

## **Research** Article

# **COVID-19 Propagation Prediction Model Using Improved Grey** Wolf Optimization Algorithms in Combination with XGBoost and Bagging-Integrated Learning

### Yonghui Duan,<sup>1</sup> Yaqi Mao <sup>(b)</sup>,<sup>1</sup> Yibin Guo,<sup>2</sup> Xiang Wang <sup>(b)</sup>,<sup>2</sup> and Shen Gao<sup>1</sup>

<sup>1</sup>Department of Civil Engineering, Henan University of Technology, No. 100, Lianhua Street, Gaoxin District, Zhengzhou 450001, China

<sup>2</sup>Department of Civil Engineering, Zhengzhou University of Aeronautics, No. 15, Wenyuan West Road, Zhengdong New District, Zhengzhou 450015, China

Correspondence should be addressed to Yaqi Mao; 728115479@qq.com

Received 3 March 2022; Revised 24 April 2022; Accepted 13 May 2022; Published 3 June 2022

Academic Editor: Pijush Samui

Copyright © 2022 Yonghui Duan 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.

Toward solving the slow convergence and low prediction accuracy problems associated with XGBoost in COVID-19-based transmission prediction, a novel algorithm based on guided aggregation is presented to optimize the XGBoost prediction model. In this study, we collect the early COVID-19 propagation data using web crawling techniques and use the Lasso algorithm to select the important attributes to simplify the attribute set. Moreover, to improve the global exploration and local mining capability of the grey wolf optimization (GWO) algorithm, a backward learning strategy has been introduced, and a chaotic search operator has been designed to improve GWO. In the end, the hyperparameters of XGBoost are continuously optimized using COLGWO in an iterative process, and Bagging is employed as a method of integrating the prediction effect of the COLGWO-XGBoost model optimization. The experiments, firstly, compared the search means and standard deviations of four search algorithms for eight standard test functions, and then, they compared and analyzed the prediction effects of fourteen models based on the COVID-19 web search data collected in China. Results show that the improved grey wolf algorithm has excellent performance benefits and that the combined model with integrated learning has good prediction ability. It demonstrates that the use of network search data in the early spread of COVID-19 can complement the historical information, and the combined model can be further extended to be applied to other prevention and control early warning tasks of public emergencies.

#### 1. Introduction

According to the World Health Organization, the new Coronavirus pneumonia outbreak constitutes a global public health emergency as of January 31, 2020 [1]. China has suffered great economic and social harm following the outbreak of Corona Virus Disease 2019 (COVID-19), in late December 2019 in Wuhan, China [2]. Considering that COVID-19 is highly infectious [3], its origin, transmission routes, and pathogenesis have remained unclear at the beginning of the outbreak, and the government's prevention and control strategy have been gradually and dynamically adapted to the epidemic's development. Scientific forecasting can help prevent major outbreaks by establishing a

proactive and anticipatory safeguard system to prevent the continuous spread of epidemics [4]. The early identification of the epidemiological trend of COVID-19 will assist in controlling the spread and progression of the disease, thereby reducing its social burden. Therefore, it is important to examine a fast and efficient prediction model for major public emergencies to develop reasonable prevention and control measures.

In COVID-19 epidemic prediction, scholars around the world used public data of the epidemic to conduct a variety of data analyses and tested a variety of prediction methods and prediction models. In combining and analyzing the categories and mechanisms of these models, it has been found that existing prediction methods can be broadly classified into three categories: curve-fitting methods, kinetic methods, and machine learning methods. Zhao et al. [5] were the first to make this prediction. They used curve-fitting based on exponential growth trends to predict the number of cases at the beginning of the outbreak and determined that the early transmission capacity of the novel coronavirus was similar to or slightly greater than the early transmission capacity of SARS. The traditional curve-fitting prediction methods are very different from reality because they do not consider the characteristics of infectious diseases. A kinetic model describes the speed, spatial extent, transmission pathways, and kinetic mechanisms of infectious diseases [6]. It is worth noting that the majority of kinetic models for the current epidemic are based on SEIR models. The authors of Geng et al. [7] created an SEIR model to evaluate the efficacy of current prevention and control strategies using the characteristics of COVID-19. To predict COVID-19 with reference to the spread of SARS in the absence of data, Wu et al. [8] added crowd flow rate to the traditional SEIR model. Nevertheless, although kinetic methods are good predictors of early trends in epidemics, they are insufficient to estimate the spread of disease in open mobile settings or allow the assumed constants of disease transmission capacity and cure probability to match reality. Hence, an analysis of epidemic trends based on these methods will be inaccurate over the long term [9]. Since the emergence of COVID-19, with the continual increase in data samples and the advent of big data, the superiority of technologies, such as artificial intelligence, has been fully apparent. Researchers Peng et al. [10] employed four machine learning models for the prediction of clinical outcomes: artificial neural networks (ANN), naive Bayesian models (NBM), logistic regression (LR), and random forest (RF) models. The study by Liu et al. [11] constructed LR, k-nearest neighbor (KNN), decision tree (DT), multilayer perceptron (MLP), random forest (RF), support vector machines (SVM), and explainable boosting machine (EBM), a total of seven COVID-19 diagnostics, and evaluated the accuracy of each model for COVID-19. Ji et al. [12] developed a new particle swarm algorithm (ADVPSO) to optimize the parameters of ANN and applied it to predicting the spread trend of COVID-19. Using RF and ANN, Balerjee et al. [13] developed a predictive model for neocoronavirus pneumonia based on the results of complete blood counts in admitted patients. The model was able to identify 85% of community neo-coronavirus-positive patients without considering patient symptoms or medical history. Using daily data from the Iranian Ministry of Health between February 19 and March 30, 2020, Moftakhar et al. [14] built ARIMA and ANN models, respectively, to forecast the number of confirmed patients to be confirmed in Iran within the next month. A study by Almazroi et al. [15] demonstrated that popular machine learning algorithms, such as ANN and DT, have high variance and bias in prediction and used four integrated models of gradient boosting decision tree (GBDT), RF, extreme gradient boosting (XGBoost), and voting regressors (VR) to predict the number of new coronary pneumonia cases in Saudi Arabia each day. The XGBoost model was found to be the strongest predictor. It is the advantage of machine learning

methods that they can learn and train from historic data to construct an intelligent prediction model for the development of the epidemic. Not only does the prediction effect depend on the quality of the prediction model but also on the data used for the prediction. Online public opinion [16] is an important vehicle for the dissemination of emergency information in the context of big data. Following the outbreak of an emergency, internet users often learn about the event through search engines or portals. Ginsberg et al. [17] became the first to propose such a study. They utilized Google's extensive user search data to accurately predict the trend of the proportion of influenza-like cases in the United States one week in advance using the "Google Flu Trends" software developed by Google in 2008. A subsequent study was conducted by some scholars as a follow up to this. Signorini et al. [18] calculated a real-time tracking prediction model for the proportion of influenza-like illnesses across the U.S. and in specific regions using the percentage of U.S. Twitter volume containing influenza-related keywords published within the U.S. as a predictor variable. Li et al. [19] noted that the web searches of historical data can better predict influenza trends, while the web searches of current data can ensure the accuracy of predictions of new changes. A few scholars have also utilized internet search data in the analysis and study of the current epidemic. In some parts of the United States, Kurian et al. [20] found a strong correlation between searches on Google Trends and outbreaks of new corona infections.

Overall, different attempts have been made to address the prediction problem of COVID-19 in the existing literature, however, some limitations remain. Firstly, traditional methods of curve-fitting and kinetic analysis have the shortcomings of having low prediction accuracy and poor generalization properties. Secondly, most of the data used in the existing literature are traditional statistical data, which present problems associated with long acquisition periods and poor timeliness, affecting prediction models' timeliness. Thirdly, although machine learning techniques are widely employed in COVID-19 prediction, most scholarly studies apply them to predicting COVID-19 diagnosis, whereas there is little literature specifically addressing the prediction of the new corona epidemic trend.

This paper attempts to address the shortcomings listed above by focusing on the following three aspects of COVID-19 epidemic transmission prediction problem.

In the first step, the COVID-19 network opinion keyword database is developed using relevant references, the Baidu search index keyword time series data is crawled, and keywords that have high correlation are screened out as input features of the machine learning model using the LASSO algorithm.

Next, the final keywords are input into the XGBoost model. To increase the accuracy of the prediction, a chaotic search factor is designed, and a backward learning strategy is introduced to create an improved grey wolf evolutionary algorithm (COLGWO). Using the algorithm, the hyperparameters of the XGBoost model are determined, and then the COLGWO-XGBoost model as a prediction study of the change in the cumulative number of confirmed COVID-19 cases in China is performed.

The third point is that, to further improve the generalization performance of COLGWO-XGBoost, a combined Bagging-COLGWO-XGBoost early warning model is proposed based on Bagging's idea of integrated learning. Such a model includes multiple benchmark models and performance evaluation metrics to provide a more scientific and comprehensive assessment of the proposed model.

#### 2. Web Search Data

2.1. Data Collection. We used the cumulative number of confirmed diagnoses nationwide as the prediction label here, which was obtained using the DXY website's (https://www. dxy.cn) interface (excluding Hong Kong, Macau, and Taiwan). Data began with the first national notification on the official website of the National Health and Wellness Commission of the People's Republic of China. As of January 20, 2020, to May 7, 2020, the period ranged from the time the national prevention and control work was mostly in the normalization stage [21], with a total of 110 valid observations.

The internet generates more data with greater timeliness with the development of information technology and the penetration of networks. During an outbreak, people will turn to search engines, such as Baidu and Google, to seek information about the outbreak's cause and the corresponding preventive measures. Utilizing open-source web search data to monitor the new corona epidemic in China is a powerful addition to traditional surveillance tools and can serve as an early warning system, guide medical treatment, and improve prevention and control strategies during the incipient stages of the epidemic. China's most frequently used search engine Baidu has a feature called Baidu search index that counts the daily search volume figures for various keywords. It can effectively reflect a keyword's media and user attention at a specific point in time and reflect the network public opinion change in an emergency situation [22]. According to the Baidu index, daily statistics are published for the previous day, while the main variable used as an early warning model must be at least two days ahead of time to have practical application [23]. As such, in the present study, we select the number of searches for Baidu index keywords related to the emergency state during the epidemic outbreak two days prior to the outbreak, as an input feature. This information is also gathered by web crawler technology on the official Baidu index web interface (https://index.baidu.com). The time period is from January 18, 2020, to May 5, 2020.

2.2. Data Preprocessing. As a keyword, it is a term that summarizes what the user is seeking most concisely. The effectiveness of this prediction depends on selecting effective web search terms relevant to this COVID-19 outbreak. There are three primary types of initial keyword selection methods: direct word selection method, technical word selection method, and range word selection method [24]. The analysis of the advantages and disadvantages of these three types of initial keyword selection methods and the review of relevant literature have led this study to adopt range word selection in combination with the direct word selection method to select 42 keyword libraries. High-dimensional data often cause problems, such as high computational complexity and long running times for models, which is why this study employs the Lasso method to reduce the dimensionality of the initially established index system and eliminate the features of no significant significance. Basically, the Lasso method is to add a penalty function to the sum of squared residuals and compress the coefficients in estimating the parameters to produce feature selection: the larger the parameter  $\lambda$ , the fewer the features selected. Our paper utilized a 10-fold cross-validation method to determine the best value for  $\lambda$ . The seven keywords with zero coefficients were removed, and a total of 35 keywords were found as predictive factors across four categories of prevention, symptoms, treatment, and common COVID-19 terms, as shown in Table 1.

The web search keywords in the table were used as input features, and the national cumulative number of confirmed diagnoses was used as a predictor label. Also, all prediction data have been normalized to eliminate prediction errors caused by different data magnitudes. Normalization is determined by the following mathematical principle:

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}.$$
 (1)

Here,  $x^*$  is the normalized data value, x is the original data value before normalization, and  $x_{\min}$  and  $x_{\max}$  are the minimum and maximum values of the original data. This normalized value falls within the range of [0, 1], which is then randomly divided into a training set and a test set (where the training set represents 80% of the data samples and the test set represents 20%), and once the model has been trained, the prediction results are back-normalized to obtain the predicted values.

2.3. Correlation Analysis of Sample Data. Spearman's correlation coefficients were calculated between the final keyword data in Table 1 and the predicted variable (cumulative number of confirmed diagnoses). The results are shown in Table 2. It can be seen that the Baidu index of 33 of the 35 keywords selected in the sample data showed a statistically significant correlation with the cumulative number of confirmed cases of COVID-19 (P < 0.01). Among them, the correlations of "fever," "malaise," "respiratory infection," and "cold" were high in the classification of symptom words, with absolute values above 0.8. Although these four symptoms are not specific to COVID-19, they are easy to identify and are of great concern to the general population and are the best indicators of possible COVID-19 infection. It also indicates that the general population is aware of the basic knowledge of the symptoms of COVID-19. In addition, the absolute values of the correlation coefficients of "N95 mask," "medical surgical mask," and "84 disinfectant" also reached above 0.8. It indicates that with the further spread of the epidemic and the escalation of prevention and control measures, residents' demand for personal protection knowledge tends to be stronger.

Category	Web search keywords
Common words	New Coronavirus $(X_1)$ , New Coronary Pneumonia $(X_2)$ , Respiratory diseases $(X_3)$ , COVID-19 $(X_4)$
Prevention	N95 mask ( $X_5$ ), medical surgical mask ( $X_6$ ), nucleic acid detection ( $X_7$ ), epidemic prevention and control ( $X_8$ ), hand washing ( $X_9$ ), seven-step hand washing technique ( $X_{10}$ ), hand sanitizer ( $X_{11}$ ), hand sanitizer ( $X_{12}$ ), disinfection ( $X_{13}$ ), 84 disinfectant ( $X_{14}$ ), disinfectant brand ( $X_{15}$ ), chlorine dioxide ( $X_{16}$ ), thermometer ( $X_{17}$ ), forehead temperature gun ( $X_{18}$ ),
	infrared thermometer $(X_{19})$
Symptoms	Fever $(X_{20})$ , malaise $(X_{21})$ , respiratory infection $(X_{22})$ , cold $(X_{23})$ , headache $(X_{24})$ , diarrhoea $(X_{25})$ , virus mutation $(X_{26})$ ,
	asymptomatic infected person $(X_{27})$ , aerosol transmission $(X_{28})$ , contact transmission $(X_{29})$ , incubation period $(X_{30})$
Treatment	Fever clinic $(X_{31})$ , cold medicine $(X_{32})$ , fever-reducing medicine $(X_{33})$ , new crown vaccine $(X_{34})$ , ventilator $(X_{35})$

TABLE 1: Web search keyword phrase database.

TABLE 2: Correlation analysis of predictor and predicted variables in sample data.

Predicted variable: cumulative number of confirmed cases in China					
Predictor variable	Correlation coefficient	P value	Predictor variable	Correlation coefficient	P value
$X_1$	0.034	> 0.05	$X_2$	0.139	>0.05
$X_3$	-0.592	< 0.01	$X_4$	0.538	< 0.01
$X_5$	-0.929	< 0.01	$X_6$	-0.874	< 0.01
$X_7$	0.536	< 0.01	$X_8$	-0.366	< 0.01
$X_9$	-0.591	< 0.01	$X_{10}$	0.485	< 0.01
$X_{11}$	-0.564	< 0.01	$X_{12}$	-0.536	< 0.01
X <sub>13</sub>	-0.762	< 0.01	$X_{14}$	-0.815	< 0.01
$X_{15}$	-0.724	< 0.01	$X_{16}$	-0.574	< 0.01
$X_{17}$	-0.789	< 0.01	$X_{18}$	-0.353	< 0.01
X19	-0.651	< 0.01	$X_{20}$	-0.820	< 0.01
X <sub>21</sub>	-0.825	< 0.01	$X_{22}$	-0.969	< 0.01
X <sub>23</sub>	-0.956	< 0.01	$X_{24}$	-0.726	< 0.01
X <sub>25</sub>	-0.727	< 0.01	$X_{26}$	-0.583	< 0.01
X <sub>27</sub>	0.692	< 0.01	$X_{28}$	-0.216	< 0.01
X <sub>29</sub>	-0.474	< 0.01	$X_{30}$	-0.727	< 0.01
X <sub>31</sub>	-0.882	< 0.01	$X_{32}$	-0.917	< 0.01
X <sub>33</sub>	-0.915	< 0.01	$X_{34}$	0.715	< 0.01
X <sub>35</sub>	0.596	< 0.01			

#### 3. Algorithm Analysis

3.1. Grey Wolf Optimizer. Mirjalili et al. [25] developed a pack intelligence optimization algorithm in 2014 that emulates the predatory behavior of grey wolf packs by simulating behaviors characteristic of grey wolf packs. During the optimization phase of GWO,  $\alpha$ ,  $\beta$ , and  $\delta$  wolves, the highest social level of the population in each generation, led the bottom  $\omega$  wolves through encirclements and searching. GWO has a simple structure, few parameters to be adjusted, and is thus easily implemented, which makes it applicable to several fields. It has been widely used for optimization.

To begin, we can describe mathematically the process by which a wolf pack searches for and slowly surrounds its prey.

$$D = \left| C \cdot X_p(t) - X(t) \right|,\tag{2}$$

$$X(t+1) = X_p(t) - A \cdot D,$$
 (3)

$$a = 2 - \frac{2I}{M},\tag{4}$$

$$A = 2a \cdot r_1 - a, \tag{5}$$

$$C = 2 \cdot r_2. \tag{6}$$

Here, X(t) is the position of the prey after the  $t^{\text{th}}$  iteration,  $X_p(t)$  is the position of the grey wolf at the  $t^{\text{th}}$  iteration, D denotes the distance between the grey wolf and the prey, X(t+1) denotes the update of the position of the grey wolf, A and C are the coefficient vectors, a is the convergence factor whose value decreases linearly from 2 to 0 with the number of iterations, I is the number of previous iterations, and M is the maximum number of iterations.  $r_1$  and  $r_2$  are the random numbers between [0, 1].

Secondly, the position of the three optimal wolves  $\alpha$ ,  $\beta$ , and  $\delta$  is constantly updated to determine the prey. The following is a mathematical description of the hunting process of a wolf pack:

$$D_a = \left| C_1 \cdot X_\alpha(t) - X(t) \right|,\tag{7}$$

$$D_{\beta} = \left| C_2 \cdot X_{\beta}(t) - X(t) \right|, \tag{8}$$

$$D_{\delta} = \left| C_3 \cdot X_{\delta}(t) - X(t) \right|, \tag{9}$$

$$X_1(t+1) = X_{\alpha}(t) - A_1 \cdot D_{\alpha},$$
 (10)

$$X_{2}(t+1) = X_{\beta}(t) - A_{2} \cdot D_{\beta}, \qquad (11)$$

$$X_{3}(t+1) = X_{\delta}(t) - A_{3} \cdot D_{\delta}, \qquad (12)$$

$$X(t+1) = \frac{X_1(t+1) + X_2(t+1) + X_3(t+1)}{3}.$$
 (13)

Here,  $X_{\alpha}(t)$ ,  $X_{\beta}(t)$ , and  $X_{\delta}(t)$  are the positions of  $\alpha$ ,  $\beta$ , and  $\delta$  wolves when the population is iterated to the  $t^{\text{th}}$ generation. X(t) is the position of individual grey wolves in the  $t^{\text{th}}$  generation.  $A_1$  and  $C_1$ ,  $A_2$  and  $C_2$ ,  $A_3$  and  $C_3$  are the coefficient vectors of  $\alpha$ ,  $\beta$ , and  $\delta$  wolves, respectively.  $X_1(t+1)$ ,  $X_2(t+1)$ , and  $X_3(t+1)$  indicate the positions of  $\alpha$ ,  $\beta$ , and  $\delta$  wolves after (t+1) iterations, respectively. X(t+1)is the position of the next generation of grey wolves.

3.2. Improving Grey Wolf Optimizer. Though the standard GWO provides better performance than most intelligent optimization methods, it is not suitable for dealing with complex functions. Hence, there is a need to improve the balance between global search and local convergence to improve the performance of the grey wolf algorithm [26]. In the standard GWO, the prey direction and distance between preys are continuously adjusted by employing the equations from (7) through (13) until the prey is caught after many iterations, and by analyzing this process, it was found that grey wolves move to the same region as the number of iterations increases, which makes the whole pack converge seriously, making the wolf pack easily fall into local optimum. The chaos opposition learning grey wolf optimizer (COLGWO) is proposed in this study as a method to enhance the defects of standard GWO. The details are provided below.

3.2.1. Opposition-Based Learning. If the grey wolves of the initialized population are near the optimal solution, the convergence is quicker, and if the grey wolves of the initialized population deviate from the optimal solution, the algorithm is slower or fails to converge. Opposition-based learning (OBL) was implemented by Tizhoosh et al. [27] to help expand the individual search space, enhance the global exploration capability, and overcome the algorithm falling into local optimum. Below is its mathematical definition.

Initially, setting the inverse point serves as the population initialization for the grey wolf optimizer. For a point  $x = (x_1, x_2, ..., x_D)$  in the *D*-dimensional space, its reverse point  $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_D)$  is calculated as follows:

$$\tilde{x}_i = a_i + b_i - x_i. \tag{14}$$

Here,  $x_i \in [a_i, b_i]$ , and  $a_i$  and  $b_i$  are the value boundaries of the point x in the *i*<sup>th</sup> dimension.

Secondly, the grey wolf algorithm is iteratively optimized using backward learning and jump probabilities to best achieve its global optimization objectives. Let  $J_r$  be the jump probability, and if rand  $[0, 1] \leq J_r$ , then the reverse learning population is generated. Setting as one of the grey wolf individuals in the  $D^{\text{th}}$  dimension,  $\vec{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D}), x_{i,j} \in [a_j, b_j]$ , the corresponding inverse solution  $\tilde{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$  for that individual is calculated as follows:

$$\widetilde{x}_{i,j} = k \cdot \left( da_j + db_j \right) - x_{i,j}.$$
(15)

Here, k takes the value of any random number between [0, 1], and  $[da_j, db_j]$  is the dynamic boundary of the particle in the  $J^{\text{th}}$  dimension, which is calculated as follows:

$$da_j = \min(x_{i,j}), db_j = \max(x_{i,j}).$$

$$(16)$$

Within the dynamic boundary, a point is generated at random if the reverse solution jumps out of it.

$$\widetilde{x}_{i,j} = \operatorname{rand}(da_j, db_j), \text{ if } \widetilde{x}_{i,j} < da_j \operatorname{or} \widetilde{x}_{i,j} > db_j.$$
(17)

3.2.2. Chaotic Search Factor. By utilizing the randomness, ergodicity, and regularity of the logistic chaotic sequence, a chaotic search input is added to the grey wolf optimizer, and a local search is performed in the vicinity of the optimal  $\alpha$ -wolf in each iteration. In addition, a shrinkage strategy is introduced to achieve a large search range at the beginning of the iteration and a smaller search range at the end, which enhances the local mining capability of the standard GWO algorithm, allowing the algorithm to perform both global and local searches equally well while improving its accuracy. The following is a mathematical description.

The first step is to generate the chaotic variables  $c_i$ .

$$c_{i+1} = 4 \cdot c_i \cdot (1 - c_i). \tag{18}$$

Here, the range of the values of *i* is [1, K], the range of the values of  $c_i$  is (0, 1), and  $c_i \neq 0.25$ , 0.5, and 0.75. *K* is the length of the chaotic sequence.

A second step is the mapping of the chaotic variable  $c_i$  generated by equation (18) to a chaotic vector  $C_i$  in the definition domain [LB, UB].

$$C_i = LB + c_i \cdot (UB - LB). \tag{19}$$

Here, i takes values within the range [1, K]. LB and UB represent the lower and upper limits, respectively.

In addition, the chaotic vector  $C_i$  and optimal  $\alpha$ -wolf are linearly combined to generate the chaotic wolf  $V_i$ .

$$V_i = (1 - \lambda) \cdot \alpha + \lambda \cdot C_i, \tag{20}$$

$$\lambda = \frac{M - I + 1}{M}.$$
 (21)

where *i* belongs to the range [1, K].  $\lambda$  is the contraction factor, *M* is the maximum number of iterations in the algorithm, and *I* is the current number of iterations.

To be able to better understand the chaos search factor, the following is a brief example of a one-dimensional function sphere, from which higher dimensions can be extended. Suppose the function is defined in the domain [-100, 100], the current optimal  $\alpha$  wolf is 10, the current iteration number is 2, and the maximum iteration number is 100. If the random number  $c_1 = 0.1$ , then according to equation (18), we have  $c_2 = 0.36$  and  $c_3 = 0.9216$ . In turn, according to equation (19), we have  $C_1 = -80$ ,  $C_2 = -28$ , and  $C_3 = 84.32$ . From equation (21), we can get  $\lambda = 0.99$  at this time. Then, the chaos vector  $C_i$  is linearly combined with the head wolf  $\alpha$  at this time by equation (20) to get the chaos wolf  $V_1 = -79.1$ ,  $V_2 = -27.62$ , and  $V_3 = 83.5768$ . However, if the generated  $V_i$  fitness value is better than that of the  $\alpha$  wolf, then the new  $\alpha$  wolf is replaced as  $V_i$ . It should be noted that if the current iteration number is 1, then  $\lambda = 1$  and  $V_i = C_i$ , i.e.,  $V_1 = 80$ ,  $V_2 = -28$ , and  $V_3 = 84.32$ . The chaotic wolves at this point are randomly generated from the chaotic vectors, independent of the head wolf. It can be observed from the above example that  $\lambda$  gradually becomes smaller with the increase of evolutionary generations, and the generated chaotic wolf is smaller at this time, which indicates that the scope of local search becomes smaller with the continuous iteration of the grey wolf optimizer.

3.3. Bagging-Integrated Learning Strategy. For integrated learning methods to improve model generalization and prediction accuracy, a combination of prediction results from multiple homogeneous or heterogeneous models is often used [28]. Bootstrap aggregating (Bagging) aims to combine the predictions of multiple homogeneous base learners, and the combination of this strategy can achieve a significant improvement in prediction model generalization capabilities and avoid the phenomenon of overfitting.

In Bagging-integrated learning strategy, data are randomly selected from a data set using a put-back sampling methodology (Bootstrap) and then N subtraining sets of equal size are produced. Following that, each subtraining set is separately trained to obtain N base learners, and finally, the final prediction results of each base learner are arithmetically averaged together. This process is illustrated in Figure 1.

3.4. Bagging-COLGWO-XGBoost Model. XGBsoost is a boosting class model that has been developed by Chen et al. [29] in 2016, which is a combination of a linear scale solver and a categorical regression tree. The basic idea behind this model is to combine multiple tree models that have low prediction accuracy through various methodologies to construct a combined model with a higher prediction accuracy. After the combinatorial model is constructed, it is iterated through gradient boosting, with each iteration producing a new tree to fit the residuals generated by the previous tree until the optimal model is obtained. The XGBoost method is based upon a secondorder Taylor expansion of the loss function. Furthermore, to mitigate the decline of the objective function and the complexity of the model, a regular term is added in addition to the objective function to arrive at the optimal solution, thereby avoiding the overfitting problem. A number of studies in recent years have demonstrated



FIGURE 1: Bagging integration model.

good performance of the XGBoost model for predictions in biology, medicine, and economics. The mathematical principles of the model are as follows:

The integration model of the definition tree is as follows:

$$\hat{y}_{i} = \sum_{m=1}^{M} f_{m}(x_{i}), f_{m} \in F.$$
(22)

In this equation,  $\hat{y}_i$  is the prediction value, M is the number of trees, F is the range of tree selections, and  $x_i$  is the  $i^{\text{th}}$  input feature.

The loss function for the XGBoost model is shown below.

$$L = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{m=1}^{M} \theta(f_m).$$
(23)

Here, the first part of the function is the error between the predicted and the actual training values of the XGBoost model, while the second is used to represent the complexity of the tree, which is important when controlling the regularization of the complexity of the model.

$$\theta(f) = \gamma T + \frac{1}{2}\tau\omega^2.$$
(24)

Here,  $\gamma$  and  $\tau$  represent penalty factors.

It is minimized by adding the incremental function  $f_t(x_i)$  to equation (23) to minimize the value of the loss function. Thus, the objective function for the  $t^{\text{th}}$  time becomes as follows:

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{m=1}^{M} \theta(f_m) = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{t-1} + f_t(x_i)) + \theta(f_t).$$
(25)

A second-order Taylor expansion of equation (25) is used to approximate the objective function at this point. Define the set of samples in each subleaf of the  $j^{\text{th}}$  tree as  $I_j = \{i | q(x_i = j)\}$ . At this point, we can approximate  $L^{(t)}$  as follows:

TABLE 3: Brief information on the 8 standard test functions.

Function name	Function form	Search interval	Minimum value
$f_1(x)$ Discus	$f_1(x) = 10^6 x_1^2 + \sum_{i=2}^D x_i^2$	[-100, 100]	0
$f_2(x)$ Bent Cigar	$f_2(x) = x_1^2 + 10^6 \sum_{i=2}^D x_i^2$	[-100, 100]	0
$f_3(x)$ Zakharov	$f_3(x) = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5x_i)^2 + (\sum_{i=1}^{D} 0.5x_i)^4$	[-100, 100]	0
$f_4(x)$ Sphere	$f_4(x) = \sum_{i=1}^D x_i^2$	[-100, 100]	0
$f_5(x)$ Rastrigin	$f_5(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	[-100, 100]	0
$f_6(x)$ Griewank	$f_6(x) = \sum_{i=1}^{D} x_i^2 / 4000 - \prod_{i=1}^{D} \cos(x_i / \sqrt{i}) + 1$	[-100, 100]	0
$f_7(x)$ Rosenbrock	$f_7(x) = \sum_{i=1}^{D-1} (100 (x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	[-100, 100]	0
$f_8(x)$ Ackley	$f_8(x) = -20 \exp\left(-0.2\sqrt{1/D\sum_{i=1}^D x_i^2}\right) - \exp\left(1/D\sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	[-100, 100]	0

TABLE 4: Optimization search results of four algorithms for standard test functions.

		DE	PSO	GWO	COLGWO
$f_1(x)$	Average value	0.00716952881962376	4.58E - 07	1.74E - 27	0
	Standard deviation	0.00892434320906037	9.67E - 07	2.64E - 27	0
£ ()	Average value	2459.11866144729 4	483153344	4.98E - 22	0
$f_2(x)$	Standard deviation	2438.72294203683	15.26685017	9.28E - 22	0
f ()	Average value	0.254499657312407	3.41E - 05	9.27E - 24	0
$f_3(x)$	Standard deviation	0.681425676307518	7.64E - 05	1.25E - 23	0
$f_4(x)$	Average value	0.00440696961956876	1.15E - 07	6.60E - 28	0
	Standard deviation	0.00489054323509337	1.96E - 07	8.25E - 28	0
$f(\alpha)$	Average value	211.342901198991	1129.880979	3.216822796	0
$\int_{5}(x)$	Standard deviation	21.4117756082338	421.9713817	4.907623241	0
$f_6(x)$	Average value	0.00407161054793563	0.033167593	0.003456494	0
	Standard deviation	0.00636993045210643	0.033820478	0.007518388	0
$f_7(x)$	Average value	34.9908955808477	0.034919137	2.485701732	0
	Standard deviation	22.3283305576275	0.035951133	12.08406381	0
$f_8(x)$	Average value	20.9805740011391	20.47832934	9.932619146	4.44E - 16
	Standard deviation	0.106398593367519	0.618180325	10.41113601	0

$$L^{(t)} \cong \sum_{i=1}^{n} \left[ g_i f_t(x_i) + \left(\frac{1}{2}\right) h_i f_t^2(x_i) \right] + \theta(f_t)$$
  
$$\cong \sum_{i=1}^{n} \left[ g_i f_t(x_i) + \left(\frac{1}{2}\right) h_i f_t^2(x_i) \right] + \gamma T + \frac{1}{2} \tau \omega^2 \qquad (26)$$
  
$$\cong \sum_{j=1}^{T} \left[ \left( \sum_{i \in I_j} g_i \right) \omega_j + \left(\frac{1}{2}\right) \left( \sum_{i \in I_j} h_i + \tau \right) \omega_j^2 \right] + \gamma T.$$

Here,  $g_i = \partial_{\widehat{y_i}^{t-1}} l(y_i, \widehat{y_i}^{t-1})$  is the first-order derivative of the loss function.  $h_i = \partial_{\widehat{y_i}^{t-1}}^{2} l(y_i, \widehat{y_i}^{t-1})$  is the second-order derivative. The subsequent equation is calculated by defining  $G_j = \sum_{i \in I_j} g_i$ ,  $H_j = \sum_{i \in I_j} h_i$ .

$$L^{(t)} \cong \sum_{j=1}^{T} \left[ G_j \omega_j + \left(\frac{1}{2}\right) \left(H_j + \tau\right) \omega_j^2 \right] + \gamma T.$$
 (27)

The following equation is obtained by taking the partial derivatives of  $\omega$ .

$$\omega_j = -\frac{G_j}{H_j + \tau}.$$
 (28)

The following equation can be obtained by substituting the weights into the objective function.

$$L^{(t)} \cong -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \tau} + \gamma T.$$
 (29)

Training the XGBoost model is heavily dependent on the choice of parameters as different parameter selections have a significant effect on the prediction results. Within the XGBoost algorithm, there are 23 hyperparameters, which can be classified into three types: general parameters for the control of macrofunctions, booster parameters to control booster details, and learning target parameters to control the training target parameters. COLGWO-XGBoost takes the three hyperparameters of *learning\_rate*, *n\_estimators*, and *max\_depth*, which have the greatest impact on the



FIGURE 2: Algorithm adaptation evolution curve.

performance of XGBoost, and uses them in the calculation of the position vector of the head wolf  $\alpha$  in COLGWO. The COLGWO algorithm is iterated until the position vector corresponding to the global optimal position is returned to the XGBoost model. Furthermore, to enhance the generalization characteristics of the prediction model, the Bagging integration strategy is employed to directly integrate the COLGWO-XGBoost model into the integrated learning process to construct the final early warning model in this paper. The combined use of this strategy can reduce the variance of the COLGWO-XGBoost model and prevent overfitting. The following are specific steps for implementing the model.

*Step 1.* Initialize the parameters by setting the maximum number of iterations *M*, the number of populations *N*, the upper bound UB, and the lower bound LB of the parameters to be searched for COLGWO.

Step 2. Randomly initialize the populations and generate equal numbers of reverse populations based on equation (14), use equation (29), select the individual positions that correspond to the optimal fitness values among the two populations to be put into the final initial population, and then select the individual positions that correspond to the optimal three fitness values as  $\alpha$ ,  $\beta$ , and  $\delta$  wolves.

Step 3. The chaotic search factor is called to generate the chaotic wolf  $V_i$  according to equations 18 through (21), and if the fitness value of  $V_i$  is greater than  $\alpha$  wolf, then  $V_i$  is replaced with  $\alpha$  wolf, and the local search ends. Otherwise, if the chaotic sequence length is K, then the local search ends as well.

Step 4. The convergence coefficient *a* and the coefficient vectors *A* and *C* were updated for  $\alpha$ ,  $\beta$ , and  $\delta$  wolves, respectively, according to equations (4) to (6).

Step 5. It is necessary to calculate the distances between  $\alpha$ ,  $\beta$ , and  $\delta$  wolves and their prey, and update the positions of these wolves according to equations (7) to (12), respectively.

*Step 6.* Update wolf locations based on equation (13) to produce a new generation of grey wolf populations.

*Step 7.* Follow Steps 3 to 6. If the maximum number of COLGWO iterations *M* is reached, then stop the iteration. Determine the position of  $\alpha$  wolf at this point as a parameter of the sought-after XGBoost model, and construct the COLGWO-XGBoost prediction model.

Step 8. Using bootstrap sampling, input data are divided into N subtraining sets of equal sample size, and then the COLGWO-XGBoost model is used to fit each subtraining set separately to generate N base learners.

*Step 9.* Using *N* base learners, the prediction data are predicted separately, and their predicted results are arithmetically averaged, resulting in the final prediction result of the Bagging-COLGWO-XGBoost model.

#### 4. Experimental Results and Discussion

#### 4.1. Improved Algorithm Performance Testing

4.1.1. Standard Test Functions. We have selected eight typical standard test functions from the global optimization benchmark functionality library to test the optimization-seeking capability of the COLGWO algorithm, whose names, functions, search intervals, and theoretical minimum values are shown in Table 3.  $f_1(x) \sim f_4(x)$  are continuous single-peaked functions, which are typically used to test the local mining abilities of algorithms.  $f_5(x) \sim f_8(x)$  are complex nonlinear multipeaked functions with a large number of local extremes, which are generally used to test the global pioneering abilities of algorithms, and their results provide guidance to algorithms.

TABLE 6: Single model prediction accuracy comparison.

Algorithm name	RMSE	MAE	MAPE
XGBoost	3965.725	1469.406	9.785
GBDT	3095.005	1356.619	10.753
SVR	8591.444	7199.914	223.206
MLP	8551.856	6216.566	86.296
LR	4683.808	3406.161	116.283
LSTM	4854.758	3522.867	12.967
RNN	6506.414	4936.082	32.877
CNN	5862.456	4442.039	27.082

Bold font indicates the best prediction accuracy among all models.

4.1.2. Experimental Analysis. A comparison of the optimization performance of DE (differential evolution), PSO (particle swarm optimization), GWO, and COLGWO in eight typical benchmark test functions was conducted in the same runtime environment using Python version 3.7 as the simulation program. The primary parameters of the four algorithms were the same, namely 30 populations, a maximum number of iterations of 500, and a dimension of 30. The algorithms were run independently 30 times, and the mean and standard deviation of the search results were calculated. Table 4 shows the experimental results. On the basis of Table 4, it can be observed that the optimization results of GWO outperform those of the two classical evolutionary algorithms, DE and PSO, for continuous single-peak functions and complex nonlinear multipeak functions. Even though the improved COLGWO algorithm does not achieve the theoretically optimal solution for the  $f_8(x)$  function, it improves significantly compared to before the improvement, and it has the best solution accuracy among the four algorithms. It indicates that the modification of the standard GWO algorithm in this study is very effective, can well-balance the global search ability and local search ability, and improve the stability of the algorithm. For the eight standard test functions, Table 4 only displays the optimal values determined by the four algorithms. A comparison graph of the convergence of the four algorithms for the test functions is shown in Figure 2 for the purpose of visualizing the convergence of the

TABLE 5: Algorithm-related parameters.

Model	Parameter setting
Bagging-COLGWO-XGBoost	$n_{\text{estimators}} = 21$
Bagging-COLGWO-GBDT	$n_{\text{estimators}} = 21$
COLGWO-XGBoost	<i>n</i> _estimators = 18; learning_rate = 0.8561; max_depth = 7
COLGWO- GBDT	<i>n</i> _estimators = 22; learning_rate = 0.3208; max_depth = 4
GWO-XGBoost	$n_{\text{estimators}} = 46$ ; learning_rate = 0.8563; max_depth = 6
GWO-GBDT	<i>n</i> _estimators = 38; learning_rate = 0.5996; max_depth = 4
XGBoost	Default parameter
GBDT	Default parameter
LSTM	Optimizer = Adam; loss = mse; epochs = 100
RNN	Optimizer = Adam; loss = mse; epochs = 100
CNN	Optimizer = Adam; loss = mse; epochs = 100
SVR	Default parameter
MLP	Default parameter
LR	Default parameter
COLGWO	Iteration number: 100; Population size: 30
GWO	Iteration number: 100; Population size: 30

Mathematical Problems in Engineering



FIGURE 3: Single model prediction effect.

algorithms. Based on Figure 2, the speed of convergence of COLGWO algorithm has been significantly accelerated. Because of this, the improved algorithm presented in this paper takes into account the convergence speed, global exploration ability, and local fine search ability of the algorithm at the same time, which is a very powerful improvement.

#### 4.2. Comparison of Prediction Effects

4.2.1. Evaluation Functions. As evaluation functions for the forecasting models, we select root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). The model's smaller RMSE, MAE,

TABLE 7: Comparison of prediction accuracy of combined algorithms.

	г
Algorithm name RMSE MAE MAP	E
Bagging-COLGWO-XGBoost 904.515 539.551 2.759	9
Bagging-COLGWO-GBDT 2273.204 819.237 2.851	L
COLGWO-XGBoost 1211.977 579.778 3.235	5
COLGWO-GBDT 2419.836 1086.354 5.179	)
GWO-XGBoost 1230.206 620.652 4.823	3
GWO-GBDT 3084.185 1260.101 10.62	7

Bold font indicates the best prediction accuracy among all models.

and MAPE values indicate better prediction performance, and the actual formula is as follows:

RMSE = 
$$\sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$
, MAE =  $\frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$ , MAPE =  $\frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$ . (30)

Here, *n* is the number of samples, *i* is the sample number,  $y_i$  stands for the true value, and  $\hat{y}_i$  stands for the predicted value of the model.

4.2.2. Experimental Analysis. The present study employs multiple algorithms as comparison models to evaluate the predictive ability of the Bagging-COLGWO-XGBoost model

in the COVID-19 cumulative number of confirmations data. For single-model prediction, in addition to four traditional machine learning models of SVR, LR, MLP, and GDBT, convolutional neural networks (CNN) [30], recurrent neural network (RNN) [31], and long short-term memory (LSTM) [32], which are three classical deep learning models, were also selected to compare and analyze the prediction effect of XGBoost. In the combined algorithm prediction, the GWO-



FIGURE 4: Combined algorithm prediction effect.

optimized XGBoost model (GWO-XGBoost), GWO-optimized GBDT model (GWO-GBDT), COLGWO-optimized GBDT (COLGWO-GBDT), COLGWO-optimized XGBoost (COLGWO-XGBoost), and Bagging-COLGWO-GBDT are selected to compare and analyze the prediction effects of the Bagging-COLGWO-XGBoost model proposed in this study. Table 5 contains the parameters of each algorithm model.

The main model and the comparison model proposed in this paper are fitted to each prediction model in the training dataset, and the predictions are performed on the test dataset. The prediction accuracy of the single model in the three indexes is given in Table 6, and Figure 3 shows the prediction effect of the single model. It can be observed that the prediction performance of XGBoost and GBDT in single model is significantly better than SVR, MLP, LR, CNN, RNN, and LSTM. XGBoost ranked first in MAPE, and GBDT ranked first in RMSE and MAE. Both models outperformed several other comparative models in three metrics, RMSE, MAE, and MAPE, and this result indicates that the selection of a suitable regression model plays a key role in improving the prediction performance under the same data premise.

To further observe the difference in prediction performance between GBDT and XGBoost, in the combined algorithm, the GWO-seeking algorithm is first proposed to optimize the hyperparameters of these two models separately, and then the COLGWO proposed in this paper is used to compare the optimization results. Finally, the

Bagging strategy is used to integrate the learning of these two models. Table 7 and Figure 4 show the experimental results of the combined algorithm. After using the GWO optimization algorithm, the prediction accuracy of GBDT is slightly improved, however, it significantly lags behind that of XGBoost, Using COLGWO for the parameter optimization of both models achieves better optimization results than GWO. Using the Bagging integration strategy can further reduce the generalization error of the combined algorithm, and this strategy has the best prediction effect when combined with XGBoost using COLGWO algorithm. It shows that the improvement of XGBoost on GBDT is successful. Since the model adds a regular term in the loss function for controlling the complexity of the model, it reduces the variance of the model in terms of the trade-off variance bias and makes the trained model simpler, thus effectively preventing overfitting. In addition, although the prediction performance of XGBoost is excellent, the model is very sensitive to parameters, and the use of an efficient parameter selection method is crucial to the prediction effect, and the experimental results confirm the superiority of the improved algorithm (COLGWO) proposed in this study in terms of hyperparameter selection.

In summary, the overall prediction accuracy of the model optimized by the combined algorithm is better than that of the single model prediction. The Bagging-COLGWO-XGBoost model with the integrated strategy ranked first in all three evaluation metrics of all algorithms, improving 77.19%, 63.28%, and 71.8%, respectively, compared to XGBoost, and 60.21%, 34.14%, and 3.23%, respectively, compared to Bagging-COLGWO-GBDT. It indicates that the COLGWO algorithm proposed in this paper is very effective in the application of hyperparameter selection for the XGBoost model in the early warning problem of COVID-19. Combining with the Bagging integration strategy will further enhance the prediction effect.

#### 5. Conclusions

This work analyses the initial web search data associated with COVID-19 at the time of its outbreak and then uses the LASSO method to select features for input into the model of XGBoost for prediction. A new grey wolf algorithm is proposed to optimize the hyperparameters of XGBoost to improve prediction accuracy. We built a Bagging-COLGWO-XGBoost early warning model, which utilizes the Bagging integration strategy, to enhance the generalizability of the combined algorithm. After analyzing the comparison of these early warning models, the subsequent conclusions were drawn.

Firstly, the data source affects the accuracy and timeliness of the prediction model. Web search data has many advantages over traditional statistics, and it is convenient, timely, and sensitive to user needs, which can effectively supplement traditional statistics.

Secondly, to address the shortcomings of the standard GWO algorithm, such as low solution accuracy, slow convergence speed, and it being easy to fall into the local optimum and impossible to jump out, it was proposed in this paper that COLGWO initially employed a mechanism based on backward learning to initialize the population. Adding jump probabilities makes the algorithm jump out of the local optimal solution. Additionally, to enhance the mining ability of the standard GWO algorithm, a chaotic search operator was developed, which uses chaotic sequences to carry out local searches around the current optimal solution. Experimental results demonstrate that the improved grey wolf algorithm satisfies the purpose of balancing the global and local search for optimal solutions of the algorithm and that the optimization results in all eight standard test functions are vastly superior to those of the grey wolf algorithm. Also, the results demonstrate that the optimization of hyperparameters of the XGBoost model using it can significantly improve prediction accuracy.

Thirdly, in terms of prediction performance, the use of Bagging-integrated learning can result in a reduction in the variance of the predictive model and enhance generalization capability. The Bagging-COLGWO-XGBoost model presented in this paper performs significantly better than the thirteen other models in the prediction problem of the cumulative number of confirmed COVID-19 cases in China, indicating that the model is a good and reliable one.

Overall, the combined predictive model Bagging-COLGWO-XGBoost developed in this study combined with web search data can predict the cumulative number of confirmed cases of COVID-19 in China under this emergency situation. In addition, this model may be continuously updated and optimized according to the latest data to provide real-time prediction, which will serve as a reliable source of data for the Chinese government to formulate reasonable and effective health policy. Furthermore, the model constructed in this study can also be used for the prediction of similar major public emergencies and can be extended to other public emergencies related to early warning.

#### **Data Availability**

The data used to support the findings of this study are available from the corresponding author upon request.

#### **Conflicts of Interest**

The authors declare no conflicts of interest.

#### Acknowledgments

This research was funded by the National Natural Science Foundation of China (Grant no. 81973791).

#### References

- L. Zhang, "Fitness of the generalized growth to the COVID-19 data," *Journal of University of Electronic Science and Technology of China*, vol. 49, no. 3, pp. 345–348, 2020.
- [2] D. G. Francesco, P. Damiano, M. Claudia et al., "Coronavirus diseases (COVID-19) current status and future perspectives: a narrative review," *International Journal of Environmental Research and Public Health*, vol. 17, no. 8, Article ID 2690, 2020.
- [3] A. R. Hussin and N. B. Siddappa, "The epidemiology and pathogenesis of coronavirus disease (COVID-19) outbreak," *Journal of Autoimmunity*, vol. 109, Article ID 102433, 2020.
- [4] H. Xu, K. J. Peng, and N. Wei, "Forecasting to empower decision making: a solution from traditional models to big data's scheme-the insights of judgement of COVID-19's epidemic trend," *Journal of Public Management*, vol. 18, no. 4, pp. 116–173, 2021.
- [5] S. Zhao, Q. Lin, J. Ran et al., "Preliminary estimation of the basic reproduction number of novel coronavirus (2019-ncov) in China, from 2019 to 2020: a data-driven analysis in the early phase of the outbreak," *International Journal of Infectious Diseases*, vol. 92, pp. 214–217, 2020.
- [6] L. H. Huang, Y. Y. Wei, S. P. Shen, W. C. Zhu, and F. Chen, "Evaluation of predictive models for novel coronavirus pneumonia," *Chinese Journal of Health Statistics*, vol. 37, no. 3, pp. 322–326, 2020.
- [7] H. Geng, A. D. Xu, X. Y. Wang et al., "Analysis of the role of current prevention and control measures in the epidemic of Corona Virus Disease 2019 based on SEIR model," *Journal of Jinan University (Natural Science & Medicine Edition)*, vol. 41, no. 2, pp. 175–180, 2020.
- [8] J. T. Wu, K. Leung, and G. M. Leung, "Nowcasting and forecasting the potential domestic and international spread of the 2019-ncov outbreak originating in Wuhan, China: a modelling study," *The Lancet*, vol. 395, no. 10225, pp. 689–697, 2020.
- [9] W. J. Mei, Z. Liu, J. Y. Zhu, and L. Du, "Extreme IR model for COVID-19 real-time forecasting," *Journal of University of*

*Electronic Science and Technology of China*, vol. 49, no. 3, pp. 362–368, 2020.

- [10] C. Peng, G. Y. Qi, C. X. Zhang, Y. F. Guo, and Z. C. Jin, "Prediction of in-hospital clinical outcomes of coronavirus disease 2019 patients based on machine learning algorithms," *Academic Journal of Second Military Medical University*, vol. 42, no. 10, pp. 1115–1123, 2021.
- [11] Q. Liu, Y. Zeng, H. L. Wang, B. Liu, W. L. Zhao, and Y. Wang, "Diagnosis of machine learning methods for COVID-19," *Chinese Journal of Medical Imaging*, vol. 29, no. 4, pp. 293– 299, 2021.
- [12] W. D. Ji, H. T. Xu, and P. Lin, "Adaptive mutation particle swarm optimization and its application in predicting the COVID-19 epidemic transmission," *Journal of Chinese Computer Systems*, vol. 42, no. 3, pp. 472–477, 2021.
- [13] B. Abhirup, R. Surajit, V. Bart et al., "Use of machine learning and artificial intelligence to predict sars-cov-2 infection from full blood counts in a population," *International Immunopharmacology*, vol. 86, Article ID 106705, 2020.
- [14] L. Moftakhar, M. Seif, and M. S. Safe, "Exponentially increasing trend of infected patients with COVID-19 in Iran: a comparison of neural network and arima forecasting models," *Iranian Journal of Public Health*, vol. 49, no. 1, pp. 92–100, 2020.
- [15] A. A. Almazroi and R. S. A. Usmani, "COVID-19 cases prediction in Saudi Arabia using tree-based ensemble models," *Intelligent Automation & Soft Computing*, vol. 32, no. 1, pp. 389–400, 2022.
- [16] J. Q. Wang, R. Li, and H. Y. Wu, "The evolution of online public opinion based on spatial autocorrelation," *Data Analysis and Knowledge Discovery*, vol. 2, no. 2, pp. 64–73, 2018.
- [17] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant, "Detecting influenza epidemics using search engine query data," *Nature*, vol. 457, no. 7232, pp. 1012–1014, 2009.
- [18] A. Signorini, A. M. Segre, and P. M. Polgreen, "The use of twitter to track levels of disease activity and public concern in the U.S. During the influenza A H1N1 pandemic," *PLoS One*, vol. 6, no. 5, Article ID e19467, 2011.
- [19] X. Li, F. Liu, J. Dong, and B. Lu, "Detecting China influenza using search engine data," *Systems Engineering*, vol. 33, pp. 3028-3034, 2013.
- [20] J. K. Shyam, A. A. Mohammed, H. T. Henry et al., "Correlations between COVID-19 Cases and Google trends data in the United States: a state-by-state analysis," *Mayo Clinic Proceedings*, vol. 95, pp. 2370–2381, 2020.
- [21] "Fitness of the generalized growth to the COVID-19 data," Gazette of the State Council of the People's Republic of China, vol. 14, pp. 15–17, 2020.
- [22] N. Jing, Y. Hu, and X. S. Han, "Trend of COVID-19 network attention based on ARIMA and LSTM," *China Safety Science Journal*, vol. 30, no. 12, pp. 37–42, 2020.
- [23] C. C. Gao, X. C. Chen, R. Zhang, Q. Y. Song, D. Yi, and Y. Z. Wu, "Application of three new intelligent algorithms in epidemic early warning model—COVID-19 epidemic warning based on Baidu search index," *Computer Engineering and Applications*, vol. 57, no. 8, pp. 256–263, 2021.
- [24] A. Ali and F. Miad, "A review of influenza detection and prediction through social networking sites," *Theoretical Bi*ology and Medical Modelling, vol. 15, no. 1, pp. 1–27, 2018.
- [25] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," Advances in Engineering Software, vol. 69, pp. 46–61, 2014.

- [26] N. Singh and S. B. Singh, "A novel hybrid gwo-sca approach for optimization problems," *Engineering Science and Technology, an International Journal*, vol. 20, no. 6, pp. 1586–1601, 2017.
- [27] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in Proceedings of the International Conference On Computational Intelligence For Modelling, Control And Automation And International Conference On Intelligent Agents, Web Technologies And Internet Commerce (Cimca-iawtic'06), pp. 695–701, Vienna, Austria, November 2005.
- [28] J. W. Xu and Y. Yang, "A survey of ensemble learning approaches," *Journal of Yunnan University (Natural Sciences Edition)*, vol. 40, no. 6, pp. 1082–1092, 2018.
- [29] T. Q. Chen and C. Garlos, "Xgboost: a scalable tree boosting system," in *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, New York, NY, U.S.A, August 2016.
- [30] J. Gu and Z. Wang, J. Kuen, M. Lianyang, S. Amir et al., Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.
- [31] L. C. Yann, B. Yoshua, and H. Geoffrey, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.