IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 14, 2019, accepted November 30, 2019, date of publication December 5, 2019,
date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2957806

Generalized Pareto Model Based on Particle Swarm Optimization for Anomaly Detection

YAN HUANG[1,2], FUYU DU[2], JIAN CHEN[3], YAN CHEN[4], QICONG WANG[1,2], AND MAOZHEN LI[5]

[1]Shenzhen Research Institute, Xiamen University, Shenzhen 518000, China
[2]Department of Computer Science, Xiamen University, Xiamen 361000, China
[3]Third Institute of Oceanography Ministry of Natural Resources, Xiamen 361000, China
[4]College of Business and Management, Xiamen Huaxia University, Xiamen 361021, China
[5]Department of Electronic and Computer Engineering, Brunel University, London UB83PH, U.K.

Corresponding author: Qicong Wang (qcwang@xmu.edu.cn)

This work was supported in part by the Shenzhen Science and Technology Projects under Grant JCYJ20180306173210774, in part by the Scientific Research Foundation of Third Institute of Oceanography, MNR, under Grant 2019030, and in part by the NSFC under Grant 61671397.

ABSTRACT Anomaly detection of time series has been widely used in various fields. Most detection methods depend either on assumptions about data distribution or manual threshold setting. If the assumption is incorrect, the effectiveness of detection technology will be greatly reduced. To deal with this problem, we propose a maximum likelihood estimation method based on particle swarm optimization for generalized Pareto model to detect outliers of time series, which can be called Generalized Pareto Model Based on Particle Swarm Optimization (GPMPSO). Because the generalized Pareto model is multidimensional, we introduce a comprehensive learning strategy to improve search ability of particle swarm algorithm. Due to the multiple peaks of the log-likelihood function of generalized Pareto model, we apply dynamic neighbors to reduce the possibility of particle swarm optimization falling into local optimum. Moreover, we propose a new processing model Big Drift Streaming Peak Over Threshold (BDSPOT) to enhance the capability of the data stream processor. Our algorithm is tested on various real-world datasets which demonstrate its very competitive performance.

INDEX TERMS Anomaly detection, generalized pareto distribution, particle swarm optimization, time series.

I. INTRODUCTION

Anomaly Detection has been one of the most important research topics for a long time because it has the nature of ubiquity. In different application fields, anomalies can represent different problems. For example, in the field of Computer Science, anomalies refer to the data that deviate from others in the samples. While in the field of health care, it relates to the physiological signal of patient beyond the safety range specified by the doctor.

Owing to the variety of usage scenarios and algorithms, the methods of anomaly detection can be divided into different types from multiple perspectives. The most common division method is based on the degree of supervision, i.e., supervised, semi-supervised, and unsupervised techniques. Another division method is based on different algorithmic theories [1] which consist of statistics-based methods [2], [3], intelligent computing methods [4], [5], Bayesian networks, and other Bayesian reasoning extensions [6], [7], and model-based approaches [8]. Statistics-based methods are the most popular techniques in anomaly detection. These methods belong to data-based as well and they are able to detect the changes of anomalies. Bayesian networks are represented in the manner of digraph. They seem to be useful in detecting and isolating failures in an early stage. According to [1], intelligent computing methods, such as neural networks [9]–[11], support vector machine (SVM) [12], fuzzy theory and rough sets [13], [14], have obvious weakness. On the one hand, they need massive samples for training. On the other hand, the phases of training and testing are independent, which lead to the lack of abilities of continuous learning. And for model-based approaches, they all require prior knowledge including the distributions of data and the predefined threshold of judging an anomaly.

The associate editor coordinating the review of this manuscript and approving it for publication was Yongtao Hao.

As we all know, the state of a system is not static all the time. In this case, if the distribution of the data has changed, our prior knowledge of the system may not be correct any more. In consequence, the result we attained will make no sense if our method does not work with the changes.

It is worth noting that Siffer *et al.* [15] put forward the extreme value theory to solve this problem. What the most powerful of the theory is that it assumes that the distributions of extreme values and the dataset are different, i.e., the extreme value follows another distribution. As a result, it has no need to pay attention to the distribution of dataset and set a threshold manually. There are three kinds of extreme value distributions in the extreme value theory, which can be collectively referred to as Generalized Extreme Value (GEV) distribution. We can only choose an extreme value in each part when applying GEV to analyze data, hence the data we can analyze are quite small. To avoid setting threshold manually, we apply Generalized Pareto Distribution (GPD) to select the data larger than a certain threshold which is picked by Peak Over Threshold (POT). And we choose Maximum Likelihood Estimation (MLE) to compute the parameters of GPD. In [15], they proposed the Grimshaw tricks [16] to calculate the maximum value of maximum likelihood function. The precision of MLE is high, and we can get the root of MLE at a higher probability. However, it is sometimes difficult to find its root by calculation. Consequently, we treat it as an optimal problem, and take advantage of Particle Swarm Optimization (PSO) to find the optimal solution.

In PSO algorithm, every particle represents a probable solution. Through the simple behavior of individual particles, it can solve the problem by information interaction in the swarm. Traditional PSO algorithm performs well in low dimension. Whereas in high dimension, it tends to trap in local optimal solutions. To this end, many researchers proposed improved PSO algorithms [17]–[19]. It is known that the topological structure in traditional PSO algorithm is static, thus the learning sample of each particle is fixed as well. As a result, the probability of attaining local optimal solutions increases. However, in dynamic neighbor topology, each particle is able to build neighbors dynamically on the basis of its own operation to avoid local optimal solutions. In addition, Bergh and Engelbrecht [20] raises that since each dimension of a particle learns from the corresponding dimension of the same particle, there exists such situation, i.e., some dimensions of a particle move to optimal solution while the other away from optimal solution. Yet in comprehensive learning strategy, each particle learns the optimal position from all its neighbors, which ensures that each dimension of the particle is the current optimal, thus avoiding local optimal solution. Based on the above analysis, we develop the PSO algorithm based on dynamic neighbors and comprehensive learning strategy to solve the optimal solution of MLE.

Finally, Sniffer *et al.* [15] encapsulates the algorithm in (D)SPOT for data with drift in stream. But the model does not perform well in the following cases. The data are stable, but there is an abnormal drift at a sudden moment after which all observations are defined as anomalies. To address the problem, we propose a BDSPOT detection model, which can update the drift according to the distribution of data.

II. RELATED WORK

Time series data are observations collected at different time stamps to describe the changes of phenomena over time. As an important category of time data, researchers have paid much attention to time series. Anomaly detection is an important topic in time series analysis, which has been applied in many fields, such as fault detection, health care, weather analysis, financial and so on. Diverse anomaly detection methods have proposed for time series.

Munir *et al.* [21] proposed a deep learning approach called DeepAnT to detect outliers in time series. It uses datasets without labels, so it belongs to unsupervised methods. DeepAnT has two modules, i.e., time series predictor and anomaly detector. The former applies Convolutional Neural Network (CNN) to predict the next time stamp. The predicted value is the input of the latter and then the detector will mark it as normal or abnormal. DeepAnT can achieve good generalization capability. Furthermore, it is superior to the state-of-the-art methods. Whereas it has limits in the size of training datasets. Although Munir mentioned that it can be trained in a small dataset, the size of datasets is quite large compared to the general methods. For example, DeepAnT uses 60% of the NAB datasets for training, whereas the dataset is consist of 58 data streams of which there are 1000-22000 instances each. The amount of data is still large even if only use the 60%.

Hu *et al.* [1] proposed a meta-feature based approach called MFAD to detect anomalies in time series. It uses six meta-features (peak, coefficient of variation, oscillation, regularity, square wave, and trend change) to describe the dynamics in the original sequence. The similarity between samples dose MFAD compares with the meta-features to avoid calculating the distance between samples, which reduces the complexity of calculation. Compared to traditional anomaly detection approaches, MFAD has lower computational complexity and higher accuracy. Besides, it is more suitable for online learning adaptively.

Lu *et al.* [22] proposed an outlier detection algorithm based on cross-correlation analysis called ODCA for time series dataset. The key parts of ODCA is threefold. First of all, they shift the assembled anomalies to isolated ones in a way of linear interpolation method. Moreover, they convert the high-dimensional datasets into 1-D cross-correlation functions and therefore determine the isolated anomalies. To output corresponding anomalies hierarchically, they adopted Otsus method to pick up threshold for each level. ODCA is capable of detecting both assembled and isolated anomalies in high-dimensional datasets. Furthermore, it preforms better than major methods both in efficiency and time complexity. However, there exists several drawbacks like users still have to set some thresholds and know the prior knowledge about the dataset.

Zheng *et al.* [23] assume that anomaly detection is a statistical hypothesis testing, based on which they developed a novel model based on a synergistic combination of statistical and fuzzy set-based technique. They also adopted an intensive fuzzification process to determine the parameters, so users do not have to input the parameters. The experimental results show that the approach can detect anomalies in an early stage with the properties of fuzziness, adaptivity. But it is distance-based, the performance is not stable, especially for sparse data.

In short, there are so many mature approached with fine performance in anomaly detection, yet there still exist problems to be addressed. Anomaly detection methods are supposed to adapt to changes of data distributions in a relatively small datasets and without prior knowledge. We reviewed the anomaly detection algorithm, especially, we consider the extreme value theory [15] which do not have to assume the distribution of the datasets either to set thresholds manually. On this basis, we propose to obtain the optimal solution of maximum likelihood function of extreme value theory by PSO algorithm.

PSO algorithm is a global stochastic optimization algorithm based on swarm intelligence, which simulates the behavior of the cattle to realize searching optimal solution. It has received extensive attention from academic community, since it has simple concept and less parameters and easy to realize. PSO algorithm has been widely utilized in application domains as well. Currently, applying PSO algorithm to solve practical problems is a hot topic. In electric system domain, Ashour *et al.* [24] applied PSO algorithm to optimize multi-objective reactive power of electric system, which effectively solved the voltage stability problem of power grid. In the field of image processing, Xuan *et al.* [25] applied particle swarm optimization algorithm to multi-threshold image segmentation, which effectively solved the problem of multi-objective optimal search for multi-threshold images. In the field of time series, Li *et al.* [26] applied PSO to solve ARMA model parameters and obtained the probabilistic optimal digital solution. For the past few years, several optimized PSO algorithm has been put forward. For example, Hu and Yen [27] developed PSO algorithm based on dynamic parameters. Beheshti and Shamsuddin [28] proposed a method to aggregate weights in order to solve multi-objective optimization problem. Liu *et al.* [29] advanced a novel PSO based on adaptive rejection factor (ARFPSO).

Its successful applications have proved its strong ability to optimize search. PSO algorithm has an advantage on solving optimization problems. On the one hand, it can search the optimal solutions efficiently and avoid local optimal solutions by certain approaches. On the other hand, PSO algorithm can apply to diverse objective functions and constraints. At last, the search speed of PSO algorithm is fast due to it has parallelism to some extent. Therefore, we choose PSO to solve the optimal solution of the maximum likelihood function of the extreme value theory.

III. THEORETICAL BACKGROUND

In this section, we introduce the basic ideas of extremum theory. We try to explain the theory we used and describe how we use it to solve our problems (read the literature [30], [31] for more details).

In some fields, we can calculate the statistical threshold by defining the probability of abnormal events, and take this threshold as the criterion for judging abnormal events. And the mathematical description of this kind of problem is as follows. Suppose $X_1, X_2 \cdots, X_n$ is an independent and identically distributed time series, X is a random variable, then its cumulative distribution function is $F_{(x)} = P(X \leq x)$. The tail of its distribution is represented as $\overline{F}_{(x)}$, i.e. $\overline{F}_{(x)} = P(X > x)$. Given a probability q, we note $x > z_q$ is the quantile at level $1 - q$, i.e. $x > z_q$ is the smallest value which meets $P(X \leq z_q) \geq 1 - q$.

A. EXTREME VALUE THEORY

In the beginning, extreme value theory is used to predict the probability of extreme events, like earthquake and flood. Fisher and Tippett [32] and Gnedenko [33] obtained a very nice result, i.e., under weak conditions, these extreme events have the same distribution, no matter what the original data distribution is. We use the following theorem [31].

Theorem 1: There is an independent and identically distributed time series $X_1, X_2 \cdots, X_n$. If there are constant sequences $a_n > 0$ and b_n meet

$$\lim_{n\to\infty} Pr(\frac{M_n - b_n}{a_n}) = H(x), \quad x \subseteq R \tag{1}$$

$H(x)$ is a non-degenerate distribution function, then H must belong to one of the following three types.

$$H_1(x) = \exp\{e^{-x}\}, -\infty < x < +\infty; \tag{2}$$

$$H_2(x; a) = \begin{cases} 0, x \leq 0, \\ exp{-x^{-a}}, x > 0, \end{cases} \quad a > 0; \tag{3}$$

$$H_3(x; a) = \begin{cases} \exp\{-(-x)^{-a}\}, x \leq 0 \\ 1, x > 0, \end{cases} \quad a > 0 \tag{4}$$

These three types of distributions are collectively called extremum distributions, which can also be uniformly expressed as

$$H(x; k, b, a) = \exp\left\{-\left(1 + k\frac{x-a}{b}\right)^{-1/k}\right\}, \quad 1 + k\frac{x-a}{b} > 0 \tag{5}$$

Here, the positional parameter a and the scale parameter b are introduced, and k is the shape parameter, where $a, k \subseteq R, b > 0$. We call H the GEV or GED.

B. GENERALIZED PARETO DISTRIBUTION

When we use GEV to analyze data, we can only select the maximum value (or minimum value) of each region. For example, if we want to analyze the temperature in an area,

we can only select the highest or lowest temperature in the area every week or every month. And this leaves us with very little data to analyze. So we consider analyzing data that exceeds a certain threshold μ, and we use the following definitions and properties of Generalized Pareto distribution (GPD) [31].

Definition 2: If the distribution function of the random variable X is

$$G(x;k,b,a) = 1 - \left(1 + k\frac{x-a}{b}\right)^{-1/k}, \quad x \geq a, 1 + k\frac{x-a}{b} > 0 \tag{6}$$

where we think X follows the generalized Pareto distribution, abbreviated as GPD, k is the shape parameter, b is the scale parameter, and a is the position parameter.

It is easy to known that the probability density function of its distribution is

$$g(x;k,b,a) = \frac{1}{b}\left(1 + k\frac{x-a}{b}\right)^{-1/k-1}, \quad x \geq a, 1 + k\frac{x-a}{b} > 0 \tag{7}$$

Definition 3: There is an independent and identically distributed time series $X_1, X_2 \cdots, X_n$, their distribution function is $F(x)$. Let $M_n = \{X_1, X_2 \cdots, X_n\}$, if there are constant sequences $a_n > 0$ and b_n meet

$$Pr(M_n \leq a_n x + b_n) \approx H(x;k,b,a) \tag{8}$$

where $H(x;k,b,a)$ is GEV. Then for a sufficiently large threshold μ, when $X > \mu$, $X - \mu$ approximately obeys the generalized Pareto distribution

$$G(y;k,\overline{b}) = 1 - (1 + k\frac{y}{b})^{-1/k}, \quad y > 0, 1 + k\frac{y}{b} > 0 \tag{9}$$

where $\overline{b} = b + k(\mu - a)$. We refer to the method of selecting data with a data set greater than a certain threshold μ as the fitting data of the generalized Pareto model as the Peak Over Threshold (POT) model. And we use the maximum likelihood estimation to estimate the parameters of the generalized Pareto model.

Property 4: If random value $X \sim G(k,b,a)$, then the average excess function is

$$e(\mu) = \frac{b - ka}{1-k} + \mu\frac{k}{1-k}, \; k < 1, b + k(\mu - a) > 0, \tag{10}$$

i.e. $E(X - \mu | X > \mu) = e(\mu)$. Here we can see that the threshold μ and function value $e(\mu)$ change linearly.

Property 5: There is an independent and identically distributed time series $X_1, X_2 \cdots, X_n$, and there is a threshold value μ makes any $X_i > \mu$ follow the distribution $G(k,b,a)$, then if we know the parameters k, b, a, and we can calculate the quantiles as follows [15].

$$z_q = \mu + a + \frac{b}{k}\left(\left(\frac{qn}{N_t}\right)^{-k} - 1\right) \tag{11}$$

where q is the expected probability, n is the number of observed values, N_t is the peak number.

C. PEAK OVER THRESHOLD (POT)

We use POT to select the data instances that are larger than a threshold μ. As described in definition 2, these instances follows GPD. And it is obvious that the value of threshold affects the accuracy of parameter estimation, and the parameter estimation affects the correctness of data analysis. Because if the threshold is too small, the selected instances may do not follow GPD. If it is too big, we cannot get enough data to fit the model.

Here we choose a method to select the threshold, and it is also a method to judge whether the data set contains a subset of data that conforms to GPD. The problem can be described as follows.

There is an independent and identically distributed time series $X_1, X_2 \cdots, X_n$, and $X_{1:n} \leq X_{2:n} \cdots X_{n:n}$ is its order statistic. Select a value k such that $X_{k:n} \leq X_{k+1:n} \cdots X_{n:n}$ can be fitted with GPD, then the threshold μ is $X_{k:n}$.

According to the property 4, the average excess and the threshold μ vary linearly. We can use the empirical mean excess function [34] to calculate the average excess

$$e_n(\mu) = \frac{1}{*\{i | X_i > \mu, 1 \leq i \leq n\}} \sum_{i=1}^{n} (X_i - \mu)_{>0} \tag{12}$$

where $*$ indicates the number of i, and $(X_i - \mu)_{>0}$ indicates that we only take the values which are greater than 0. If there is a value k meets that the points $(\mu, e_n(\mu))$ fluctuate around a straight line when $\mu > X_{k:n}$. And we can set $u_0 = X_{k_{min}:n}$. This phenomenon can also be used to determine whether the GPD model can be used for fitting.

IV. MAXIMUM LIKELIHOOD ESTIMATION BASED ON PARTICLE SWARM OPTIMIZATION

A. THEORETICAL SUPPORT

1) MAXIMUM LIKELIHOOD ESTIMATION (MLE)

Maximum likelihood estimation is an application of probability theory in statistics. It is one of the methods of parameter estimation. Suppose that a random sample satisfies a certain probability distribution, but the specific parameters are unknown. The parameter estimation is to observe the results through several experiments, and use the results to derive the approximate values of the parameters. The maximum likelihood estimation is based on the idea that a certain parameter is known to maximize the probability of occurrence of this sample. Of course, we will not choose other small probability samples, so we simply use this parameter as the estimated reality value.

The mathematical description of the maximum likelihood estimation [15] is as follows. If $X_1, X_2 \cdots, X_n$ are n independent observations of random variable X, and the probability density function $f_\vartheta(x)$ is represented by the parameter ϑ (possibly a vector), then the likelihood expression can be defined as follows.

$$L(X_1, X_2, \cdots X_n; \vartheta) = \prod_{i=1}^{n} f_\vartheta(X_i) \tag{13}$$

It represents the joint probability of n observations. Since $X_1, X_2, \cdots X_n$ are fixed in the context, we try to find the parameter ϑ that maximizes the value of the expression, which is what we want. In fact, we generally use the log likelihood form. For the GPD model, we need to maximize

$$\log L\left(X_1, X_2, \cdots X_n; k, b\right) = -N_t \log b - \left(\frac{1}{k} + 1\right) \sum_{i=1}^{n} \log\left(1 + \frac{k}{b} Y_i\right) \quad (14)$$

where Y_i is the excess, i.e. $Y_i = X_i - \mu$. In addition, since the log likelihood of the three-parameter form is monotonically increasing and unbounded with respect to the positional parameter a, it is better to choose the data to approach the model than to estimate the positional parameter.

2) MAXIMUM LIKELIHOOD ESTIMATION BASED ON PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) [35] is a relatively new optimization technique in optimization algorithms. It searches by mimicking the behavior of herds and flocks. Due to its simple concept, few control parameters, simple implementation, and certain parallelism, it has received extensive attention from the academic community since its introduction.

Particle swarm optimization is a cluster-based intelligent algorithm. Each member of the population is called a particle and represents a potential feasible solution. The population searches for the optimal solution in the D-dimensional space. Each particle has two basic properties, i.e., current position and flight speed. During the flight, each particle has the ability to remember, they will record their historical optimal solution and the historical optimal solution of the entire population (that is, the parameters value corresponding to the maximum value of the function). In order to approximate the position of the optimal solution, each particle learns from its optimal position and the optimal position of the population. The mathematical description of the original particle swarm algorithm is as follows. Suppose the population size is N, which means there are N particles in the population. At t iteration time, the position of each particle in the D-dimensional space is $x_i^t = (x_i^1, x_i^2, \cdots, x_i^D)$, and the velocity vector of the particle is $v_i^t = (v_i^1, v_i^2, \cdots, v_i^D)$. The updating rules of its position and velocity at $t+1$ are as follows.

$$v_i^{t+1} = v_i^t + c_1 r_1 (b_i^t - x_i^t) + c_2 r_2 (b_g^t - x_i^t) \quad (15)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (16)$$

$$v_i^d = v_{max}^d, \quad if\ v_d^i > v_{max}^d \quad (17)$$

$$v_i^d = -v_{max}^d, \quad if\ v_d^i < -v_{max}^d \quad (18)$$

where b_i^t represents the best position of particle i, b_g^t represents the best position of the whole swarm, c_1 and c_2 indicate the particle confidences in itself cognition and in the swarm social behavior respectively, which usually take values from 0 to 2, and r_1 and r_2 are two random numbers uniformly distributed in the range from 0 to 1.

In order to improve the problem of easy to fall into the local optimal solution when solving complex multi-peak problems, we design a particle swarm optimization algorithm based on dynamic neighbor and comprehensive learning.

1) Dynamic neighbor. The particle swarm algorithm converges to the optimal solution through the mutual learning of the particles in the population, and the link is the neighbor topology between the particles.

In [19], [36], [37], some dynamic neighborhood topology based particle swarm optimization algorithms are proposed to improve the search range of particles by adjusting the topology of neighbors. If the neighbor topology of a particle is unchanged from the beginning of learning to the end, the topology is static. And if the learning samples of the particle have fallen into local optimum, then the particle is likely to learn them and converge to the local optimum, which can increase the probability that the population will fall into the local optimal solution.

It should be noted that the log-likelihood function of the generalized Pareto model may have multi-peak phenomenon, so the possibility that the original particle swarm algorithm falls into the local optimal solution is greatly increased. Therefore, in order to overcome this shortcoming, we adopt a dynamic neighbor topology to update neighbors of each particle according to its running state during iterative process of the algorithm. The definition is as follows.

$$M_i^t = \left\{d_{in} | d_{in} = \left\|x_i^t - x_n^t\right\|, n \neq i, n \subseteq PS\right\} \quad (19)$$

$$Neighbor_i^t = pick\left\{min[sort(L_i^t), m]\right\} \quad (20)$$

where M_i^t is the Euclidean distance set of the particle i and other particles in the population at the time t, PS is the population size, $Neighbor_i^t$ is the neighbors of the particle i, $pick$ means identifying the position of the particle, and m is the number of neighbors of a particle, which is $1/3$ of PS in this paper. In order to avoid the whole population falling into local optimum, its neighbors need to be updated if a particle does not update its historical optimum in successive iterations.

2) Comprehensive learning strategy. In traditional PSO, each dimension of a particle learns from the corresponding dimension of the same learning samples. Some dimensions may move closer to the optimal solution, whereas others may be farther away from the optimal solution [20]. So the comprehensive learning is developed to improve the ability of particles to jump out from the local optimal solution [38], [39]. In this paper, the generalized Pareto model is at least a two-parameter model, and the solution space is at least two-dimensional space. If only one learning sample is referenced for each dimension of movement, the above situation may happen. To this end, we introduce a comprehensive learning strategy in which the learning samples for

TABLE 1. MLE + Grimshaw.

	k			b		
n	true value	Bias	MSE	true value	Bias	MSE
50	-2	-2	4	1	0.668	0.47
	-1	-1	1	1	0.498	0.249
	1	1	1	1	-10.69	1.8e3
	2	2	4	1	-1.02e4	3.3e9
200	-2	-2	4	1	0.667	0.445
	-1	-1	1	1	0.501	0.251
	1	1	1	1	-10.03	540.5
	2	2	4	1	-19342	9.3e9

TABLE 2. MLE + PSO.

	k			b		
n	true value	Bias	MSE	true value	Bias	MSE
50	-2	0.344	0.122	1	0.172	0.030
	-1	0.34	0.121	1	0.17	0.030
	1	0.895	0.801	1	0.447	0.200
	2	0.895	0.801	1	0.447	0.200
200	-2	0.338	0.181	1	0.169	0.029
	-1	0.342	0.121	1	0.171	0.030
	1	0.895	0.801	1	0.447	0.200
	2	0.895	0.801	1	0.447	0.200

each dimension of the particle are determined as follows.

$$p^d_{bin(i)} = pick\left\{max[\frac{Log(x_i) - Log(p_j)}{x^d_i - p^d_j}]\right\}, \quad i \subseteq PS, j \subseteq Neighbor_i \tag{21}$$

$$v^{t+1}_i = v^t_i + c_1 r_1(p^t_i - x^t_i) + c_2 r_2(p^t_{bin(i)} - x^t_i) \tag{22}$$

We make some modifications according to the theory of the traditional methods. $Log(x_i)$ is the logarithmic Maximum Likelihood Function of Generalized Pareto Model. The part that learns optimally from itself is not merged into $p^t_{bin(i)}$. Self-optimal is the itself cognition part, and $p^t_{bin(i)}$ is the swarm social part. In addition, we also change the specific parameters, and changes the selection criteria of the social part to the adaptive function value of the particle at t iteration, rather than its historical optimal value, which can better reflect the distance between the particle and the optimal position of each particle at iteration time t, thereby determining the learning samples of the social part of the d-dimensional.

B. EVALUATION EXPERIMENT

In this section, we verify the effectiveness of the algorithm through simulation studies, and compare the advantages and disadvantages of the algorithm and the original algorithm through experimental results.

This section uses the random number of GP distribution generated by matlab as the data source. Since the effect of the parameter estimation depends on many factors, such as sample size, parameter size, and algorithm stability. Therefore, we generated 8 sets of data according to the sample size of 50, 200 shape parameters k of $-2, -1, 1, 2$ to compare the fitting effect of the algorithm. Here, set $b = 1, a = 0$. The estimated effect is evaluated by bias and mean square error (MSE). The formula is as follows.

$$Bias = E(\vartheta_{est} - \vartheta_{true}), MSE = E[(\vartheta_{est} - \vartheta_{true})^2] \tag{23}$$

where ϑ_{est}, ϑ_{true} are the estimated value and the true value of the parameter.

Table.1 and Table.2 show the results of the original algorithm and the algorithm of this section for fitting 100 times of 8 data sets.

As shown in the above table, since the original algorithm uses the Grimshaw method for parameter estimation calculation, in some cases, the numerical solution of the equation root cannot be obtained, and the final result is only $\vartheta = \frac{k}{b} = 0$. At this time, $k = 0$. In this case, the detection effect in the application field basically depends on the estimation of the scale parameter b. It can be seen in the experimental chapter that although some data can still be detected, the effect is not ideal, there is a slight gap compared to other algorithms. The PSO algorithm does not need to find the roots, and successfully avoids problems that the roots sometimes cannot obtain. However, there are some problems in the PSO algorithm applied in this section. For example, the execution time has a certain probability problem. It depends on the random movement of particles to find the actual optimal solution in time. Therefore, the execution time may be slightly longer than the original algorithm. But considering the application of the POT model, our program only processes part of the data, so the execution time is acceptable. In addition, this section uses a constrained PSO, so it requires some prior knowledge, such as the approximate range of shape parameter k and scale parameter b. The more accurate the estimation range, the better detection effect when applied in the actual scene.

V. FLOW DETECTOR

When dealing with time series, the environment often requires us to perform real-time analysis, and we cannot scan the second time. Therefore, we need a data stream processing model to encapsulate the detection algorithm. This section mainly introduce the data stream processing model— (D)SPOT, and present our view.

A. (D)SPOT

The process of SPOT is as follows. First, we use the first n data given as initialization data to initialize the peak threshold μ and the abnormal threshold z_q. Then it starts accepting input from data stream. For each data instance, system determines its state. If the status is abnormal, the system issues an alert. As for normal data, if it exceeds the peak threshold μ, then it is added to peak set and used to update the system status.

SPOT is only available when the data are ''stationary'', i.e. the distribution of data does not drift. So Siffer *et al.* purposed (D)SPOT, which uses the relative value of the data

instead of the absolute value. In the initialization phase, (D)SPOT initializes the moving average M_i using the first d (called window parameter) data. Then use the relative value of the remaining initialization data $X_i' = X_i - M_i$ to initialize the peak threshold μ and the abnormal threshold z_q. In the detection phase, system also uses the relative value of the data. At the same time, use the data to update the moving average $M_i = \frac{1}{d}\sum_{k=1}^{d} X_{i-k}^*$ where $X_{i-d}^*, X_{i-d+1}^* \cdots X_{i-1}^*$ is the most recently observed values.

B. BDSPOT

In [15], the (D)SPOT flow detector is used to detect anomalies with offsets in the data set. This method selects normal data when calculating the average value. However, it should be noted that in the experiment, we found that although this calculation method can prevent the too high abnormal value from leading to an excessively high average in the scene with stable data, which may lead to the high statistical threshold, but in the following case, the effect of this method is not ideal, i.e., the data are stable (the data do not drift or has a gentle drift), but there is an abnormal level drift at a certain moment, because the original method will treat all observations after this drift as anomalies, resulting in the moving average is not updated, so the detection after this will be invalid. Therefore, for the above case, we made a few changes to the (D)SPOT flow detector.

Because the data are generally stable, in general, the update of the mean still uses non-outliers. However, once the system detects an abnormality for d consecutive times, the system will consider that its data distribution has changed, then the next m values are updated for moving averages, whether they are abnormal or not. The values here are determined according to the stability of the system to be tested and the detection requirements. The specific values will affect the false positive rate of the test (the actual number of instances that are normal but detected as abnormal accounts for the percentage of all detected abnormalities) and the false negative rate. (The actual number of instances that are abnormal but are detected as normal accounts for the percentage of all tests that are normal).

VI. EXPERIMENT

Before the GPD model is applied to detect anomalies, it has played an import role in earthquake, flood, finance and so on. The measurement of water level is one of the most basic means in the process of mastering hydrological information. Modern water level measurement is mainly done by water level data acquisition system. Many water level data acquisition systems work in rivers, reservoirs, and other places for a long time. In general, the data density is very large. It is obviously unrealistic to monitor the changes by human resources alone, and the occurrence of its maximum value (or minimum value) may represent flood (or drought) or mechanical damage (such as dam damage, etc.). Therefore, it is very meaningful to detect the maximum value (minimum value) of the water level data.

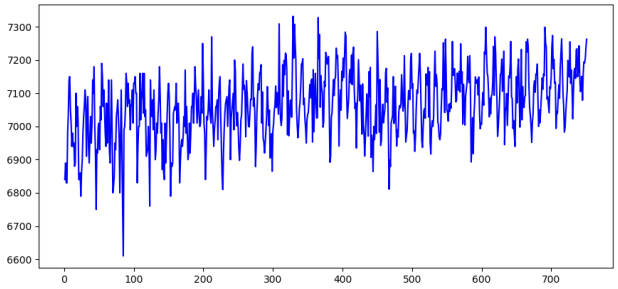

FIGURE 1. Sea level data at a site in Reykjavik.

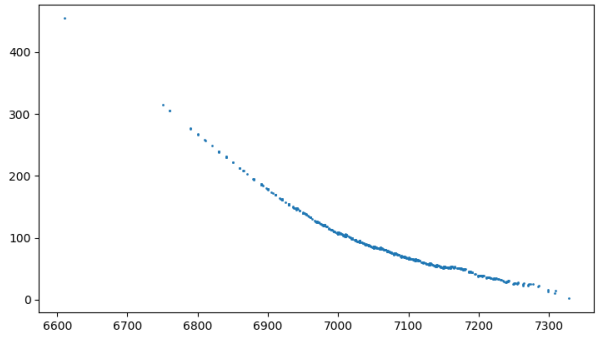

FIGURE 2. Empirical average excess function for Reykjavik sea level data.

In order to evaluate the effects of the detection algorithm and the processing model, we applied it to the anomaly detection of sea level data. We downloaded the sea level data of a site in San Francisco and the sea level data of a site in Reykjavik through matlab in PSMSL (`http://www.psmsl.org/data/obtaining/rlr.monthly.data/10.rlrdata`), each of which has its own characteristics, i.e., the water level data of a site in San Francisco has a smooth drift and the water level data of a site in Reykjavik is "stationary". In addition, to prove the situation that BDSPOT can handle, we changed the first data set (shifting the second half of the data up) as the third data set.

A. THE SEA LEVEL DATA OF A SITE IN REYKJAVIK

As shown in Fig. 1, the sea level data at a site in Reykjavik does not drift, so we use the SPOT model to process the data.

1) THRESHOLD SELECTION

As can be seen from the Fig. 2, the point at the end of the image fluctuates roughly in the vicinity of a straight line. By contrast, we choose a value greater than 7133, which is 25%, and the expected probability is 0.01.

2) SIMULATION DISPLAY

We use two algorithms to carry out simulation experiments, and the detection effects are as follows. The lines with different colors and the decimals after them in the legend correspond to different algorithms and the percentages of detecting outliers. In Fig. 3, the blue curve represents raw data, the pink

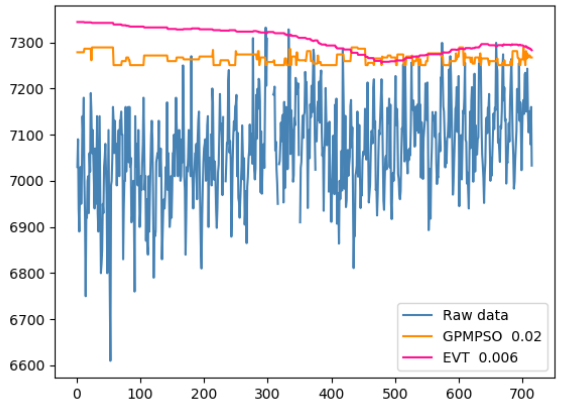

FIGURE 3. Simulated effects of various algorithms for Reykjavik sea level data.

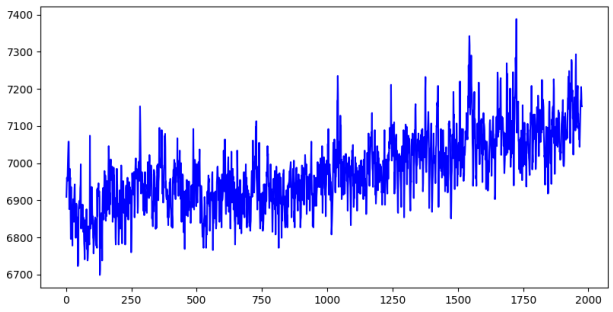

FIGURE 4. Sea level data at a site in San Francisco.

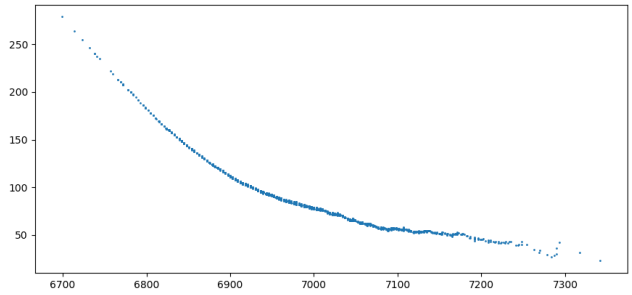

FIGURE 5. Empirical average excess function for San Francisco sea level data.

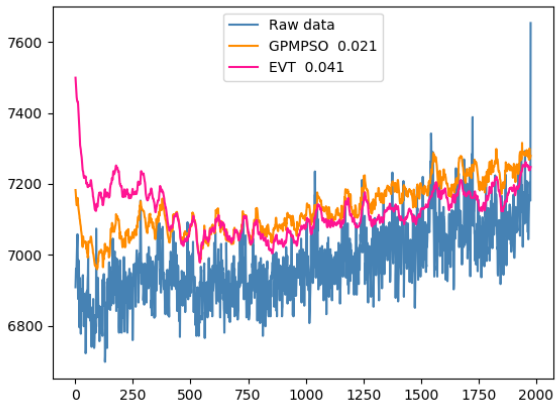

FIGURE 6. Simulated effects of various algorithms for San Francisco sea level data.

curve represents the result of the original EVT algorithm, and the orange curve is the result of GPMPSO algorithm. This experiment used partial data for initialization (not shown in Fig. 3). The figure shows the effect of the detection step after initialization. It can be seen from the figure that the detection effect of the original EVT algorithm is not good because the estimated value of k is 0. The prior knowledge of the parameters of the GPMPSO algorithm is relatively accurate, so the effect of the proposed approach is very good, close to the prediction effect.

B. THE SEA LEVEL DATA AT A SITE IN SAN FRANCISCO

As shown in Fig. 4, the sea level data at a site in San Francisco drifts smoothly over time, so we use the (D)SPOT model to process the data.

1) THRESHOLD SELECTION

As can be seen from the Fig. 5, the point at the end of the image fluctuates roughly in the vicinity of a straight line. By contrast, we choose a value greater than 7043, which is 25%, and the expected probability is 0.01.

2) SIMULATION DISPLAY

We performed the simulations for two algorithms, and the results are as follows. All the labels in Fig. 6 represent the same in Fig. 3. This experiment used partial data for initialization (not shown in Fig. 6). The figure shows the effect of the detection step after initialization. Because of the cliff-like drop at the junction of the two parts of data, the initial detection is not good. The up and down fluctuations of the two algorithms are similar. The prior knowledge of the parameters of the GPMPSO algorithm is relatively accurate, so the detection effect of our approach is very good, which is the closest to the prediction effect. However, the original EVT algorithm has a relatively poor detection effect because the data set contains some very large or very small values.

C. PROCESSED SEA LEVEL DATA

We changed the San Francisco sea level data (shifting the second half of the data up) as the third data set. As the figure shows, the data have a big jump at some point, so we use the BDSPOT. Regarding the selection of the threshold, we can use the threshold selected above.

1) SIMULATION DISPLAY

The simulations were performed using two algorithms, and the effects are as follows. The label content in Fig. 8 is the same as Fig. 3. The up and down fluctuations of the two algorithms are similar. The GPMPSO algorithm has the best effect in this experiment, the detection ratio is 0.051. The original algorithm has relatively poor detection effect. Because the data have a big jump, it takes a certain time for the system to adapt to the change of the data distribution, and

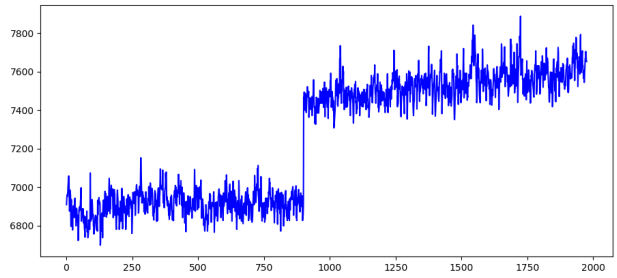

FIGURE 7. Processed sea level data.

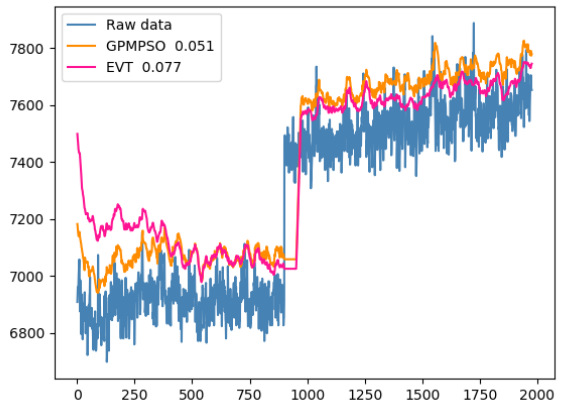

FIGURE 8. Simulated effects of various algorithms for processed sea level data.

the detection ratio of the two algorithms is larger than the result in the last subsection.

VII. SUMMARY

In this paper, based on the extreme value theory proposed in the literature [15], a new attempt is made. During the research process, we found that the maximum likelihood estimation based on the root-seeking method has certain defects, and the proposed method—GPMPSO treats solving problems as optimal problems. But Each of the two algorithms has a suitable field. We cannot guarantee that an algorithm will perform better than any other algorithm in any scene. However, from the above experiments, if the given parameter range is sufficiently accurate, the maximum likelihood estimation based on particle swarm optimization is generally the best. In addition, we have made improvements to (D)SPOT for the distribution of data.

However, there are still many aspects to the method that can be further improved. First of all, in the field of application, this article is still only limited to the case of a one-digit sequence. Secondly, the PSO algorithm we apply requires some prior knowledge when performing parameter estimation. That is, the approximate range of parameters, which will lead to the complication of the work. Therefore, we need to be able to automatically estimate the parameter range.

Finally, the BDSPOT model has a certain adaptation time when the data distribution changes, which will cause the system to fail for a period of time. Therefore, how to adapt to the changes in data distribution as accurately and quickly as possible is worth exploring.

ACKNOWLEDGMENT

(Yan Huang and Fuyu Du are co-first authors.)

REFERENCES

[1] M. Hu, Z. Ji, K. Yan, Y. Guo, X. Feng, J. Gong, X. Zhao, and L. Dong, "Detecting anomalies in time series data via a meta-feature based approach," *IEEE Access*, vol. 6, pp. 27760–27776, 2018.
[2] E. Garoudja, F. Harrou, Y. Sun, K. Kara, C. Aissa, and S. Silvestre, "Statistical fault detection in photovoltaic systems," *Sol. Energy*, vol. 15, pp. 485–499, Jul. 2017.
[3] M. Madakyaru, F. Harrou, and Y. Sun, "Improved data-based fault detection strategy and application to distillation columns," *Process Saf. Environ. Protection*, vol. 107, pp. 22–34, Apr. 2017.
[4] L. Dong, S. Liu, and H. Zhang, "A method of anomaly detection and fault diagnosis with online adaptive learning under small training samples," *Pattern Recognit.*, vol. 64, pp. 374–385, Apr. 2017.
[5] H. Mekki, A. Mellit, and H. Salhi, "Artificial neural network-based modelling and fault detection of partial shaded photovoltaic modules," *Simul. Model. Pract. Theory*, vol. 67, pp. 1–13, Sep. 2016.
[6] J. C. M. Oliveira, K. V. Pontes, I. Sartori, and M. Embiruçu, "Fault detection and diagnosis in dynamic systems using weightless neural networks," *Expert Syst. Appl.*, vol. 84, pp. 200–219, Oct. 2017.
[7] Z. Ji, Q. Xia, and G. Meng, "A review of parameter learning methods in Bayesian network," in *Advanced Intelligent Computing Theories and Applications*, D.-Huang and K. Han, Eds. Cham, Switzerland: Springer, 2015, pp. 3–12.
[8] M. Gan, C. L. P. Chen, H. X. Li, and L. Chen, "Gradient radial basis function based varying-coefficient autoregressive model for nonlinear and nonstationary time series," *IEEE Signal Process. Lett.*, vol. 22, no. 7, pp. 809–812, Jul. 2015.
[9] C. K. Lau, K. Ghosh, M. A. Hussain, and C. R. C. Hassan, "Fault diagnosis of Tennessee Eastman process with multi-scale PCA and ANFIS," *Chemometrics Intell. Lab. Syst.*, vol. 120, pp. 1–14, Jan. 2013.
[10] J. Zarei, "Induction motors bearing fault detection using pattern recognition techniques," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 68–73, Jan. 2012.
[11] S. Kanarachos, S.-R. G. Christopoulos, A. Chroneos, and M. E. Fitzpatrick, "Detecting anomalies in time series data via a deep learning algorithm combining wavelets, neural networks and Hilbert transform," *Expert Syst. Appl.*, vol. 85, pp. 292–304, Nov. 2017.
[12] Z. Ji, B. Wang, S. P. Deng, and Z. You, "Predicting dynamic deformation of retaining structure by LSSVR-based time series method," *Neruocomputing*, vol. 137, pp. 165–172, Aug. 2014.
[13] G. C. Silva, R. M. Palhares, and W. M. Caminhas, "Immune inspired fault detection and diagnosis: A fuzzy-based approach of the negative selection algorithm and participatory clustering," *Expert Syst. Appl.*, vol. 39, no. 16, pp. 12474–12486, 2012.
[14] Z. Geng and Q. Zhu, "Rough set-based heuristic hybrid recognizer and its application in fault diagnosis," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 2711–2718, 2009.
[15] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouét, "Anomaly detection in streams with extreme value theory," in *Proc. 23rd ACM SIGKDD Int. Conf.*, 2017, pp. 1067–1075.
[16] S. D. Grimshaw, "Computing maximum likelihood estimates for the generalized Pareto distribution," *Technometrics*, vol. 35, no. 2, pp. 185–191, 1993.
[17] W. Dong and M. Zhou, "A supervised learning and control method to improve particle swarm optimization algorithms," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 7, pp. 1135–1148, Jul. 2017.
[18] S. U. Khan, S. Yang, L. Wang, and L. Liu, "A modified particle swarm optimization algorithm for global optimizations of inverse problems," *IEEE Trans. Magn.*, vol. 52, no. 3, Mar. 2016, Art. no. 7000804.
[19] C. Wang, Y. Liu, and Y. Zhao, "Application of dynamic neighborhood small population particle swarm optimization for reconfiguration of shipboard power system," *Eng. Appl. Artif. Intell.*, vol. 26, no. 4, pp. 1255–1262, 2013.
[20] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–239, Jun. 2004.

[21] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: A deep learning approach for unsupervised anomaly detection in time series," *IEEE Access*, vol. 7, pp. 1991–2005, 2019.
[22] H. Lu, Y. Liu, Z. Fei, and C. Guan, "An outlier detection algorithm based on cross-correlation analysis for time series dataset," *IEEE Access*, vol. 6, pp. 53593–53610, 2018.
[23] D. Zheng, F. Li, and T. Zhao, "Self-adaptive statistical process control for anomaly detection in time series," *Expert Syst. Appl.*, vol. 57, pp. 324–336, Sep. 2016.
[24] K. Jagatheesan, B. Anand, S. Samanta, N. Dey, A. Baskaran, and V. E. Balas, "Design of a proportional-integral-derivative controller for an automatic generation control of multi-area power thermal systems using firefly algorithm," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 2, pp. 503–515, Mar. 2019.
[25] T. X. Pham, P. Siarry, and H. Oulhadj, "Integrating fuzzy entropy clustering with an improved PSO for MRI brain image segmentation," *Appl. Soft Comput.*, vol. 65, pp. 230–242, Apr. 2018.
[26] C. Li, E. Chi, L. He, and Z. Li, "Prediction of nonstationary downburst wind velocity based on time-varying arma and emd-pso-lssvm algorithms," *J. Vib. Shock*, vol. 17, pp. 33–38, 2016.
[27] W. Hu and G. G. Yen, "Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 1–18, Feb. 2015.
[28] Z. Beheshti and S. Shamsuddin, "Non-parametric particle swarm optimization for global optimization," *Appl. Soft Comput.*, vol. 28, pp. 345–359, Mar. 2015.
[29] Y. Liu, Z. Zhang, Y. Luo, and X. Wu, "An improved PSO for multimodal complex problem," in *Intelligent Computing in Bioinformatics*, D. Huang, K. Han, and M. Gromiha, Eds. Cham, Switzerland: Springer, 2014, pp. 371–378.
[30] E. J. Gumbel, "Statistics of extremes," *Empire Surv. Rev.*, vol. 15, no. 114, pp. 187–190, 2011.
[31] D. Shi, *Practical Extreme Value Statistical Method*. Tianjin, China: Tianjin Science and Technology Press, (in Chinese), 2006.
[32] R. A. Fisher and L. H. C. Tippett, "Limiting forms of the frequency distribution of the largest or smallest member of a sample," *Math. Proc. Cambridge Philos. Soc.*, vol. 24, no. 2, pp. 180–190, Apr. 1928.
[33] B. Gnedenko, "Sur la distribution limite du terme maximum d'une serie aleatoire," *Ann. Math.*, vol. 44, pp. 423–453, Jul. 1943.
[34] H. Wei, "The application of extreme value statistics in catastrophe insurance," Ph.D. dissertation, Henan Univ., Kaifeng, China, 2008.
[35] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.
[36] X. Hu and R. Eberhart, "Multiobjective optimization using dynamic neighborhood particle swarm optimization," in *Proc. IEEE World Congr. Comput. Intell.*, vol. 2. May 2002, pp. 1677–1681.
[37] Y.-M. Liu, Q.-Z. Zhao, and B. Niu, "Improved particle swarm optimization based on adaptive dynamic neighborhood and generalized learning," *J. Comput. Appl.*, vol. 30, no. 10, pp. 2578–2581, 2010.
[38] Y. Cao, H. Zhang, W. Li, M. Zhou, Y. Zhang, and W. A. Chaovalitwongse, "Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 718–731, Aug. 2019.
[39] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.

YAN HUANG is currently pursuing the M.S. degree in computer science with Xiamen University, Xiamen, China. Her current research interests include time series analysis and computer vision.

FUYU DU received the B.S. degree in computer science from Xiamen University, Xiamen, China, in 2019. His current research interests include time series analysis and information security.

JIAN CHEN received the Ph.D. degree from Zhejiang University, Hangzhou, China. He is currently a Research Professor with the Third Institute of Oceanography, Ministry of Natural Resources, China. He have undertaken many marine geology investigation projects, such as Multiple Disciplinary Investigation in Fujian Coastal Zone, Marine Geology Survey in Taiwan Strait, Sea Sand Resources Evaluation in China Coastal Sea, and Study of Environment Variability and Ecosystem Regulation Strategy in Minjiang Estuary. He has published monographs, such as the China Island Annals in South Fujian, Impacts of Discharged Materials Variability of Minjiang River to Minjiang Estuary and Adjacent Area, and Report of Integrated Investigation and Evaluation in Fujian Coastal Sea. His research interests include marine information processing and analytic, marine sedimentology, marine sediment dynamics in coastal zone, sea sand resources prospect, and coastal zone and island vulnerability study and sustainable development.

YAN CHEN received the M.S. degree from the College of Business, Arts and Social Sciences, Brunel University London, U.K., in 2018. She is currently an Assistant Teacher with the College of Business and Management, Xiamen Huaxia University, Xiamen, China. Her current research interest includes data analysis and modeling.

QICONG WANG received the Ph.D. degree in information and communication engineering from Zhejiang University, Hangzhou, China, in 2007. He is currently an Associate Professor with the Department of Computer Science, Xiamen University, Xiamen, China. His research interests include computer vision, machine learning, big data analytic, and marine information processing and analytic.

MAOZHEN LI received the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences, in 1997. He is currently a Professor with the Department of Electronic and Computer Engineering, Brunel University London, U.K. His main research interests include high performance computing, big data analytic and intelligent systems with applications to smart grid, and smart manufacturing and smart cities. He has over 180 research publications in these areas, including four books. He has served over 30 IEEE conferences and is on the editorial board of a number of journals. He is a Fellow of the British Computer Society and IET.