Research Article

Hybrid Model of Machine Learning Refractory Data Prediction Based on IoT Smart Cities

Xuewei Li, Kai Huang, and Lei Xu

School of Civil Engineering, Dalian Minzu University, Dalian, 116650 Liaoning, China

Correspondence should be addressed to Lei Xu; xulgb@163.com

Received 18 August 2021; Revised 5 November 2021; Accepted 2 December 2021; Published 23 February 2022

Academic Editor: Han Liu

Copyright © 2022 Xuewei Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the advent of the digital age in recent years, the application of artificial intelligence in urban Internet of Things (IoT) systems has become increasingly important. The concept of smart cities has gradually formed, and smart firefighting under the smart city system has also become important. The method of machine learning is now applied in various fields, but seldom to the data prediction of smart firefighting. Various types of applications including data applications of machine learning algorithms in smart firefighting have yet to be explored. In this article, we propose using machine learning algorithms to predict building fire-resistance data, aiming to provide more theoretical and technical support for IoT smart cities. This article adopts the fire-resistance data of building beam components in a real fire environment, using three integrated machine learning algorithms, Extreme random Tree (ET), AdaBoost, and Gradient Boosting Machine (GBM), and the grey wolf optimization algorithm to optimize. We improve the grey wolf algorithm and combine the grey wolf algorithm with the machine learning model. The algorithm constitutes three machine learning hybrid models: GWO-ET, GWO-AdaBoost, and GWO-GBM. Compared with traditional grid tuning, particle swarm optimization (PSO), and genetic algorithm (GA) optimization, the robustness and accuracy of the three optimization algorithms and the machine learning hybrid algorithm on the data set are compared and analyzed. Performance is measured through various performance comparisons and experimental result comparisons. For various building beam component data sets under real fires, the optimization and comparison show that the mean square error (MSE) of the proposed algorithm is extremely small. The results indicate that the GWO machine learning hybrid model is superior to other models and has a smaller prediction error.

1. Introduction

With the continuous development of modern smart cities, flammable building materials have increased the risk of fires that threaten lives and livelihoods, so the application of smart firefighting in urban Internet of Things (IoT) artificial intelligence has become more important. Machine learning technology is widely used in various fields, such as data mining, image processing, intelligent transportation, smart cities, medical health, intelligent prediction, and the IoT. Machine learning prediction is used in many fields, such as the prediction of stocks in the financial sector; the prediction of biological information, such as breast cancer classification and image recognition; the use of big data to mine social media data; and recommendation prediction in e-commerce. With the rapid development of artificial intelligence,

the use of algorithms to predict related data has also become the top priority of smart fire protection and related fields under the urban IoT.

In [1, 2], the fire-resistance performance of most components in a fire is calculated by numerical, empirical, and computational analysis methods. In [3, 4], the fire-resistance limit is determined by the degree of damage deflection of the component. In this article, we use numerical analysis of fire-resistance data to conduct supervised learning to predict the fire-resistance limit of components. In previous studies, machine learning algorithms have been used to predict the fire-resistance data of building components. In [5], the artificial neural network (ANN) was used to predict the fire-resistance performance of concrete-clad steel composite columns. The predictive ability of the ANN was reasonable. The proposed ANN model was also better

than analytical equations in terms of prediction accuracy, indicating that the application of machine learning algorithms to the prediction of fire-resistance limit data for smart fire protection has a certain theoretical basis and higher accuracy. In [6], an integrated model was proposed to predict the bearing capacity of composite honeycomb steel beams in a fire, using gene expression programming (GEP), multiple linear regression (MLR), and principal component regression (PCR). The experimental results show that the combination of GEP and MLR was considered to predict the best model of CCSB carrying capacity. The above prediction was based on the algorithm and used the ambient temperature of the international standard temperature. In this article, the real fire environment temperature simulated by FDS is used, and the data are simulated and calculated using the ABAQUS software.

In [7], a pavement technical condition index attenuation prediction model based on LightGBM (Light Gradient Boost Machine) was proposed, which proved to be reliable and practical under the conditions of highway pavement. In [8], the ultimate gradient boosting (XGBoost) algorithm was used to predict the bearing capacity of fiber reinforced polymer (FRP) reinforced concrete (RC) columns. The prediction error was relatively low, but the network search method was used for optimization. We try to use the grey wolf optimizer (GWO) optimization method for Extreme random Tree (ET), AdaBoost, and Gradient Boosting Machine (GBM), three algorithms used to predict refractory data. In [9], a decomposition method based on the long short-term memory (LSTM) network and GWO was proposed to develop a new wind speed prediction hybrid model. In [10], the Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) was proposed for use to accurately predict the output power of photovoltaic systems. In [11], it was proposed to apply grey wolf optimization and PSO to predict rheumatoid arthritis data, and the accuracy rate improved greatly. In [12], a grey wolf optimizer composed of a new hybrid algorithm was introduced to evaluate the parameters of SVM; it was used for power system load forecasting and was compared with the prediction performance of PSO-SVM and GA-SVM, which proved the performance of the provided method. In [13], a GWO-SVM method was proposed to establish a CO detection correction model in the process of coal combustion loss in coal mines to correct the CO concentration measurement. The results show that the GWO-SVM model had higher accuracy and stability than other models. In the latest development of GWO, in [14], GWO introduced a search strategy called Representation-Based Hunting (RH) to form the R-GWO algorithm, which is a representation-based grey wolf optimizer for solving engineering problems. In [15], Gaussian walk and Lévy flight were used to improve the exploration and development capabilities of GGWO and predict the COVID-19 pandemic in the United States. Here, we use GWO optimization for Extreme random Tree (ET), AdaBoost, and Gradient Boosting Machine (GBM) hyperparameter tuning of three machine learning algorithms and add a set of random numbers to the cycle into the settings and strategies to transform the data form, make it more suitable for the model, reach the

search faster, and compare and analyze its accuracy with PSO algorithm and GA optimization in refractory data prediction.

In recent years, the IoT has emerged in various fields and become the main core of Industry 4.0. In [16], an effective technique called SIOMT (Swarm Intelligence optimization technique for the IoMT) is proposed in this paper for periodically discovering, clustering, analyzing, and managing useful data about potential patients. Rarely, only ANNs, linear regression, PCA, and other algorithms are not used in integrated algorithms, nor are they used in optimization algorithms. This article will use three integrated machine learning algorithms, Extreme random Tree (ET), AdaBoost, and Gradient Boosting Machine (GBM), and will apply the grey wolf optimizer to the machine learning prediction model of refractory data to form a machine learning hybrid model. This represents new exploration in the field of smart cities and smart fire protection as well as a new test of whether machine learning algorithms can be used in the prediction of fire-resistance data.

The main research contents of this article include the following:

- (1) After obtaining the fire-resistance limit data set through the ABAQUS software, we perform data preprocessing on it and select three machine learning integration algorithms, Extreme random Tree (ET), AdaBoost, and Gradient Boosting Machine (GBM), to make a prediction on the data set through algorithm evaluation
- (2) The grey wolf optimization algorithm is improved to form a machine learning hybrid model and optimize the training of the data set
- (3) The results and performance of PSO, GA, and grid tuning are compared, and their effects and accuracy are analyzed

The main contributions and novelties of this article are as follows.

- (1) The refractory data at ambient temperature based on an FDS real fire simulation is used for machine learning modeling, including Extreme random Tree (ET), AdaBoost, and Gradient Boosting Machine (GBM)
- (2) The grey wolf optimizer is used for algorithm prediction and to improve the grey wolf optimization algorithm; a set of random numbers is used to cycle into the judgment of the optimal solution, improve the data input form and iterative method, and make it suitable for the hyperparameters of the selected algorithm
- (3) The grey wolf optimization algorithm and machine learning model are combined, and the optimal parameters obtained by the grey wolf optimization algorithm are substituted into the machine learning model to obtain three machine learning hybrid

models: GWO-ET, GWO-AdaBoost, and GWO-GBM

- (4) Comparing the performance of PSO, GA, and traditional optimization on different refractory data sets, it is proved that the grey wolf optimizer optimizes the prediction effect, greatly reduces its errors, and improves the accuracy

2. Research Methods Using Machine Learning Algorithms to Predict Refractory Data

In this study, we first select three ensemble algorithms for evaluation, Extreme random Tree (ET), AdaBoost, and Gradient Boosting Machine (GBM), and use the grey wolf optimizer to optimize the prediction model.

2.1. Integrated Algorithm

(1) ET algorithm

The ET is composed of many decision trees, including root nodes, internal nodes, and leaf nodes, representing a mapping relationship between object attributes and object values. The leaf node corresponds to the decision result, and the route from the root node to one of the leaf nodes is the entire decision process of the class corresponding to the current leaf node.

The decision tree obtains the entropy by using the algorithm ID3, C4.5.

(a) Introduction to ID3 algorithm

The ID3 algorithm calculates the information gain each time it splits and then selects the largest information gain for standard splitting. D is the sample set before the current decision split, K is the sample category, and the information entropy is defined as follows:

$$\text{Ent}(D) = - \sum_{k=1}^K p_k \log_2 p_k. \quad (1)$$

By calculating the information gain, the sample is split according to the attribute set to output a subset of each attribute sample, and then, the information entropy of a single subset is calculated according to the category of each sample subset:

$$\text{Ent}(D_V) = - \sum_{k=1}^K p_k \log_2 p_k. \quad (2)$$

Afterward, the weight of each sample subset is assigned, and the calculation is performed according to the ratio of the number of samples at each branch node to the total number of samples. The following is the calculation of information gain:

$$\text{Gain}(D, a) = \text{Ent}(D) - \sum_{v=1}^V \frac{D_v}{D} \text{Ent}(D_v). \quad (3)$$

(b) Introduction to C4.5 algorithm

The C4.5 algorithm divides the optimal attribute using the gain rate. The formula for the gain rate is as follows, where the larger a is, the larger V and IV are.

$$\begin{aligned} \text{Gain}_{\text{ratio}(D,a)} &= \frac{\text{Gain}(D, a)}{IV(a)}, \\ IV(a) &= - \sum_{v=1}^V \frac{|D_v|}{|D|} \log_2 \frac{|D_v|}{|D|}. \end{aligned} \quad (4)$$

(c) Gini index

The purity of the data set can be judged not only by information entropy but also by the Gini index. The following is the formula for the Gini index:

$$\text{Gini}(D) = \sum_{k=1}^K \sum_{k^* \neq k} p_k^+ p_{k^*}^- = 1 - \sum_{k=1}^K p_k^2. \quad (5)$$

$\text{Gini}(D)$ reflects that when two samples are randomly selected, the probability of a different category is different. At the same time, the smaller the $\text{Gini}(D)$, the higher the purity of the sample. Therefore, the formula for the Gini index of attribute a is as follows. Use the Gini index to calculate the splitting of the node, which is the process of calculating $a_v = \text{argmin Gini_index}(D, a)$.

$$\text{Gini}_{\text{index}(D,a)} = \sum_{v=1}^V \frac{|D_v|}{|D|} \text{Gini}(D_v). \quad (6)$$

(2) AdaBoost algorithm

Input training data, and $T = \{\{x_1, y_1\}, \{x_2, y_2\}, \dots, \{x_N, y_N\}\}$. Among them, $x_i \in X, y_i \in Y = -1, 1$, and the number of iterations is M . The weight distribution of the initial training sample is $D_1 = (\omega_{1,1}, \omega_{1,2}, \dots, \omega_{1,i}), \omega_{1,i} = 1/N, i = 1, 2, \dots, N$. For $m = 1, 2, \dots, M$,

- (a) Use the training data set with the weight distribution D_m to learn and obtain the weak classifier $C_m(x)$
- (b) Calculate the classification error rate of $C_m(x)$ on the training data set

$$e_m = \sum_{i=1}^N \omega_{m,i} I(G_m(x_i) \neq y_i). \quad (7)$$

(c) Calculate the weight of $G_m(x)$ in the strong classifier

$$\alpha_m = \frac{1}{2} \log \frac{1 - e_m}{e_m}. \quad (8)$$

(d) Update the weight distribution of the training data set, where z_m is the normalization factor, so that the sum of the probability distribution of the sample is 1

$$\begin{aligned} \omega_{m+1,i} &= \frac{\omega_{m,i}}{z_m} \exp(-\alpha_m y_i G_m(x_i)), i = 1, 2, \dots, 10, \\ z_m &= \sum_{i=1}^N \omega_{m,i} \exp(-\alpha_m y_i G_m(x_i)). \end{aligned} \quad (9)$$

(e) Obtain the final classifier

$$F(x) = \text{sign} \left(\sum_{i=1}^N \alpha_m G_m(x) \right). \quad (10)$$

(3) GBM algorithm

GBM is based on the CART regression tree and gradient boosting tree.

Initialize the weak learner. For $m = 1, 2, \dots, M$,

(a) For each sample $i = 1, 2, \dots, N$, calculate the negative gradient, that is, the residual

$$\gamma_{im} = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f(x)=f_{m-1}(x)}. \quad (11)$$

(b) Use the residual obtained in the previous step as the new true value of the sample, and use the data (x_i, γ_{im}) , $i = 1, 2, \dots, N$ as the training data of the next tree to obtain a new regression tree, $f_m(x)$. Its corresponding leaf node area is R_{jm} , $j = 1, 2, \dots, J$, where J is the number of leaf nodes of the regression tree t

(c) Calculate the best fit value for the leaf area $j = 1, 2, \dots, J$

$$\gamma_{jm} = \underset{x_i \in R_{jm}}{\text{argmin}} \sum L(y_i, f_{m-1}(x_i) + \gamma). \quad (12)$$

(d) Calculate the best fit value for the leaf area $j = 1, 2, \dots, J$

$$f_m(x) = f_{m-1}(x) + \sum_{j=1}^J \gamma_{jm} I(x \in R_{jm}). \quad (13)$$

Obtain the final learner

$$f(x) = f_M(x) = f_0(x) + \sum_{m=1}^M \sum_{j=1}^J \gamma_{jm} I(x \in R_{jm}). \quad (14)$$

2.2. Grey Wolf Optimizer. The grey wolf optimizer (GWO) is a population intelligence optimization algorithm proposed by Griffith University scholar Mirjalili and others in 2014. The algorithm was inspired by the hunting activities of grey wolves and then developed into an optimized search method. It has the characteristics of strong convergence performance, few parameters, and easy implementation. In recent years, it has received extensive attention from scholars and has been successfully applied to workshop scheduling, parameter optimization, image classification, and other fields. Usually, GWO is used for SVM optimization, but in this article, the characteristics of the algorithm are used to optimize search so as to optimize the hyperparameters of the integrated algorithm.

Algorithm principle

There are three wolves, α , β , and γ , in a wolf pack. α is the wolf king, and β and γ are ranked second and third, respectively. Both β and γ listen to α , and γ listens to β . The three wolves guide the other wolves in search of prey. The process of wolves looking for prey represents the process of the GWO algorithm finding the optimal solution. The GWO optimization process also includes social hierarchical stratification, tracking, encircling and attacking prey, and searching for prey. However, its core behavior is only hunting, that is, finding the optimal solution. During each iteration, the best three wolves, α , β , and γ , in the population will be retained, and the positions of other search agents will be updated according to their position information.

(1) Surround the prey

The grey wolf will gradually approach and surround its prey when hunting. Its mathematical model is as follows [17]:

$$\begin{aligned} D &= C_0 X_p(t) - X(t), \\ X(t+1) &= X_p(t) - A_0 D, \\ A &= 2ar_1 - a, \\ C &= 2r_2. \end{aligned} \quad (15)$$

FIGURE 1: Correlation matrix diagram description.

Here, t is the current iteration number, A and C are the coordination coefficient vectors, X_p is the position vector of Lei Wu, $X(t)$ is the current position vector of the grey wolf, and r_1 and r_2 are random vectors between $[0,1]$.

(2) Hunting behavior

Grey wolves have the ability to identify the position of potential prey (optimal solution), that is, to exhibit hunting behavior. The mathematical model is as follows:

$$D_\alpha = C_1 X_\alpha - X, D_\beta = C_2 X_\beta - X, D_\delta = C_3 X_\delta - X, \quad (16)$$

$$X_1 = X_\alpha - A_1 D_\alpha, X_2 = X_\beta - A_2 D_\beta, X_3 = X_\delta - A_3 D_\delta, \quad (17)$$

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

Here, X_α , X_β , and X_δ represent the position vector of α , β , and δ , respectively, in the current population; D_α , D_β , and D_δ represent the distance between the current candidate wolf pack and the best three wolves, respectively; when $|A| > 1$, grey wolves will scatter in various areas to look for prey. When $|A| < 1$, grey wolves will focus on one or some areas to look for prey [17].

- 1: Read CSV data
- 2: Build tag
- 3: Perform data preprocessing
- 4: Separate data sets
- 5: Normalize data
- 6: Call GWO algorithm and objective function

ALGORITHM 1: Model 1 steps: GWO-ET model.

(3) Attack the prey

The attacking prey will also decrease according to the decrease of the value of a , the random vector on $A \in [-a, a]$, where a decreases linearly in the iterative process. When $A \in [-1, 1]$ is in the interval, the next time position of the search agent can be anywhere between the current grey wolf and the prey.

(4) Looking for prey

Grey wolves rely on α , β , and δ to find their prey. They start to search for prey location information in a scattered manner and then gather to attack the prey and complete this optimization through iteration [17].

(5) Improvement of GWO algorithm

Input: objective function, Parameter adjustment range of super parameter, Xtrain, Ytrain, Xtest, Ytest, PopSize, Niter
Output: The global optimum (gbest)

```

1: Procedure GWO
2: Initialize input parameters, alpha, beta, delta, All individual samples
3: # Iterative optimization, # main loop of the algorithm
4: For l=1: Niter
5:   For i=1:PopSize
6:     Call objective function, Generates a set of random numbers in the hyperparametric range
7:     Update the optimal location of Alpha, Beta, and Delta
8:     Update  $X_1, X_2, X_3$  by equation (18)
9:     I=i+1
10:   End for
11:   l=l+1
12: End for
13: Return the global optimum (gbest)
14: End Procedure

```

ALGORITHM 2: Optimized GWO.

Input: Over parameter adjustment range, Xtrain, Ytrain, Xtest, Ytest
Output: MSE

```

1: Procedure objective function
2: For i in A set of random numbers # Cyclic substitution
3: i data conversion # int, float or other more complex forms
4: Substitute in ExtraTreesRegressor
5: Train
6: Calculate MSE by formula (19)
7: Return MSE
8: End Procedure

```

ALGORITHM 3: Objective function setting.

TABLE 1: Data set 1/2 parameter selection range.

Parameter	Tuning range
n_estimators	(1, 150)
max_depth	(20, 30)
max_features	["auto," "sqrt"]
min_samples_split	[2, 5, 10]
min_samples_leaf	[1, 2, 4]
Bootstrap	[true, false]

TABLE 2: Data set 3 parameter selection range.

Parameter	Tuning range
n_estimators	(1, 50)
max_depth	(1, 21)
Subsample	(0.1,0.9)
Learning_rate	(0.1, 1.1)
min_samples_split	(2, 11)
min_samples_leaf	(1, 11)

TABLE 3: MSE changes before and after tuning.

Model	MSE changes before	RandomizedSearchCV MSE	GWO optimized MSE
Data set 1	0.0022	0.0021	0.0015
Data set 2	0.093	0.0447	0.0115
Data set 3	0.2300	0.2013	0.0420

Using the GWO algorithm to scatter the grey wolf to find the prey and then focus on the area to find the prey, the optimal solution is that it gradually approaches the optimal solution target, continuously narrows the search range, and finally finds the optimal solution through iteration. We narrow the scope to find the prey by improving the GWO algorithm, apply it to the form of the machine learning algorithm to adjust the hyperparameter, and add a set of random numbers in each round of iterations and looping to determine the ET, AdaBoost, and GBM super parameter. At the same time, because the GWO is set according to dimensions, its data form is converted into a form suitable for algorithm hyperparameters. Then, the optimal solution is further

FIGURE 2: Data set fitting effect.

searched, and the optimal parameter is found with the smallest MSE.

Unlike the previous GWO algorithm,

- 1) When looking for a local optimal solution, we use a set of random numbers to cycle into and judge the optimal solution and perform related data format conversions to apply the hyperparameters of the algorithm
- 2) We combine the improved GWO algorithm and the machine learning model to obtain the GWO optimized machine learning prediction model

2.3. Evaluation Index. We use mean squared error (MSE) as the evaluation index. The formula of MSE is as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^m \omega_i (y_i - \hat{y}_i)^2, \quad (19)$$

where n is the number of samples, y_i is the real predicted data, and $\hat{y}_i$ is the fitted predicted data.

3. Refractory Data Prediction Model Based on Machine Learning

3.1. Data Collection. In this article, the numerical simulation software ABAQUS is used to establish the model to obtain the data. For the three beams, finite element models with different cross-sections (BC1 and BC2), different load ratios (Q), and different protective layer thicknesses (C) are established. For steel beams, different waist heights (h) are established. The finite element model of waist thickness includes the (b), leg width (b), upper flange thickness ($z1$), and lower flange thickness ($z2$). The steel is Q345 steel, the strength grade of the concrete is C35, and the absolute zero is set to what. The fire condition is that the beam is exposed to fire on three sides, and the back of the upper flange of the steel beam and below the upper flange are all exposed to fire.

The heat radiation coefficient of the fire surface is 0.5, the radiation coefficient distribution is consistent, and the surface heat exchange film has a heat dissipation coefficient of 25. We import the real FDS simulation and set the corresponding analysis step and time for the fire environment temperature.

Then, we import the result database file obtained from the temperature field model into the predefined field of the mechanical model, set different loads according to the established model, apply uniformly distributed loads on the upper surface of the beam through pressure, set the boundary conditions and constraints, and perform finite element calculations on each model. We obtain various beam deformations and beam midspan deflection. From the mechanical model of each beam, we obtain the change of beam midspan deflection with time during the fire. According to "Standard for Fire Test of Building Components" (GB/T9978-2008) [18], when the maximum deflection of the beam reaches the deflection, and the deformation exceeds $L/30$ (mm) (L is the calculated span of the beam and slab), the beam can be judged to have reached the fire-resistance limit. The data set required by the fire-resistance limit prediction model is thus obtained.

3.2. Data Processing. Through the previous finite element modeling, the different parameters of the beam and the corresponding fire-resistance limit are obtained, which constitutes the data set required for this research. The training data set is divided by 80%, the evaluation data set is divided by 20%, and the data set is divided by what. We perform unified processing of the format and unit. Figure 1 shows a description of the correlation matrix diagrams of the three data sets.

Standardization of data is normalization. Data standardization is based on the columns of the feature matrix. The attributes of the data are converted to obey the normal distribution, and the mean and variance are normalized; that is, the mean is 0, and the standard deviation is 1. Its transformation function is as follows:

$$x^* = \frac{x - \mu}{\sigma}. \quad (20)$$

Among them, μ is the mean of all sample data, and σ is the standard deviation of all sample data.

3.3. Model Steps and Optimization. After preprocessing the data set, we begin to build a grey wolf optimized machine learning hybrid model. Data set 1 uses ET, data set 2 uses AdaBoost, and data set 3 uses GBM for training. To make the machine learning model better for learning and hyperparameter tuning, based on the principal formula of Section 2.2, we perform the setting update of the objective function and the data form conversion and compose it into every 10 ones. The random numbers of the group are cyclically substituted into the machine learning model for learning, and then, the grey wolf optimization population and iterative optimization are performed. We form the grey wolf before encircling and hunting, and it loops through the ran-

TABLE 4: GWO algorithm, PSO algorithm, and GA optimization comparison.

MSE	Data set 1	Data set 2	Data set 3
GWO optimized MSE	0.0015	0.0115	0.0420
PSO optimized MSE	0.0015	0.0149	0.0525
GA optimized MSE	0.0017	0.0525	0.0643

FIGURE 3: GWO algorithm, PSO algorithm, and GA optimization MSE comparison diagram.

dom array first and then encloses the optimal value of the objective function to increase the speed and probability of encircling the prey. The specific steps of this model are shown in Algorithm 1, model 1. Taking the GWO-ET model as an example, the pseudo code of the grey wolf optimization algorithm is shown in Algorithm 2. The basic grey wolf optimization algorithm of this code refers to [19–22]. The setting of the objective function is shown in Algorithm 3. PopSize is the population size, and Niter is the number of iterations.

4. Model Verification and Comparison

4.1. Comparison of Grid Tuning. In this section, we use RandomizedSearchCV grid tuning and grey wolf optimization parameters to compare the models. Data set 1 uses ET, data set 2 uses AdaBoost, and data set 3 uses GBM. For training, the range of grid adjustment parameters is shown in Tables 1 and 2. We set the objective function of the GWO algorithm; the population size is 20, the number of iterations is 100, and the algorithms are compared. Before tuning, the comparison results of grid tuning and GWO optimization MSE are as shown in Table 3.

Figure 2 shows the fitting effect after GWO optimization of three data sets. The red line represents the predicted value, and the green line represents the true value. As shown, the models optimized by the grey wolf optimizer greatly reduce the MSE, and the fitting effect is also better.

4.2. Comparison with PSO Algorithm and GA. In this section, we compare the GWO machine learning hybrid model with the PSO and the GA machine learning hybrid model. The population size is set to 20, and the number of iterations

FIGURE 4: Convergence process comparison.

is 100, as shown in Table 4 and Figure 3. In the comparison of the optimization algorithm MSE of GWO, PSO, and GA, the GWO optimization MSE is the smallest, followed by the PSO model.

4.3. Convergence Process Comparison. After GWO quickly finds the optimal solution by narrowing the scope, iterative optimization is faster and more stable, and the convergence process is better. PSO simulates the birds in a flock of birds through a flock of particles, initializes it as a flock of random particles, and then iteratively optimizes. GA is suitable for multivariable, multiobjective complex optimization, but it is also prone to premature convergence. Figure 4 shows the comparison of the convergence process of GWO, PSO, and GA. The red line `fval_best` represents the optimized target value, and the blue line `fval_mean` represents the mean value. GWO narrows down the range after quickly finding a smaller value to finally enclose the best value. The convergence process of PSO and GWO is similar, while the convergence process of GA is relatively unstable and is not suitable for refractory data sets, more complex nonlinear function relations, or multiobjective optimization.

4.4. Calculation Performance Comparison. Although the convergence process of GWO is relatively stable, it consumes more calculation time owing to the number of iterations. However, although it does not require many iterations to find the optimal parameters, it is CPU-intensive. PSO and GA save more time. Under the same population size and iteration number of the GWO algorithm, PSO algorithm, and GA, the calculation time of

PSO and GA is half that of GWO. Therefore, the GWO optimization algorithm saves calculation time and has faster convergence speed and more accurate optimization compared with grid tuning. Compared with PSO, the GA has a longer calculation time, but the optimization is more accurate, and the convergence is more stable.

5. Conclusion

This article mainly uses machine learning algorithms to predict three fire-resistant data sets. The fire-resistant data obtained under the real fire environment temperature based on FDS simulation is preprocessed, and ET, AdaBoost, and Gradient Boosting Machine (GBM) are combined with the grey wolf optimizer to form a hybrid prediction model and improve the grey wolf optimization algorithm. It uses a set of random numbers to cycle into and judge the optimal solution and performs related data format conversions to apply the hyperparameters of the hybrid model algorithm. The improved optimization algorithm hybrid model proposed in this article can greatly improve the accuracy of predicting the refractory data of the IoT. After comparing with the grid parameter adjustment, optimization algorithm PSO, and GA hybrid model, the GWO machine learning hybrid model is used in the prediction of refractory data and has a higher accuracy rate. The shortcomings of this article are that the research model is not perfect, more experimental data are needed, the calculation time of GWO calculation is longer than other optimization algorithms, and it needs to be optimized. In the future, artificial intelligence technology based on IoT smart cities will become more widespread, and the

application of machine learning algorithms for smart fire protection, big data prediction, and other smart cities will increase. These applications will require us to conduct more in-depth research.

Data Availability

The data underlying the results presented in the study are available within the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgments

The research reported in the paper is part of the Project 52178461 supported by the National Natural Science Foundation of China and the Project LJYT201908 supported by the Foundation of Department of Education of Liaoning Province. The financial support is highly appreciated.

References

- [1] S. F. El-Fitiany and M. A. Youssef, "Assessing the flexural and axial behaviour of reinforced concrete members at elevated temperatures using sectional analysis," *Fire Safety Journal*, vol. 44, pp. 691–703, 2009.
- [2] M. B. Dwaikat and V. K. R. Kodur, "Response of restrained concrete beams under design fire exposure," *Journal of Structural Engineering*, vol. 135, pp. 1408–1417, 2009.
- [3] M. B. Dwaikat and V. K. R. Kodur, "A numerical approach for modeling the fire induced restraint effects in reinforced concrete beams," *Fire Safety Journal*, vol. 43, pp. 291–307, 2008.
- [4] V. K. R. Kodur and M. Dwaikat, "Flexural response of reinforced concrete beams exposed to fire," *Structural Concrete*, vol. 9, pp. 45–54, 2008.
- [5] J. Y. Shan Li, R. Liew, and M.-X. Xiong, "Prediction of fire resistance of concrete encased steel composite columns using artificial neural network," *Engineering Structures*, vol. 245, 2021.
- [6] B. Karami, A. Shishegaran, H. Taghavizade, and T. Rabczuk, "Presenting innovative ensemble model for prediction of the load carrying capacity of composite castellated steel beam under fire," *Structures*, vol. 33, pp. 4031–4052, 2021.
- [7] L. Pei, T. Yu, L. Xu, W. Li, Y. Han, and X. Li, "Prediction of Decay of Pavement Quality or Performance Index Based on Light Gradient Boost Machine," *Advances in Intelligent Automation and Soft Computing*, vol. 80, pp. 1173–1179, 2022.
- [8] A. S. Bakouregui, H. M. Mohamed, A. Yahia, and B. Benmokrane, "Explainable extreme gradient boosting tree-based prediction of load-carrying capacity of FRP-RC columns," *Engineering Structures*, vol. 245, 2021.
- [9] A. Altan, S. Karasu, and E. Zio, "A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer," *Applied Soft Computing*, vol. 100, p. 106996, 2021.
- [10] M. Abdel-Nasser, "Accurate photovoltaic power forecasting models using deep LSTM-RNN," *Neural Computing and Applications*, vol. 31, 2019.
- [11] S. Sundaramurthy and P. Jayavel, "A hybrid grey wolf optimization and particle swarm optimization with C4.5 approach for prediction of rheumatoid arthritis," *Applied Soft Computing*, vol. 94, 2020.
- [12] M. Barman, D. Choudhury, and N. Behari, "A similarity based hybrid GWO-SVM method of power system load forecasting for regional special event days in anomalous load situations in Assam, India," *Sustainable Cities and Society*, vol. 61, 2020.
- [13] J. Deng, W. L. Chen, C. Liang et al., "Correction model for CO detection in the coal combustion loss process in mines based on GWO-SVM," *Journal of Loss Prevention in the Process Industries*, vol. 71, 2021.
- [14] M. Banaie-Dezfouli, M. H. Nadimi-Shahraki, and Z. Beheshti, "R-GWO: representative-based grey wolf optimizer for solving engineering problems," *Applied Soft Computing*, vol. 106, 2021.
- [15] S. Khalilpourazari, H. H. Doulabi, A. Ö. Çiftçioglu, and G.-W. Weber, "Gradient-based grey wolf optimizer with Gaussian walk: application in modelling and prediction of the COVID-19 pandemic," *Expert Systems with Applications*, vol. 177, 2021.
- [16] E. El-shafeiy, "A clustering based Swarm Intelligence optimization technique for the Internet of Medical Things," *Expert Systems with Applications*, vol. 173, 2021.
- [17] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [18] GB/T9978-2008, *Fire resistance test methods of building components*, National Standard of the People's Republic of China, 2008.
- [19] Genlovy-Hoo, "utils_hoo[Source code]," 2021, https://github.com/Genlovy-Hoo/utils_hoo/tree/master/utils_hoo.
- [20] H. Faris, I. Aljarah, S. Mirjalili, P. Castillo, and J. Merelo, "EvoLoPy: an open-source nature-inspired optimization framework in Python," in *Proceedings of the 8th International Joint Conference on Computational Intelligence*, vol. 1, p. 171, 2016.
- [21] R. Qaddoura, H. Faris, I. Aljarah, and P. A. Castillo, "EvoCluster: an open-source nature-inspired optimization clustering framework in Python," in *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, pp. 20–36, Springer, Cham, 2020.
- [22] R. A. Khurma, I. Aljarah, A. Sharieh, and S. Mirjalili, "EvoLoPy-FS: An Open-Source Nature-Inspired Optimization Framework in Python for Feature Selection," in *Evolutionary Machine Learning Techniques*, pp. 131–173, Springer, 2020.