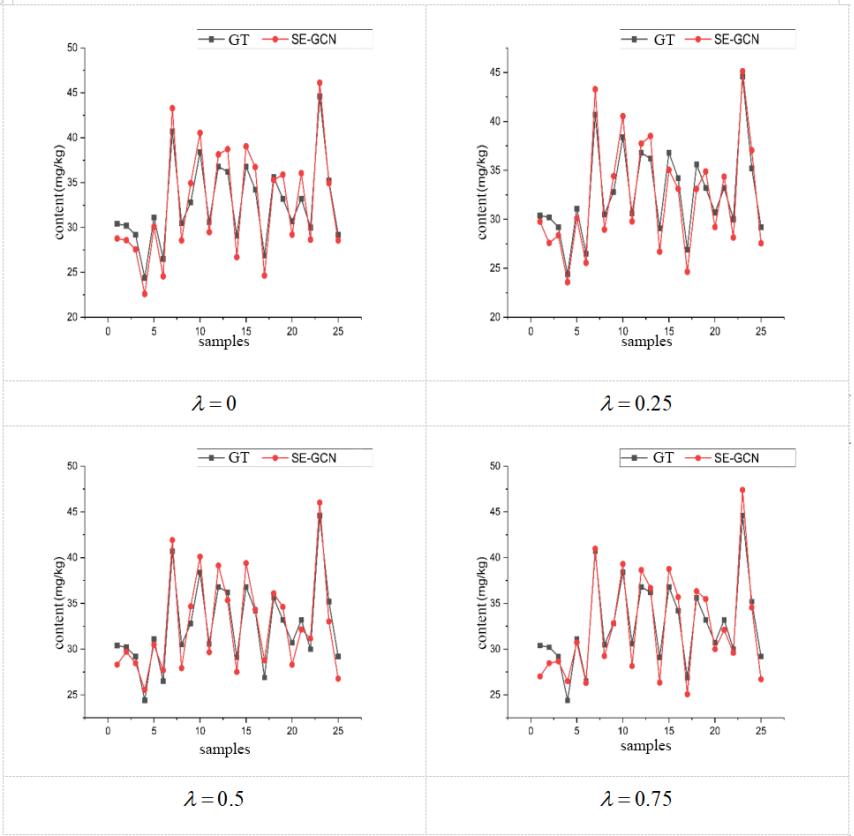


Figure 10. Fitting curves of predicted and actual values of Ni for different λ.

Table 3. Error indicators for each heavy metal for different models.

Element	Model	RMSE	MAE	R^2
Hg	SE-GCN	0.0207	0.0004	0.912
	GCN	0.0273	0.0007	0.848
	DCNN	0.0219	0.0005	0.902
	Kriging	0.0393	0.0015	0.686
	RBF	0.0370	0.0014	0.720
Ni	SE-GCN	0.774	3.517	0.918
	GCN	2.394	5.733	0.871
	DCNN	2.722	7.41	0.834
	Kriging	3.512	12.335	0.723
	RBF	3.687	13.596	0.695

In Table 3, the RMSE, MAE, and R^2 of the SE-GCN model were superior to those of the GCN model, DCNN, Kriging, and RBF. In predicting Hg, the SE-GCN model had a 24.18% lower RMSE, 42.86% lower MAE, and R^2 was 0.064 lower than the GCN model; R^2 decreased by 0.064. Compared with the DCNN model, the RMSE of SE-GCN decreased by 5.48%, the MAE decreased by 20%, and R^2 decreased by 0.1. Compared with the Kriging model, the RMSE of SE-GCN decreased by 47.33%, the MAE decreased by 73.33%, and R^2 decreased by 0.226. Compared with the RBF model, in predicting Ni, the SE-GCN model had a 67.67% lower RMSE, 38.65% lower MAE, and R^2 was 0.047 lower. Compared to the GCN model, the SE-GCN model had a 71.57% lower RMSE, 71.57% lower MAE, and R^2 was 0.047 lower. Compared to the DCNN model, the SE-GCN model had a 71.57% lower RMSE, 52.54% lower MAE, and R^2 was 0.084 lower. Compared with the Kriging model, SE-GCN had a 77.96% lower RMSE, 71.49% lower MAE, and R^2 was 0.195 lower. Compared with the RBF model, SE-GCN had a 79.01% lower RMSE, 74.13% lower MAE, and R^2 was 0.223 lower.

Figures 11 and 12 show the absolute errors between the actual and predicted values of heavy-metal content for different models.

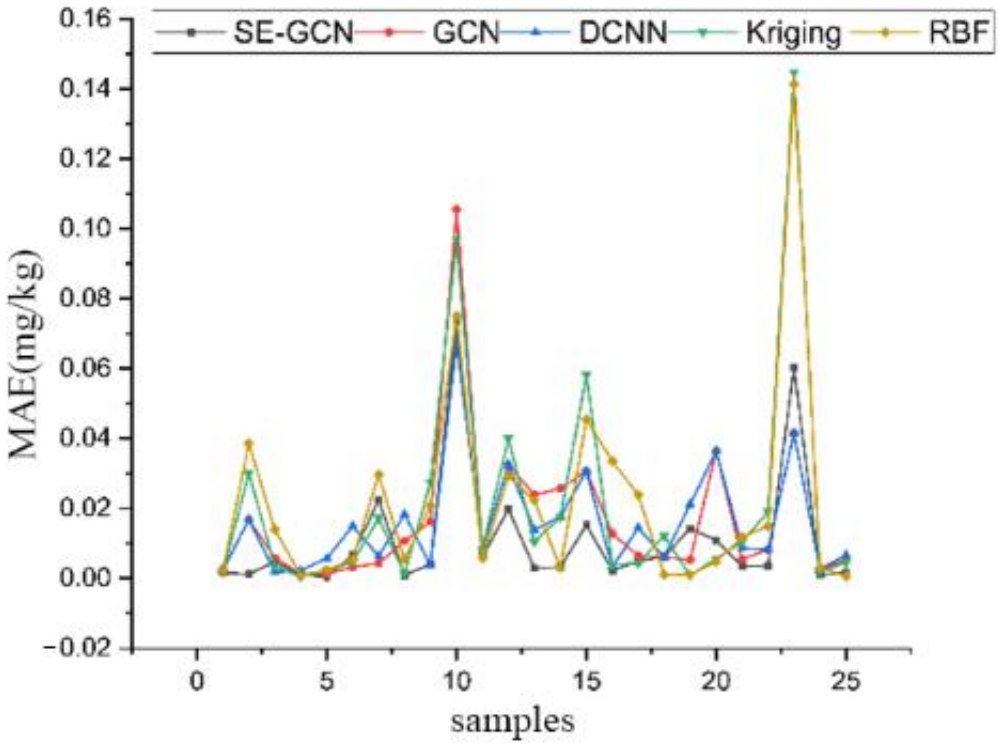


Figure 11. Absolute error between actual and predicted values of Hg for different models.

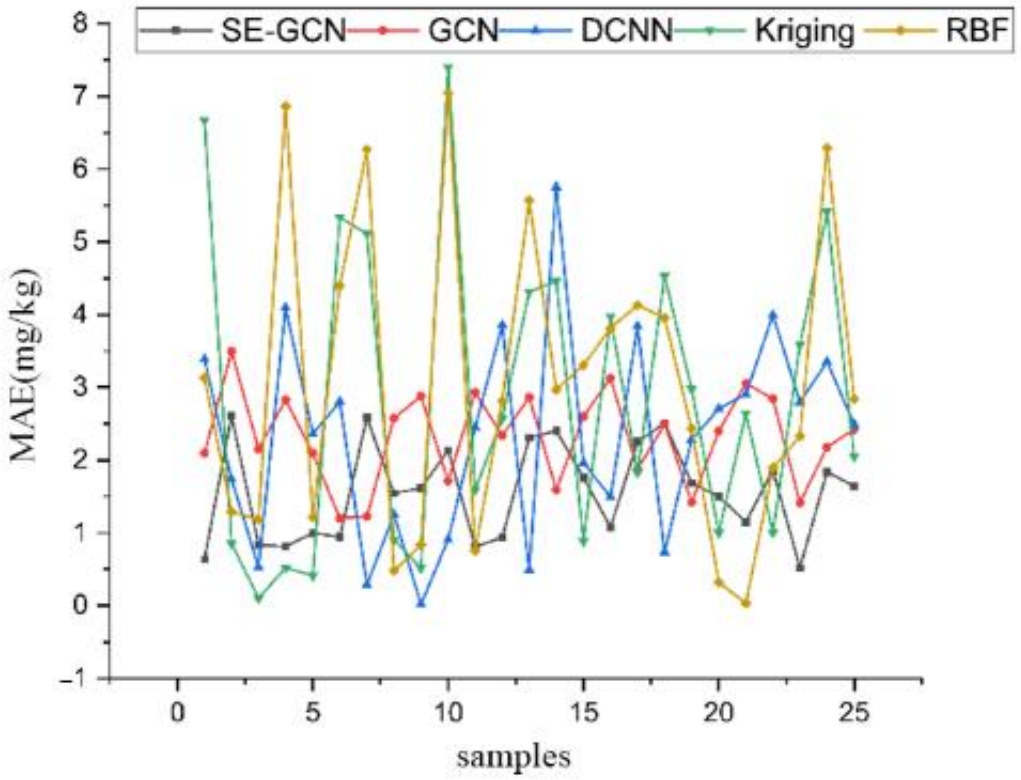


Figure 12. Absolute error between actual and predicted values of Ni for different models.

From Figures 11 and 12, the absolute errors of the SE-GCN model predictions for Hg and Ni were smaller than those of the GCN, DCNN, Kriging, and RBF models. For Ni, the absolute error of the predictions of the SE-GCN model fluctuated less than those of the GCN, DCNN, Kriging, and RBF models. Thus, compared to GCN, DCNN, Kriging, and RBF models, the proposed SE-GCN model demonstrated higher accuracy and better performance in predicting soil heavy-metal content.

In experiments conducted on the SE-GCN model, the training time and prediction time were recorded. The model exhibited stable training and prediction times. The average training time among 100 experiments was 2.71 s, and the average prediction time was 0.0044 s. These experiments were conducted in an environment with a CPU i7-12650H, 16 GB of memory, and running on Windows 11.

5.3. Analysis of Two Improved Applications of Neural Network Models

This article discusses the principles and methods of traditional backpropagation neural networks (BPNN) and graph convolutional neural networks (GCNN) for predicting soil heavy-metal pollution, and validates the accuracy and reliability of the predictions of both models through experiments. Overall, the predicted values obtained by these models indicate a good fit with the actual soil heavy-metal values. Thus, neural network models have good practical value in predicting heavy-metal pollution in soil.

However, through experiments and comparative analysis, it is observed that compared with traditional neural networks, graph convolutional neural networks have a lower overall average error in predicting soil heavy-metal pollution. This is because graph convolutional neural networks are a type of deep learning model that extracts features from graph data that can be used for accurate prediction. Traditional neural networks, with their inherent problems, produce a higher overall average error in predicting soil heavy-metal pollution using this method. Both models demonstrate good efficiency and relatively high fitting accuracy.

Through experiments and analysis, the following conclusions have been drawn:

(1) This study used a simulated annealing algorithm to optimize the fruit fly algorithm and replace the traditional BP neural network parameter optimization method, improving the traditional BP neural network with better prediction results for missing values of heavy metals in soil. Although the traditional neural network has fewer layers and lower simulation accuracy, it can still be used as a soil heavy metal prediction model when few data samples are available.

(2) The graph convolutional network that integrates spatial information has a high accuracy in predicting soil heavy-metal content. However, research has found that predictive accuracy distribution is not uniform. For areas with flat terrain, the predictive accuracy for soil heavy-metal content was relatively high, whereas for areas with significant terrain changes, the predictive accuracy was relatively low. Thus, prediction of soil heavy-metal content is related to spatial distribution and also to the elevation of the sample points.

(3) Different neural network models are suitable for different application scenarios and data sample sizes. Traditional neural network models are more appropriate for scenarios with limited data samples and few predicted variables.

(4) Models such as the GCNN are applicable for application scenarios with relatively large sample sizes and significant prediction tasks.

6. Conclusions

This study used two neural network algorithms to predict the missing values of heavy metals in soil and the heavy-metal content. Two neural network models were proposed: the SA-FOA-BP and SE-GCN models. The SA-FOA-BP neural network model was used to predict the missing values of heavy metals in the soil; Compared to the traditional BP neural network model, the prediction errors for Hg, Cd, and Pb in this model fall within the range of 1.71% to 26.20% when compared to the actual values, with an average error of 15.8033%. The error was within the range of similar studies. This model improves the prediction accuracy of missing values for soil heavy-metal content by performing data interpolation and optimizing neural network parameters. Thus, it is feasible to use the SA-FOA-BP neural network model to predict the missing values of heavy metals in soil samples. The SE-GCN model considers point-of-interest information as input features and proposes a spatial information encoder. The spatial information encoder can be used to learn the spatial contextual information of each point coordinate. Using the spatial information encoder, a coordinate embedding matrix can be obtained; it is combined with the node features, providing a new approach for predicting the content of heavy metals at unknown points in soil. A comparative experiment was conducted using the SE-GCN, GCN, DCNN, Kriging, and RBF models. The experiment showed that, incorporating spatial autocorrelation can alleviate most of the overfitting issues in the model. Furthermore, the SE-GCN model yields lower values for REMS, MAE, and R^2 in predicting Hg and Ni compared to the GCN, DCNN, Kriging, and RBF models. Additionally, the absolute prediction error of this model is smaller than that of the GCN, DCNN, Kriging, and RBF models. Thus, the SE-GCN model had higher accuracy and better performance in predicting the heavy-metal content in soil. An application analysis of the two improved neural network models was carried out; their application scenarios and applicability were analyzed, indicating that both models have practical application value in predicting soil pollution.

Author Contributions: Conceptualization, Z.W. and W.Z.; methodology, Z.W., W.Z. and Y.H.; software, W.Z. and Y.H.; validation, Z.W., W.Z. and Y.H.; formal analysis, Z.W. and W.Z.; investigation, W.Z. and Y.H.; resources and data curation, W.Z. and Y.H.; writing—original draft preparation, W.Z. and Y.H.; writing—review and editing, Z.W. and W.Z.; visualization, W.Z. and Y.H.; supervision, project administration and funding acquisition, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key R&D Program of China (2018YFC1800203) and The Scientific Research Project of Beijing Educational Committee (KM201811232010).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Akhtar, M.N.; Shaikh, A.J.; Khan, A.; Awais, H.; Bakar, E.A.; Othman, A.R. Smart sensing with edge computing in precision agriculture for soil assessment and heavy metal monitoring: A review. *Agriculture* **2021**, *11*, 475. [CrossRef]
2. Wei, J.; Kong, H.; Fan, W. Application of Big Data in the Remediation of Contaminated Sites. *Asian Agric. Res.* **2021**, *13*, 39–40.
3. Li, Y.; Cao, Z.; Long, H.; Liu, Y.; Li, W. Dynamic analysis of ecological environment combined with land cover and NDVI changes and implications for sustainable urban–rural development: The case of Mu Us Sandy Land, China. *J. Clean. Prod.* **2017**, *142*, 697–715. [CrossRef]
4. Shao, J.; Meng, W.; Sun, G. Evaluation of missing value imputation methods for wireless soil datasets. *Pers. Ubiquit. Comput.* **2017**, *21*, 113–123. [CrossRef]
5. Freeman, B.S.; Taylor, G.; Gharabaghi, B.; Thé, J. Forecasting air quality time series using deep learning. *J. Air Waste Manag. Assoc.* **2018**, *68*, 866–886. [CrossRef]
6. Deng, W.; Wang, G.; Zhang, X. A novel hybrid water quality time series prediction method based on cloud model and fuzzy forecasting. *Chemom. Intell. Lab. Syst.* **2015**, *149*, 39–49. [CrossRef]
7. Park, J.; Müller, J.; Arora, B.; Faybishenko, B.; Pastorello, G.; Varadharajan, C.; Sahu, R.; Agarwal, D. Long-term missing value imputation for time series data using deep neural networks. *Neural Comput. Appl.* **2023**, *35*, 9071–9091. [CrossRef]
8. Kayid, M. One Generalized Mixture Pareto Distribution and Estimation of the Parameters by the EM Algorithm for Complete and Right-Censored Data. *IEEE Access* **2021**, *9*, 149372–149382. [CrossRef]
9. Bashir, F.; Wei, H.L. Handling missing data in multivariate time series using a vector autoregressive model-imputation algorithm. *Neurocomputing* **2018**, *276*, 23–30. [CrossRef]
10. Gondara, L.; Wang, K. Mida: Multiple imputation using denoising autoencoders. In *Advances in Knowledge Discovery and Data Mining, Proceedings of the 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, 3–6 June 2018*; Springer International Publishing: Cham, Switzerland, 2018; pp. 260–272. [CrossRef]
11. Rizal, N.N.M.; Hayder, G.; Mnzool, M.; Elnaim, B.M.E.; Mohammed, A.O.Y.; Khayyat, M.M. Comparison between regression models, support vector machine (SVM), and artificial neural network (ANN) in river water quality prediction. *Processes* **2022**, *10*, 1652. [CrossRef]
12. Wang, H.; Yuan, Z.; Chen, Y.; Shen, B.; Wu, A. An industrial missing values processing method based on generating model. *Comput. Netw.* **2019**, *158*, 61–68. [CrossRef]
13. Cao, W.; Zhang, C. A collaborative compound neural network model for soil heavy metal content prediction. *IEEE Access* **2020**, *8*, 129497–129509. [CrossRef]
14. Yin, G.; Chen, X.; Zhu, H.; Chen, Z.; Su, C.; He, Z.; Qiu, J.; Wang, T. A novel interpolation method to predict soil heavy metals based on a genetic algorithm and neural network model. *Sci. Total Environ.* **2022**, *825*, 153948. [CrossRef]
15. Ma, W.; Tan, K.; Du, P. Predicting soil heavy metal based on Random Forest model. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 4331–4334.
16. Gao, Z.Y.; Xiao, R.B.; Wang, P.; Deng, Y.R.; Dai, W.J.; Liu, C.F. Improved Regression Kriging Prediction of the Spatial Distribution of the Soil Cadmium by Integrating Natural and Human Factors. *Huan Jing Ke Xue Huanjing Kexue* **2021**, *42*, 343–352. [PubMed]
17. Niu, J.; Zhong, W.; Liang, Y.; Luo, N.; Qian, F. Fruit Fly Optimization Algorithm Based On Differential Evolution and Its Application on Gasification Process Operation Optimization. *Knowl.-Based Syst.* **2015**, *88*, 253–263. [CrossRef]
18. Dowsland, K.A.; Thompson, J. Simulated Annealing. *Handb. Nat. Comput.* **2012**, *43*, 1623–1655.
19. Kirkpatrick, S.; Vecchi, M.P. Optimization by Simulated Annealing. In *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications*; World Scientific Publishing Company: Singapore, 1987.
20. Iscan, H.; Gunduz, M. Parameter Analysis on Fruit Fly Optimization Algorithm. *J. Comput. Commun.* **2016**, *2*, 137–141. [CrossRef]
21. Deng, Y.; Zhou, X.; Shen, J.; Xiao, G.; Hong, H.; Lin, H.; Wu, F.; Liao, B.Q. New Methods Based on Back Propagation (BP) and Radial Basis Function (RBF) Artificial Neural Networks (ANNs) for Predicting the Occurrence of Haloketones in Tap Water. *Sci. Total Environ.* **2021**, *772*, 145534. [CrossRef]
22. Zhang, Y.; Du, D.; Shi, S.; Li, W.; Wang, S. Effects of the Earthquake Nonstationary Characteristics on the Structural Dynamic Response: Base on the BP Neural Networks Modified by the Genetic Algorithm. *Buildings* **2021**, *11*, 69. [CrossRef]
23. Wu, Q. Image retrieval method based on deep learning semantic feature extraction and regularization softmax. *Multimed Tools Appl.* **2020**, *79*, 9419–9433. [CrossRef]
24. Shi, K.; Chang, Z.; Chen, Z.; Wu, J.; Yu, B. Identifying and evaluating poverty using multisource remote sensing and point of interest (POI) data: A case study of Chongqing, China. *J. Clean. Prod.* **2020**, *255*, 120245. [CrossRef]
25. Danel, T.; Spurek, P.; Tabor, J.; Śmieja, M.; Struski, Ł.; Słowik, A.; Maziarka, Ł. Spatial graph convolutional networks. In Proceedings of the International Conference on Neural Information Processing, Bangkok, Thailand, 18–22 November 2020; Springer International Publishing: Cham, Switzerland, 2020; pp. 668–675.

26. Wang, F.; Li, C.T.; Qu, Y. Collective Geographical Embedding for Geolocating Social Network Users. In *Advances in Knowledge Discovery and Data Mining, Proceedings of the 21st Pacific-Asia Conference, PAKDD 2017, Jeju, Republic of Korea, 23–26 May 2017*; Springer International Publishing: Cham, Switzerland, 2017; pp. 599–611.
27. Dwivedi, V.P.; Luu, A.T.; Laurent, T.; Bengio, Y.; Bresson, X. Graph neural networks with learnable structural and positional representations. *arXiv* **2021**, arXiv:2110.07875.
28. Kumar, H.S.; Manjunath, S.H. Use of empirical mode decomposition and K-nearest neighbour classifier for rolling element bearing fault diagnosis. *Mater. Today Proc.* **2022**, *52*, 796–801. [CrossRef]
29. Zhu, D.; Liu, Y.; Yao, X.; Fischer, M.M. Spatial regression graph convolutional neural networks: A deep learning paradigm for spatial multivariate distributions. *GeoInformatica* **2021**, *26*, 645–676. [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.