sensors

MDPI

Article

An Enhanced RIME Optimizer with Horizontal and Vertical Crossover for Discriminating Microseismic and Blasting Signals in Deep Mines

Wei Zhu [1], Zhihui Li [1], Ali Asghar Heidari [2], Shuihua Wang [3,4], Huiling Chen [5,*] and Yudong Zhang [4,*]

[1] School of Resources and Safety Engineering, Central South University, Changsha 410083, China; csuzhuwei@csu.edu.cn (W.Z.); zhihuili0807@gmail.com (Z.L.)
[2] School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran 1417466191, Iran; aliasghar68@gmail.com
[3] Department of Biological Sciences, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China; shuihuawang@ieee.org
[4] School of Computing and Mathematical Sciences, University of Leicester, Leicester LE1 7RH, UK
[5] Key Laboratory of Intelligent Informatics for Safety & Emergency of Zhejiang Province, Wenzhou University, Wenzhou 325035, China
* Correspondence: chenhuiling.jlu@gmail.com (H.C.); yudongzhang@ieee.org (Y.Z.)

Abstract: Real-time monitoring of rock stability during the mining process is critical. This paper first proposed a RIME algorithm (CCRIME) based on vertical and horizontal crossover search strategies to improve the quality of the solutions obtained by the RIME algorithm and further enhance its search capabilities. Then, by constructing a binary version of CCRIME, the key parameters of FKNN were optimized using a binary conversion method. Finally, a discrete CCRIME-based BCCRIME was developed, which uses an S-shaped function transformation approach to address the feature selection issue by converting the search result into a real number that can only be zero or one. The performance of CCRIME was examined in this study from various perspectives, utilizing 30 benchmark functions from IEEE CEC2017. Basic algorithm comparison tests and sophisticated variant algorithm comparison experiments were also carried out. In addition, this paper also used collected microseismic and blasting data for classification prediction to verify the ability of the BCCRIME-FKNN model to process real data. This paper provides new ideas and methods for real-time monitoring of rock mass stability during deep well mineral resource mining.

Keywords: RIME; machine learning; swarm intelligence; feature selection; microseismic; blasting

check for updates

Citation: Zhu, W.; Li, Z.; Heidari, A.A.; Wang, S.; Chen, H.; Zhang, Y. An Enhanced RIME Optimizer with Horizontal and Vertical Crossover for Discriminating Microseismic and Blasting Signals in Deep Mines. *Sensors* **2023**, *23*, 8787. https://doi.org/10.3390/s23218787

Academic Editor: Gilbert-Rainer Gillich

Received: 25 July 2023
Revised: 20 October 2023
Accepted: 24 October 2023
Published: 28 October 2023

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

1. Introduction

Many challenges arise as mineral resources are extracted at greater depths, including elevated ground stress, increased well depth, and heightened subsurface temperature, rendering the mining environment increasingly intricate. Engineering disturbances, such as rock drilling and blasting, can influence the stress distribution of deep well rock masses [1]. The rock mass exhibits heterogeneity due to the presence of diverse joints and cracks [2]. When subjected to external pressures, the rock mass experiences a disturbance, resulting in the fracturing of fissures within the rock mass. These fractures subsequently propagate and enlarge until the accumulated energy is released as elastic waves. In instances of significant severity, this phenomenon can result in the destabilization of rock masses, the collapse of regions that have been mined out, the occurrence of roof falls and rock bursts, and several other calamities related to rock masses. Implementing advanced and precise preventative and control methods poses challenges to the safe and effective recovery of deep mineral resources [3]. Hence, the pressing issue at hand pertains to effectively monitoring the real-time stability of rock masses during deep well natural resource mining,

appropriately managing areas of pressure concentration, and guaranteeing the safety of humans and equipment.

The primary purpose of microseismic monitoring equipment is to observe and analyze the vibrational signals that arise from the fracturing of rock masses during excavation activities. The stability of the rock mass is evaluated by evaluating and processing the signals, solving the source parameters, and studying the spatiotemporal evolution law of microseismic events. Microseismic monitoring technology's efficacy is contingent upon the precision of microseismic signal detection [4]. The signals obtained from the existing microseismic monitoring system are frequently contaminated with significant levels of noise, including disturbances caused by blasting and rock drilling activities. This interference significantly hampers the effectiveness of microseismic monitoring technology in analyzing the stability of rock masses and issuing timely warnings for potential disasters. Historically, operators at the site have relied on their personal expertise to categorize microseismic and blasting data, a process that has been both time-consuming and demanding in terms of labor. Moreover, this approach has been prone to subjective interpretation. Consequently, to address this issue, a growing body of scholars has been investigating it.

Although previous studies [5,6] have achieved outstanding results in the field of microseismic and blasting evaluation, most of them were still insufficient in efficiently identifying the two types of signals due to the complex on-site production environment, multiple and mixed noise sources, and large blasting impact. This resulted in a lot of background noise and useless data mixed with adequate microseismic information. In addition, there was a large overlap between the frequency distribution and other source parameters of microseismic and blasting events. In order to address this issue, this paper attempted to develop a model with strong identification capabilities based on fuzzy k-nearest neighbor (FKNN). FKNN is an excellent classification model with no prior data distribution assumptions, no training needs, and simple thinking. As a result, it is widely used and has a wide range of applications. There have also been many related studies in recent years.

Zhao et al. [7] introduced the DPC-FWSN algorithm, which uses fuzzy and weighted shared neighbor methods for clustering uneven density datasets. By employing an improved binary salp swarm algorithm-based FKNN approach (IBSSA-FKNN) to differentiate between patients with mild cognitive impairment, Alzheimer's disease, and healthy controls, Lu et al. [8] created an early diagnostic method for Alzheimer's disease (AD). To diagnose epileptic EEGs, Liu et al. [9] suggested a modified binary grey wolf optimization-based fuzzy KNN method. A GSHHO-FKNN machine learning model was created by Zhang et al. [10] by combining an enhanced Harris hawk's optimization (GSHHO) based on the Gaussian mutation mechanism and a simulated annealing approach with a fuzzy KNN. Using a machine learning technique that combines an enhanced binary mutant quantum grey wolf optimizer (MQGWO) with FKNN, Hu et al. [11] examined 3069 data points from 314 HD patients. Ahmad et al. [9] offered an ensemble classifier composed of FKNN, random forest, k-nearest neighbor, and support vector machine to enhance prediction outcomes further. In order to concurrently execute feature selection and parameter optimization, Wu et al.'s [12] enhanced sine cosine algorithm-based optimization approach was used to create an FKNN model. In order to generate the membership matrix, Wan et al. [13] used FKNN to combine the intraclass and interclass weighted matrices.

However, two important parameters affect FKNN's classification accuracy and are also extremely important for feature extraction of microseismic and blasting images in deep areas. Therefore, optimizing FKNN's parameters is key to establishing strong identification capabilities. Parameter optimization methods have unique advantages in finding the best parameters of a classifier or regressor, or solutions for a certain goal selected from many sub-optimal solutions [14–16]. By monitoring and mimicking the behavioral traits of natural occurrences like biological foraging and nesting, swarm intelligence optimization approaches have offered several innovative and effective optimization strategies. They provide fresh approaches and methods for resolving several complicated issues and have

high parallelism and autonomous exploration capabilities. They exhibit remarkable performance across various domains such as feature selection [17,18], image segmentation [19,20], economic scheduling [21,22], multi-objective problems [23,24], expensive optimization problems [25,26], multi-attribute decision making [27–30], and disease diagnosis [31,32].

Numerous swarm intelligence optimization techniques have been presented in recent years, such as the RIME [33], multi-verse optimizer (MVO) [34], bat-inspired algorithm (BA) [35], Harris hawks optimization (HHO) [36], particle swarm optimization (PSO) [37], salp swarm algorithm (SSA) [38], whale optimization algorithm (WOA) [39], weighted mean of vectors (INFO) [40], Runge Kutta optimizer (RUN) [41], colony predation algorithm (CPA) [42], stochastic fractal search (SFS) [43], slime mold algorithm (SMA) [44,45], and hunger games search (HGS) [46]. In addition, in order to further improve the performance of swarm intelligence optimization algorithms, many variant algorithms have been proposed, such as the bat algorithm based on collaborative and dynamic learning (CDLOBA) [47], the ant colony optimizer with grade-based search (GACO) [48], hybridizing grey wolf optimization with differential evolution (HGWO) [49], evolutionary biogeography-based whale optimization (EWOA) [50], the chaotic mutative moth-flame-inspired optimizer (CLS-GMFO) [51], the moth-flame optimizer with mutation strategy (LGCMFO) [52], Cauchy and Gaussian sine cosine optimization (CGSCA) [53], and the double adaptive random spare reinforced whale optimization algorithm (RDWOA) [54], the A–C parametric whale optimization algorithm (ACWOA) [55], Harris hawks optimization with Gaussian mutation (GCHHO) [56], an enhanced sine cosine algorithm (LSCA) [57], and the Gaussian kernel probability-driven slime mold algorithm with new movement mechanism (MGSMA) [58]. Among them, RIME is a recently proposed optimization algorithm with strong optimization performance [33]. Therefore, this paper proposes a RIME based on vertical and horizontal cross-cross search through the study of RIME to enhance the performance of FKNN and improve the identification ability of microseismic and blasting.

The main contributions and innovations of this paper are as follows:

- Through examining and analyzing the RIME algorithm, this paper proposes a novel method called CCRIME, which is based on vertical and horizontal crossover. The introduction of CCRIME not only enhances the quality of the solutions found but also improves the overall search capabilities.
- To enhance the classification capabilities of the FKNN model, a binary version of CCRIME was created through the use of the binary transformation method. This approach aimed to optimize the important parameters inside the FKNN model. The CCRIME-FKNN model, which is optimized for CCRIME, is an abbreviation for the CCRIME-optimized FKNN model.
- The performance of CCRIME, an optimization method based on swarm intelligence, was evaluated using 30 benchmark functions from IEEE CEC2017. The results of this study demonstrated that CCRIME exhibits exceptional performance across several perspectives, establishing it as a highly effective algorithm.
- This study utilized microseismic and blasting images to extract and select appropriate features. Through the application of CCRIME-FKNN, the identification of microseismic and blasting events was successfully achieved, resulting in a high level of accuracy.

The remaining content of this paper is arranged as follows: Section 2 describes the collected microseismic and blasting data. Section 3 elaborates on the details of the proposed CCRIME. Section 4 explains how to use CCRIME to optimize FKNN. Section 5 analyzes the performance of CCRIME using benchmark functions and identifies microseismic and blasting signals using CCRIME-FKNN. Section 6 summarizes the research work carried out in this paper and points out future research directions.

2. Related Work

2.1. Data Description

Linglong Gold Mine is one of the gold mines with the longest mining history and the deepest mining depth in China. The surface elevation of the Daikoutou mining section in the Jiuqu mining area is +255, and the current mining depth has reached below 900 m underground. Due to the simple, early mining methods and the failure to fill the mined-out areas, a large number of mined-out areas have been left under the ground. Within an area affected by a mined-out area, secondary stress is highly concentrated, and the surrounding rock of the ore body is granite with a strong tendency toward rock burst. When carrying out tunnel excavation and other production activities in this area, blasting disturbance causes stress redistribution, quickly forming stress concentration areas and thus triggering geological disasters such as rock bursts. Significant rock burst phenomena have occurred during the development of the −620 m, −670 m, and −694 m levels, causing injuries to underground personnel and loss of mine property.

As the mining depth continues to increase, the ground stress also increases, and the risk of geological disasters such as rock bursts gradually increases. To this end, Linglong Gold Mine has carried out scientific research in cooperation with Central South University to build an online microseismic monitoring system for ground pressure in the Daikoutou mining area for real-time and long-term monitoring of deep areas. The waveform information collected by the IMS microseismic monitoring system automatically picks up the arrival time of P-waves and S-waves through its own software, TRACE. The software lists the number, code, and order of sensors triggered by microseismic monitoring signals, displays the moment when signals trigger the sensors, and collects waveform images from each sensor, as shown in Figure 1.

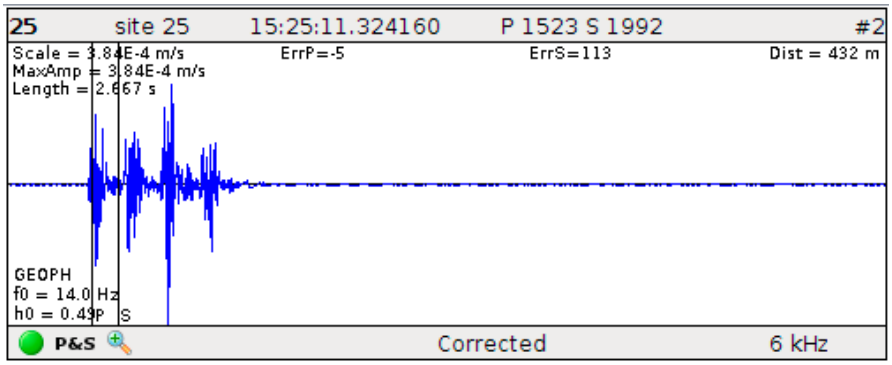


Figure 1. Monitoring and control interface of the Linglong Gold Mine microseismic system software.

The monitoring technician screens the blasts and microseismic events by comparing the source trigger time with the source waveforms. Accurate source localization is ensured by calibrating the P-wave arrival time based on the waveform onset time. Due to the difference in the generation mechanisms of the blasts and microseismic events, the two waveforms have their own characteristics. For example, the energy of the blast waveform is concentrated in the front end of the rapid decay without an obvious S-wave, and the energy of the microseismic events waveforms is evenly distributed in the window with an obvious S-wave. The amplitudes of the blast waveforms are generally larger than that of the microseismic event waveforms. The microseismic monitoring database of the Linglong Gold Mine was constructed from the artificially screened blasts and microseismic events waveforms to analyze the microseismic activities and evaluate the engineering health of the rock mass. Therefore, the original data source for this study is the waveform images collected in the Linglong Gold Mine microseismic monitoring system database.

2.2. Microseismicity and Blasting

Since microseismicity and blasting have different characteristics and effects, understanding the differences between them can help us better understand and respond to earthquakes and blasting activities. For example, recognizing microseismic activity can better predict possible large earthquakes and facilitate appropriate preventive measures. As for blasting, we can minimize the negative impact on the environment by understanding its characteristics and impacts and taking the necessary measures. Computer-aided technology can improve the accuracy of identifying microseisms and blasting by automating the analysis of large amounts of data. Compared with traditional manual identification methods, computer-aided technology can eliminate the influence of human factors and reduce the possibility of misjudgment and omission.

For example, Jiang et al. [59] suggested a time-frequency discrimination approach to automatically detect these signals and produce more trustworthy data for mass spectrometry monitoring analysis. Singular value decomposition is used to decrease the amount of data while obtaining distinctive characteristics. Random Forest (RF) is employed to accomplish automated identification, and the essential characteristics are determined based on changes in the classification error and Gini coefficient. Dai et al. [60] developed a technique for automatically recognizing MS signals in the presence of blasting signal interference. Improved adaptive noise with full integrated empirical mode decomposition, singular value decomposition, and the k-nearest neighbor algorithm are all part of this technique. The original multi-frequency signal is decomposed into many single-frequency signal subcomponents using integrated empirical mode decomposition, and singular values are extracted from the matrix generated by the decomposition results using SVD.

In order to distinguish between coal mine microseismic signals and blasting signals, Li et al. [61] studied the differences between them by analyzing their main frequency, signal duration, waveform after peak attenuation, and multiple fractal parameters, laying a foundation for automatic identification of mine-used MS signals and blasting signals recorded underground in coal mines. Li et al. [62] analyzed the fractal characteristics of microseismic waves and blasting waves in a certain mine based on the nonlinear characteristics of waveforms using simple fractal and multiple fractal theory. This method can more clearly and significantly identify mine MS and blasting waves, thereby monitoring coal rock fracture more accurately. Peng et al. [63] proposed an automatic classification method for microseismic signals based on the Gaussian mixture model-hidden Markov model. This method only uses the signal's extracted Mel frequency cepstral coefficient (MFCC) features for training and testing. The results show that the classification accuracy of this method reaches 92.46%, verifying its effectiveness in the automatic classification of underground microseismic data.

3. The Proposed CCRIME

3.1. An Overview of RIME

RIME is a new and efficient optimization algorithm proposed by Su et al. [33]. It mainly simulates the peculiar physical phenomenon of uncondensed water vapor in the air condensing on branches or objects and freezing when encountering low temperatures. The RIME algorithm consists of 4 main stages: the initial rime ice crystal cluster formation, soft RIME search strategy, hard RIME puncture mechanism, and greedy selection mechanism for the optimal solution.

In the initial stage of rime formation, each search agent in the population is treated as a RIME agent of the RIME algorithm. All RIME agents are combined into a whole as the initial population of the algorithm. First, each agent in the population is initialized within a given search space through Equation (1):

$$R_i = B_{min} + r_i \times (B_{max} - B_{min}) \quad (1)$$

where R_i represents each search agent, B_{max} and B_{min} represent the upper and lower boundaries, respectively, and r_i is a pseudo-random sequence between [0, 1].

The soft RIME search strategy mainly simulates that when free particles move near the soft rime, they will be captured and condensed by the particles in the soft rime and change the stability of the soft rime. When the distance between the particles and the soft rime particles is far, they will not be captured and continue to be in a free state. As free particles are gradually captured and condensed by the soft rime, the coverage area of the soft rime will gradually increase, and its probability of capturing particles will also increase. However, due to environmental factors, the coverage area of the soft rime will not increase indefinitely and will gradually show a steady state. By simulating the movement rules of free particles and soft rime, the specific mathematical model is as follows:

$$R_{ij}^{new} = R_{best,j} + Rf \times \left(h \times \left(B_{\max(i,j)} - B_{\min(i,j)}\right) + B_{\min(i,j)}\right), r_2 < E \tag{2}$$

$$Rf = r_1 \times \cos\theta \times \beta \tag{3}$$

$$\theta = \pi \cdot \frac{t}{10 \cdot T} \tag{4}$$

$$\beta = 1 - \left[\frac{w \cdot t}{T}\right] / w \tag{5}$$

$$E = \sqrt{(t/T)} \tag{6}$$

where R_{ij}^{new} is the position of the free particle after moving and $R_{best,j}$ is the best RIME agent in the RIME population. r_1 is a random number between $[-1, 1]$. $\cos\theta$ will change as the algorithm iterates. β represents the environmental factors in rime and changes as the algorithm iterates, controlling the algorithm's convergence. h is a random value between [0, 1], simulating the distance between free particles. t and T represent the current number of iterations of the algorithm and the upper limit of the number of iterations, respectively. w controls the number of segments of the step function, with a default value of 5. E represents the probability of free particles being captured. It will change as the algorithm iterates and generally shows an increasing trend.

The hard RIME puncture mechanism mainly simulates that in a stronger wind environment, the movement of particles will be affected by the wind direction, and due to the same growth direction, it is easy for a cross phenomenon called hard rime puncture to occur between each hard rime. In addition, as the hard rime grows, the phenomenon of hard rime puncture will become more frequent. Therefore, the hard rime puncture mechanism can be used for position updates between rime agents, allowing rime agents to exchange particles and improve the convergence ability of the algorithm to help jump out of local optima. The mathematical model of the hard RIME puncture strategy is as follows:

$$R_{ij}^{new} = R_{best,j}, r_3 < F^{normr}(S_i) \tag{7}$$

where R_{ij}^{new} is the *jth* particle of the best rime agent in the population and $R_{best,j}$ new is the updated particle location. The fitness value of the current agent is normalized to produce $F^{normr}(S_i)$, which represents the likelihood that the agent will experience a hard rime puncture. A number at random between [0, 1] makes up r_3.

3.2. Horizontal and Vertical Crossover Search

In order to perform the horizontal crossing operation, two different rime agents are generally used. The ability of distinct agents to communicate with and learn from one another as a consequence of the horizontal crossover search significantly increases the agents' capacity for exploration and accelerates algorithm convergence. Assume that

the parent agents x_i and x_j experience a horizontal crossover operation, which may be described by Equations (8) and (9):

$$MS_i^n = \varepsilon_1 \times x_{in} + (1 - \varepsilon_1) \times x_{jn} + c_1 \times \left(x_{in} - x_{jn}\right) \tag{8}$$

$$MS_j^n = \varepsilon_2 \times x_{jn} + (1 - \varepsilon_2) \times x_{in} + c_2 \times \left(x_{jn} - x_{in}\right) \tag{9}$$

where the given equation involves the use of random variables ε_1 and ε_2, which are uniformly distributed within the interval [0, 1]. Additionally, the variables c_1 and c_2 are also uniformly distributed within the interval $[-1, 1]$. The variables x_{in} and x_{jn} represent the nth dimension of the *ith* and *jth* agents, respectively. The offspring of the *nth* position vectors, denoted by MS_i^n and MS_j^n, are generated through the horizontal crisscross operation of the agents x_i and x_j.

The vertical crossover operator is a computational procedure that involves the manipulation of two discrete position vectors belonging to individual agents within a given population. This approach potentially enables certain position vectors that are confined to a local optimum to persist in their search while simultaneously minimizing alterations to the position vectors that are typically explored. Consequently, in the latter phases of the exploration process, ants frequently encounter a local optimum due to the stagnation of specific position vectors. Implementing a vertical crisscross search technique can facilitate inter-agent learning of position vectors, thereby enhancing their capacity to evade a local optimum. Assuming the vertical crossover operation is performed on the *mth* and *nth* position vectors of agent *i*, as described in Equation (10):

$$MS_i^m = \varepsilon \times x_{im} + (1 - \varepsilon) \times x_{in} \tag{10}$$

where ε represents a random number between [0, 1] and MS_i^m represents the *mth* position vector of the offspring generated by the vertical crisscross operation between the *mth* agent *nth* position vectors of agent *i*.

3.3. The Proposed CCRIME

Since the original research on RIME was only an optimization algorithm proposed by simulating the changing shape of rime, no further research or analysis was conducted on the algorithm's performance. Further, in this paper, we have found that it has a greater advantage in convergence through benchmark function experiments, and it has further room for improvement. In addition, when solving the feature selection problem, through experiments, we also have found that it has a greater advantage than the general traditional algorithms in finding high-quality individuals for solving practical problems. Therefore, in order to further study the performance of the RIME algorithm and improve its performance, this paper compared RIME with some similar algorithms from multiple angles and found that there is still room for improvement in the quality of the solutions obtained by RIME and its search capabilities need to be further enhanced to avoid falling into local optima.

Based on these facts, this paper found through the study of vertical and horizontal cross-search strategies that it has a high ability to improve the performance of algorithms, and some studies have also used this method to enhance the performance of some optimization algorithms [40–42]. Inspired by this, in order to improve RIME's search capabilities, enhance the quality of the solutions obtained, and avoid falling into local optima during the search process, this paper enhanced the original RIME by using horizontal and vertical crossover search strategies. In the enhancement process, the horizontal and vertical crossover search strategies mainly act after the hard RIME puncture mechanism. By improving the search capabilities of RIME, CCRIME can obtain higher-quality solutions and effectively improve its optimization capabilities in practical applications. The pseudocode is shown in Algorithm 1.

Algorithm 1 Pseudocode of CCRIME

Initialize the population of rime R
Calculate the fitness of each agent
Select the optimal agent
While $t < T$
  Update the particle capture probability E
  If $r_2 < E$
    Update rime agent location by Equation (10)
  End If
  If $r_3 < F^{normr}(S_i)$
    Cross-updating between agents by Equation (14)
  End If
  If $f\left(R_i^{new}\right) < f(R_i)$
    Replace R_i with R_i^{new}
    If $f\left(R_i^{new}\right) < f(R_{best})$
      Replace R_{best} with R_i^{new}
    End If
  End If
  Perform horizontal crossover search and vertical crossover search
  $t = t + 1$
  End While

4. The Proposed CCRIME-FKNN Model

4.1. Binary Transformation Method

The goal of feature selection is to select sample features with obvious distinctiveness, and feature selection based on swarm intelligence algorithms selects the main attributes representing the dataset. Suppose the given feature set is F, represented as $F_i = \left\{f_{i,1}, f_{i,2}, \cdots, f_{i,j}, \cdots, f_{i,t}\right\}$, where t represents the number of features after preprocessing and i represents the original number of datasets. FS is the new feature subset selected by the feature selection algorithm, represented as $FS_i = \left\{fs_{i,1}, fs_{i,2}, fs_{i,j}, \cdots, fs_{i,m}\right\}$, where m is the length of the selected features and $fs_{i,j} \in \{1, 0\}, j = 1, 2, \cdots, m$. If $fs_{i,j} = 1$, this indicates that the selected feature j in the dataset is a high-quality feature. If $fs_{i,j} = 0$, this indicates that the selected feature j in the dataset is useless. In the process of using a binary RIME algorithm to select features, each position of a rime individual represents a subset of a feature dataset. The rime population includes a set of positions of rime individuals, where the position set is represented by binary 0 or 1. The binary representation of the solution space is shown in Equation (11):

$$X_d(t+1) = \begin{cases} 1, & sigmoid\left(X_d(t)\right) \geq rand \\ 0, & otherwise \end{cases} \tag{11}$$

The variable *rand* denotes a uniformly distributed random number within the interval [0, 1]. The notation $X_d(t+1)$ refers to the updated binary solution at time $t+1$. The mathematical expression for the Sigmoid function is presented in Equation (12) herein, where the variable x represents the solution produced during the CCRIME iteration.

$$sigmoid(x) = \frac{1}{1 + e^{-2x}} \tag{12}$$

4.2. Feature Extraction Method

The process of obtaining a "non-image" representation or description of an image, such as numerical values, vectors, and symbols, in order to derive useful data or information, is referred to as feature extraction. The resulting features are the extracted "non-image" representations or descriptions. The present study employed color moments, gray-level co-occurrence matrices, and Local Binary Patterns (LBP) as feature extraction techniques. Subsequently, we derived distinctive attributes from the intrinsic shape properties of

microseismic and blasting images. These attributes encompass the area, bounding box, major axis, minor axis, eccentricity, orientation, Euler number, equivalent circle diameter, fullness, expansion degree, and perimeter.

The mathematical theory of color matrixes is that all colors in an image can be described using matrix methods. At the same time, because the color distribution of the image is concentrated in low-order matrixes, the distribution of image colors can be described by first-order and second-order colors. This is most effective when it only contains a single target picture. The retrieval results are very effective when using three color coordinate axes, each including color moments axes. In this paper, we used color moments to extract 3 grayscale color features from the microseismic and blasting images, 12 RGB (RGB stands for red, green, and blue channels) color features, and 12 HSI (perceived colors based on three basic characteristics: hue, saturation, and brightness) color features.

As per previous works, the gray-level co-occurrence matrix stands as one of the well-established approaches to statistical texture feature extraction from images. This method mainly obtains features such as energy and inertia of the image gray-level co-occurrence matrix. In terms of statistics, one commonly used method is to use the autocorrelation function of the image to extract texture features of the image. By processing the energy spectrum function of the image, a good extraction of related texture parameters can be achieved. In this paper, for texture features, we extracted 20 gray-level co-occurrence matrix texture features, 6 gray-level co-occurrence smooth matrix texture features, and 15 gray-level co-occurrence gradient matrix texture features.

The LBP operator recognizes the central pixel based on the combined difference distribution between adjacent pixels and the central one. Essentially, it calculates the first derivative of the binary gradient direction between neighboring pixels on a circle and center pixels. Therefore, it also calculates the gradient relationship in local neighborhoods of images. The encoding outcome from the LBP operator reveals certain discernible local attributes in the original images, such as boundaries, corners, flat areas, and feature points. Additionally, histogram statistics can establish a somewhat unique vector to express the image features.

In this paper, for local texture features, we used the basic LBP equivalent LBP rotation invariant to extract 36 features.

4.3. Fuzzy k-Nearest Neighbor

The k-nearest neighbor (KNN) algorithm functions as a classification method influenced by neighboring elements. This straightforward, efficient, and non-parametric form of supervised learning hinges on providing a training set equipped with class labels or numerical attributes. For a sample to be classified or predicted, find k nearest training samples to it and then determine its output based on the neighbors' information. The output is the most common category among neighbors.

The incorporation of fuzzy logic into the KNN algorithm has been proposed to enhance the flexibility and interpretability of the model in the context of classification tasks. This has led to the development of FKNN. In contrast to a solitary KNN class, the process of conducting FKNN classification involves determining the appropriate values for the number of neighbors (k) and the fuzzy strength coefficient (m), as well as selecting an optimization strategy to enhance the k and m values. These factors are crucial in influencing the outcomes of the classification. Consequently, various optimization algorithms can be employed in experimental settings to optimize diverse k and m values and attain the optimal solution for k and m. The formula for determining the degree of membership of data in a fuzzy set is expressed as follows:

$$u_i(x) = \frac{\sum_{j=1}^{k} u_{ij}\left(1/||x - x_j||^{2/(m-1)}\right)}{\sum_{j=1}^{k} \left(1/||x - x_j||^{2/(m-1)}\right)} \tag{13}$$

where $u_i(x)$ represents the specific membership of vector x. $i \in [1:C]$ represents the number of classes. $j \in [1:k]$ represents the number of nearest neighbors. In addition, the fuzzy strength m parameter is used to determine the distance weight. m represents the degree of distance weighting when calculating the contribution of each neighbor to the membership value.

4.4. The Proposed CCRIME-FKNN Model

Feature selection technology is a binary optimization approach, whereas the CCRIME algorithm is specifically developed to address real continuous optimization problems. Consequently, it is not feasible to employ the CCRIME algorithm for directly resolving feature selection issues. During the optimization process, executing a binary operator is necessary to convert real values to binary values. The present study introduces a discrete iteration of CCRIME, namely BCCRIME, which was derived from the continuous CCRIME. The BCCRIME methodology establishes a bounded search range for the solution, specifically between 0 and 1. This approach treats the feature selection problem as a constrained optimization problem. Subsequently, the outcome of the search process is subjected to a Sigmoid function transformation technique, resulting in a binary numerical representation consisting solely of 0 and 1. In this representation, a value of 1 indicates that the model has selected the corresponding feature, while a value of 0 indicates that it has not been selected. A comprehensive flowchart is presented herein (Figure 2).

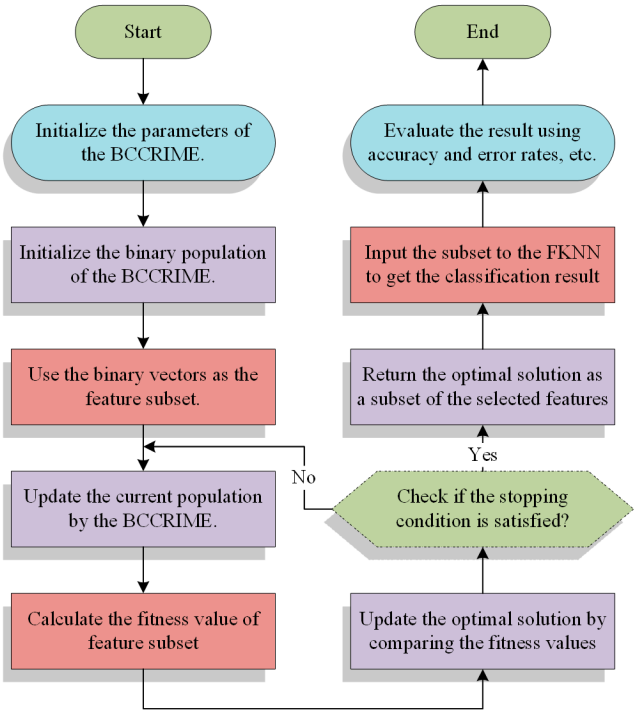


Figure 2. The flowchart of the developed CCRIME-FKNN model.

The evaluation function of the combined use of BCCRIME and the FKNN classifier is deemed overly simplistic and lacks universality. When utilizing the CCRIME algorithm for feature selection, it becomes imperative to reevaluate the approach to establishing the evaluation function. As a preliminary step in data analysis, feature selection necessitates two primary prerequisites. The feature subset acquired exhibits a notable degree of classification accuracy, indicating a robust correlation between the feature subset and the respective category. It is advisable to minimize the number of features in the selected feature subset to mitigate the effects of the curse of dimensionality. Hence, the present study redefined an

evaluation function that is appropriate for the BCCRIME-FKNN hybrid model, as depicted in Equation (14):

$$Fitness = \alpha \cdot \text{ error } + \beta \cdot \frac{|R|}{|D|} \tag{14}$$

where the formula provided denotes the relationship between the error rate of a classifier model, the dimensionality of a dataset, and the number of attributes in the resulting subset. Specifically, the variable *error* represents the error rate of the classifier model, while the variable $|D|$ represents the dimensionality of the dataset, which refers to the total number of attributes. Lastly, the variable $|R|$ represents the number of attributes in the resulting subset. The variable α is utilized as a metric for assessing the significance of the classification error rate, whereas β denotes the extent of the chosen characteristics. Consistent with prior literature, the present study employs $\alpha = 0.99$ and $\beta = 0.01$.

5. Experiments, Results, and Analysis

This section primarily focused on two aspects and utilized experimental validation to implement the proposed methodology. Initially, a set of benchmark function experiments were carried out to authenticate the efficacy of CCRIME. Subsequently, the CCRIME-FKNN approach was utilized in various feature selection-related classification prediction tasks, successfully showcasing its robust classification prediction abilities. The CCRIME-FKNN method was effectively utilized in the investigation of deep mine microseismic and blasting identification, resulting in outstanding outcomes.

5.1. Benchmark Function Validation

In this section, based on the IEEE CEC2017 benchmark function experiment, not only were CCRIME and 11 basic algorithms compared experimentally, but CCRIME was also compared experimentally with 11 advanced variant algorithms, fully demonstrating the core advantages of CCRIME's strong convergence performance and resistance to falling into local optima.

5.1.1. Experimental Setup

Benchmark datasets are a commonly used tool for evaluating various computational characteristics, using a set of recognized criteria to determine which technology performs best overall [64,65]. The experimental foundation for the benchmark function experiment consisted of 30 benchmark functions from IEEE CEC2017, as presented in Table 1. Throughout the experiment, the primary focus was on comparing basic and advanced variant algorithms. The experiment on algorithm comparison involved several algorithms, namely RIME [33], MVO [34], BA [35], HHO [36], PSO [37], SSA [38], WOA [39], JAYA [66], PO [67], SFS [43], SMA [45], and HGS [46]. The superior performance of the subject has been adequately demonstrated in prior original research, thereby rendering the comparative outcomes with said research compelling. As RIME is categorized as a variant algorithm, it is imperative to conduct a comparative analysis with other advanced variant algorithms such as CDLOBA [47], GACO [48], HGWO [49], EWOA [50], CLSGMFO [51], LGCMFO [52], CGSCA [53], RDWOA [54], ACWOA [55], GCHHO [56], LSCA [57], and MGSMA [58]. The variant algorithms that are involved in the participation not only exhibit commendable performance but have also demonstrated successful application across various domains. In order to guarantee equity and precision in the experimental procedure, all algorithms utilized in the comparison were executed under identical circumstances. This study established a population size of 30 and uniformly designated a maximum evaluation number of 300,000 iterations. Furthermore, the algorithms underwent 30 independent tests in order to mitigate the influence of stochastic variables. Comprehensive statistical analyses were conducted on the experimental outcomes of the benchmark functions, utilizing measures such as the mean, variance, Wilcoxon signed-rank test, and Friedman test.

Table 1. The description of 30 benchmark functions.

Item	The Function Class	The Function Name	Search Space	The Optimal Fitness
F1	Unimodal Functions	Shifted and Rotated Bent Cigar Function	[−100, 100]	100
F2		Shifted and Rotated Sum of Different Power Function	[−100, 100]	200
F3		Shifted and Rotated Zakharov Function	[−100, 100]	300
F4	Multimodal Functions	Shifted and Rotated Rosenbrocks Function	[−100, 100]	400
F5		Shifted and Rotated Rastrigins Function	[−100, 100]	500
F6		Shifted and Rotated Expanded Scaffers F6 Function	[−100, 100]	600
F7		Shifted and Rotated Lunacek Bi_Rastrigin Function	[−100, 100]	700
F8		Shifted and Rotated Non-Continuous Rastrigins Function	[−100, 100]	800
F9		Shifted and Rotated Levy Function	[−100, 100]	900
F10		Shifted and Rotated Schwefels Function	[−100, 100]	1000
F11	Hybrid Functions	Hybrid Function 1 (N = 3)	[−100, 100]	1100
F12		Hybrid Function 2 (N = 3)	[−100, 100]	1200
F13		Hybrid Function 3 (N = 3)	[−100, 100]	1300
F14		Hybrid Function 4 (N = 4)	[−100, 100]	1400
F15		Hybrid Function 5 (N = 4)	[−100, 100]	1500
F16		Hybrid Function 6 (N = 4)	[−100, 100]	1600
F17		Hybrid Function 6 (N = 5)	[−100, 100]	1700
F18		Hybrid Function 6 (N = 5)	[−100, 100]	1800
F19		Hybrid Function 6 (N = 5)	[−100, 100]	1900
F20		Hybrid Function 6 (N = 6)	[−100, 100]	2000
F21	Composition Functions	Composition Function 1 (N = 3)	[−100, 100]	2100
F22		Composition Function 2 (N = 3)	[−100, 100]	2200
F23		Composition Function 3 (N = 4)	[−100, 100]	2300
F24		Composition Function 4 (N = 4)	[−100, 100]	2400
F25		Composition Function 5 (N = 5)	[−100, 100]	2500
F26		Composition Function 6 (N = 5)	[−100, 100]	2600
F27		Composition Function 7 (N = 6)	[−100, 100]	2700
F28		Composition Function 8 (N = 6)	[−100, 100]	2800
F29		Composition Function 9 (N = 3)	[−100, 100]	2900
F30		Composition Function 10 (N = 3)	[−100, 100]	3000

5.1.2. Comparison with Basic Algorithms

This section presents a comparison between CCRIME and 11 fundamental algorithms in IEEE CEC2017. This study conducted a comparison of various algorithms, namely RIME [33], MVO [34], BA [35], HHO [36], PSO [37], SSA [38], WOA [39], JAYA [66], PO [67], SFS [43], SMA [45], and HGS [46]. Table A1 displays the statistical measures of the central tendency and variability, specifically the mean and standard deviation, respectively. The "AVG" column denotes the arithmetic mean, while the "STD" column represents the standard deviation. These values were obtained as a result of the conducted experiment. Upon examination of the mean and standard deviation, it has been determined that CCRIME achieved the lowest mean on 21 functions, while SSA achieved the lowest mean on 3 functions. Additionally, RIME, BA, and PSO each achieved the lowest mean on 2 functions. An analysis of the mean revealed that CCRIME exhibited superior performance on over two-thirds of the functions, thus implying its robust optimization capabilities and ability to yield solutions of high quality. Furthermore, it is evident that CCRIME exhibited favorable performance with respect to the standard deviation, implying its commendable stability.

Moreover, Table 2 presents the performance hierarchy of all algorithms across individual benchmark functions, as well as the comprehensive ranking across 30 benchmark functions. Through empirical observation, it is evident that CCRIME achieved the highest ranking in all categories of functions within IEEE CEC2017, as well as securing the top position in the overall ranking across 30 functions. This serves as a comprehensive demonstration of the exceptional performance of CCRIME. Furthermore, for the purpose of enhancing the dependability of the experimental outcomes, Table 2 furnishes the analysis outcomes of the Wilcoxon signed-rank test. The symbols "+" and "−" are employed to denote that CCRIME outperformed and underperformed the comparison algorithm, respectively, while "=" signifies that CCRIME and the comparison algorithm exhibited comparable performance. The analysis results indicated that CCRIME outperformed other algorithms on 21 out of 30 functions, thereby providing substantial evidence to support the superior performance of CCRIME.

Table 2. The performance ranking of all algorithms on each benchmark function.

	F1	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**	**F11**
CCRIME	2	4	6	4	1	1	1	1	1	3	1
RIME	5	7	5	8	4	2	2	2	2	1	2
MVO	6	6	4	7	3	5	5	4	3	5	7
BA	7	1	3	2	13	13	13	13	13	9	10
HHO	9	8	9	10	9	11	11	9	11	8	5
PSO	11	9	7	1	10	10	8	10	10	11	8
SSA	1	3	1	6	7	9	6	5	5	7	6
WOA	8	10	13	11	12	12	12	11	12	12	12
JAYA	13	12	11	13	11	7	10	12	6	13	13
PO	10	13	12	9	2	6	3	6	7	6	9
SFS	12	11	10	12	8	8	9	8	8	10	11
SMA	4	2	2	5	5	3	4	3	4	4	4
HGS	3	5	8	3	6	4	7	7	9	2	3
	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22
CCRIME	1	2	6	1	1	1	1	1	1	1	1
RIME	6	1	5	2	2	4	7	3	4	5	4
MVO	7	5	3	3	5	5	3	4	2	4	8
BA	4	8	2	10	12	13	4	9	13	13	13
HHO	8	9	8	7	9	11	10	7	9	10	11
PSO	10	11	4	12	8	8	5	11	10	11	9
SSA	5	6	1	8	4	2	2	8	3	6	6
WOA	11	7	13	9	13	10	12	12	12	12	12
JAYA	12	12	11	13	11	9	11	10	8	9	3
PO	13	13	12	11	10	12	13	13	11	2	5
SFS	9	10	10	5	6	3	9	5	5	8	2
SMA	3	4	7	6	3	6	8	6	6	3	10
HGS	2	3	9	4	7	7	6	2	7	7	7
	F23	F24	F25	F26	F27	F28	F29	F30	+/−/=	Mean	Rank
CCRIME	1	1	1	1	4	2	1	1	N/A	1.8	1
RIME	3	5	5	4	5	6	2	3	21/2/7	3.87	2
MVO	2	2	3	2	2	4	4	6	22/2/6	4.3	3
BA	13	12	9	13	13	1	13	8	25/4/1	9.33	9
HHO	12	13	10	11	11	9	9	9	29/0/1	9.43	11
PSO	11	10	8	7	1	8	8	10	25/1/4	8.57	8
SSA	5	3	7	5	7	3	6	7	23/3/4	5	5
WOA	10	9	11	12	9	10	12	11	30/0/0	11.07	13
JAYA	9	8	13	10	10	12	10	12	30/0/0	10.47	12
PO	8	11	6	9	12	13	11	13	27/0/3	9.37	10
SFS	7	7	12	8	8	11	7	5	30/0/0	8.13	7
SMA	4	4	2	3	3	7	3	2	23/2/5	4.33	4
HGS	6	6	4	6	6	5	5	4	24/0/6	5.33	6

Following the Wilcoxon signed-rank test analysis, the ranking results subsequent to the Friedman test analysis are presented in Figure 3. These results indicate that CCRIME holds the top rank with a score of 2.55, followed by RIME at No. 2 with a score of 4.22, and MVO at No.3 with a score of 4.65. Consequently, upon conducting an analysis, it can be

deduced that CCRIME exhibited a noteworthy edge over RIME, which is ranked second, and MVO, which is ranked third. This observation further implies that CCRIME holds a superior position in comparison to other algorithms. Figure 4 illustrates the convergence behavior of various functions to showcase the efficacy of CCRIME. The results indicate that CCRIME outperformed other fundamental algorithms in terms of convergence, and it exhibited robust search capabilities.

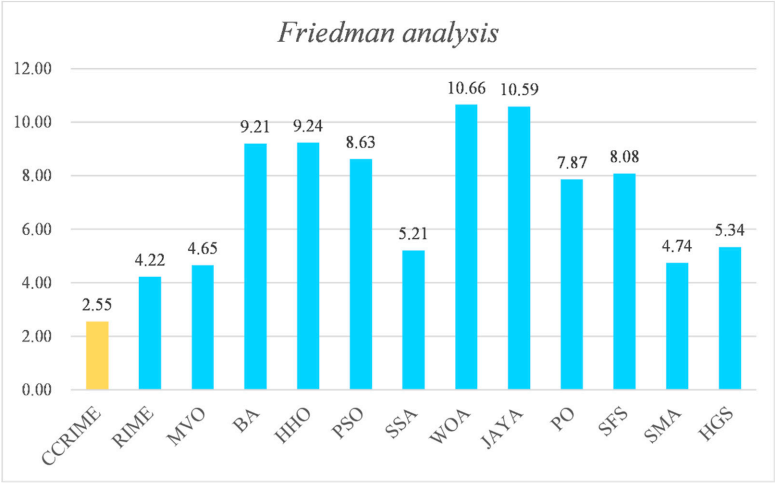


Figure 3. The ranking results after the Friedman test analysis.

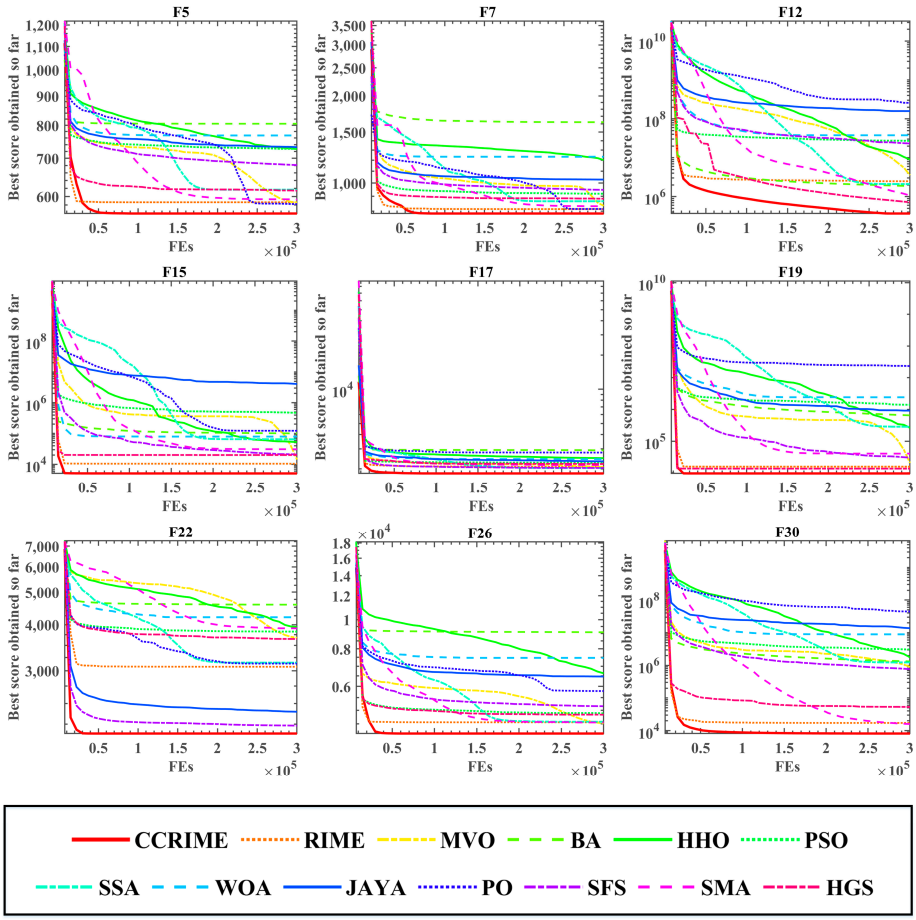


Figure 4. The convergence curves of some functions.

Consequently, the experimental analysis presented above provides evidence that CCRIME exhibited strong convergence performance and a robust ability to avoid local optima in the basic algorithm comparison experiment.

5.1.3. Comparison with State-of-the-Art Variants

This section presents a comparative analysis between CCRIME and 11 advanced variant algorithms, whereby CCRIME is a derivative algorithm that incorporates the horizontal and vertical crossover strategy into RIME. Table A2 presents the mean and standard deviation values for each function. The results indicate that CCRIME achieved the highest mean value across 17 functions, while CDLOBA and LSCA obtained the highest mean value across 3 functions. Additionally, GCHHO, LGCMFO, and MGSMA demonstrated strong performance on 2 functions, while GACO performed well on 1 function. An analysis of the average performance indicates that CCRIME exhibited a distinct superiority in comparison to alternative algorithms. Furthermore, the performance of CCRIME in relation to sexually transmitted diseases is noteworthy, indicating a level of stability.

Table 3 presents the ranking of all algorithms on each of the 30 benchmark functions, with the aim of conducting a more in-depth analysis of CCRIME's performance. The results demonstrate CCRIME's clear advantage over other comparable algorithms, both in terms of its performance on individual functions and its overall performance. The findings of the Wilcoxon signed-rank test confirm the superior performance of CCRIME. Specifically, CCRIME outperformed other comparable algorithms on 20 benchmark functions.

Table 3. The ranking of all algorithms on each function.

	F1	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**	**F11**
CCRIME	2	1	2	2	1	1	1	1	1	1	1
CDLOBA	4	8	4	1	13	13	13	13	13	9	10
GACO	8	4	5	3	4	2	5	2	2	11	4
HGWO	12	12	13	11	10	9	9	10	6	12	13
EWOA	3	7	6	4	7	8	7	8	10	5	5
CLSGMFO	6	6	7	7	6	7	6	6	7	6	6
LGCMFO	7	5	9	5	5	5	4	5	5	4	7
CGSCA	13	13	11	13	11	11	11	12	11	13	11
RDWOA	9	9	10	9	8	6	8	9	9	7	8
ACWOA	11	11	12	12	12	12	12	11	12	10	12
GCHHO	1	3	3	8	9	10	10	7	8	8	9
LSCA	10	10	8	10	2	4	3	3	4	3	3
MGSMA	5	2	1	6	3	3	2	4	3	2	2
	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22
CCRIME	1	4	3	3	2	1	2	1	1	1	1
CDLOBA	3	8	1	10	10	13	1	9	13	13	13
GACO	2	5	6	8	4	3	8	8	2	4	12
HGWO	11	12	13	13	11	10	11	11	12	10	4
EWOA	7	3	9	7	6	8	9	5	9	7	9
CLSGMFO	4	9	8	4	7	6	5	4	8	6	2
LGCMFO	5	6	4	1	5	5	3	2	5	5	3
CGSCA	13	13	11	12	12	11	13	13	10	11	5
RDWOA	9	2	10	5	9	7	10	6	6	9	11
ACWOA	12	11	12	11	13	12	12	12	11	12	8
GCHHO	6	1	5	2	8	9	6	3	7	8	6
LSCA	10	10	7	9	1	2	7	10	3	2	10
MGSMA	8	7	2	6	3	4	4	7	4	3	7

Table 3. *Cont.*

	F23	F24	F25	F26	F27	F28	F29	F30	+/−/=	Mean	Rank
CCRIME	2	2	3	3	3	1	1	1	N/A	1.67	1
CDLOBA	13	13	10	13	13	7	13	9	24/1/5	9.53	10
GACO	3	6	1	6	4	2	4	3	20/0/10	4.7	3
HGWO	9	8	11	10	10	11	10	12	30/0/0	10.53	11
EWOA	7	7	7	7	6	6	8	5	26/0/4	6.73	8
CLSGMFO	6	5	4	2	9	8	6	8	24/0/6	6.03	6
LGCMFO	5	4	5	1	8	4	5	6	25/0/5	4.77	4
CGSCA	11	10	13	11	11	13	12	13	30/0/0	11.6	13
RDWOA	8	11	8	9	5	10	7	4	28/0/2	7.93	9
ACWOA	12	12	12	12	12	12	11	11	30/0/0	11.57	12
GCHHO	10	9	6	8	7	3	9	2	24/1/5	6.37	7
LSCA	1	1	9	5	2	9	2	10	22/1/7	5.67	5
MGSMA	4	3	2	4	1	5	3	7	21/3/6	3.9	2

Subsequently, the Friedman test was employed to evaluate the efficacy of the algorithms. The ranking results of each algorithm are presented in Figure 5 according to the Friedman test. Upon examination of the ranking outcomes, it is evident that CCRIME attained the topmost position with a score of 2.59, while MGSMA secured the second position with a score of 4.10. This outcome effectively signifies that CCRIME outperformed MGSMA and establishes its superiority over other analogous algorithms. Figure 6 displays the convergence curves of various algorithms on certain functions. It is evident that CCRIME outperformed other comparable variant algorithms in terms of convergence performance.

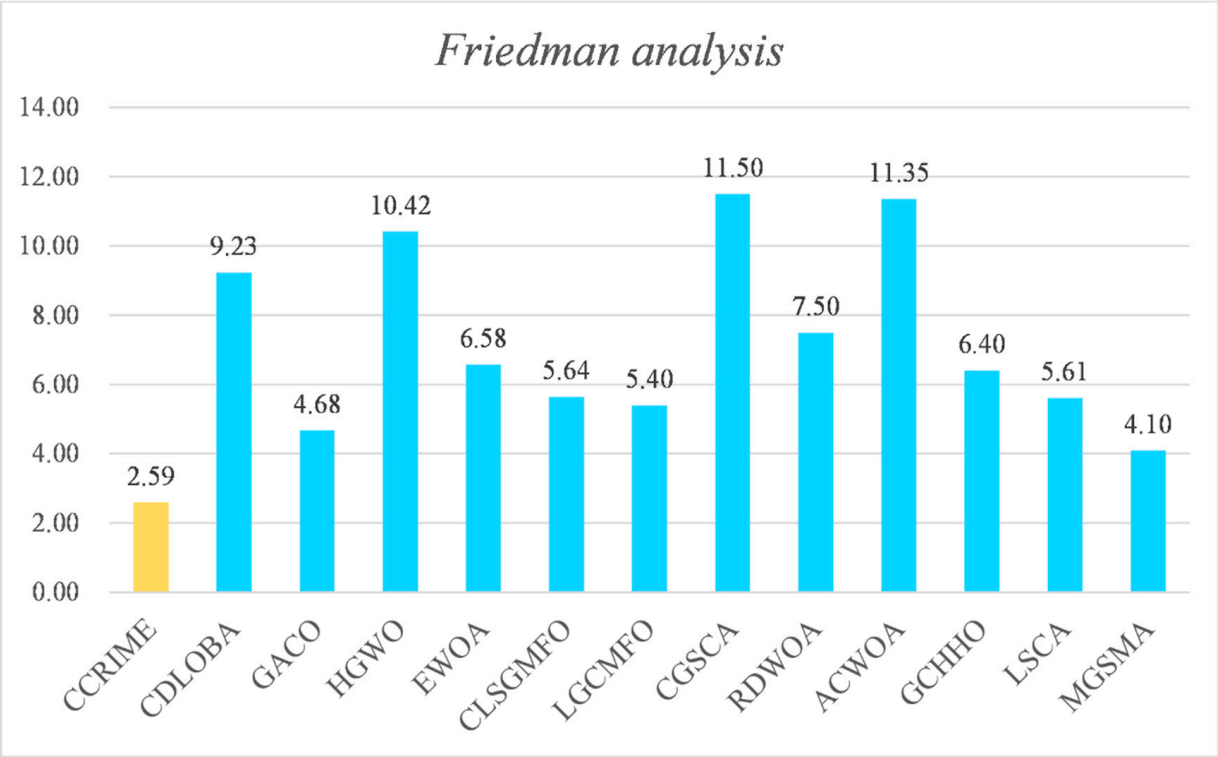


Figure 5. The Friedman ranking results of each algorithm.

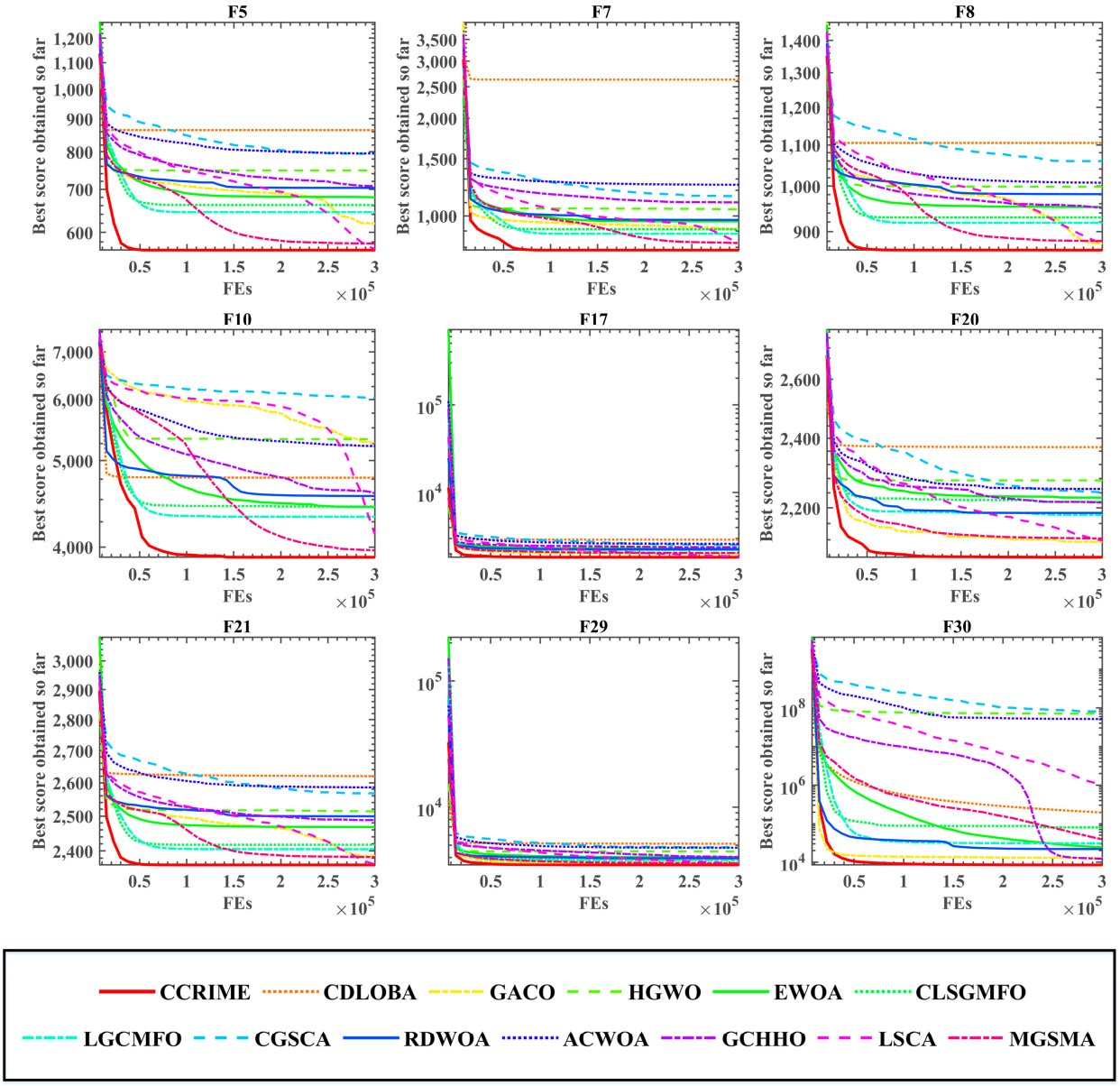


Figure 6. The convergence curves of all algorithms on some functions.

The core performance of CCRIME, an optimization algorithm based on swarm intelligence, was demonstrated through a comparative experiment with advanced variant algorithms. The results effectively indicate that CCRIME is an outstanding algorithm in this field. Owing to its formidable optimization capabilities, the proposed CCRIME exhibits versatility suitable for various applications, including but not limited to the resolution of the blocking flowtops problem [68], object tracking [69,70], plant disease recognition [71], and constrained optimization [72–74].

5.2. Feature Selection Experiments

This section first describes the evaluation criteria for the earthquake data prediction experiment. Then, the proposed BCCRIME-FKNN model is compared with other similar classic prediction models, such as BP, CART, RandomF, etc. Further, BCCRIME-FKNN is compared with other BCCRIME-based models, such as KNN, KELM, MLP, etc. Next, BCCRIME-FKNN is compared with other swarm intelligence-based FKNN prediction models, such as BPSO, BCS, etc. Finally, the key features selected by BCCRIME-FKNN are shown.

5.2.1. Experimental Setup

Since the original source of raw data for this paper is the waveform images acquired by each sensor, color moments, gray-level covariance matrices, and local binary patterns (LBP) were mainly used as feature extraction techniques in the preprocessing stage of the data. Subsequently, we extracted some shape attributes from the microseismic and blast images, including the area, bounding box, major axis, minor axis, eccentricity, orientation, Euler number, equivalent circle diameter, fullness, expansion, and perimeter. The experiments mainly used the ten-fold cross-validation approach to train, test, and validate the proposed algorithms. The experiment on feature selection was conducted using the benchmark function's experimental environment, focusing on incorporating five distinct feature selection evaluation metrics. The content in question is as follows.

A confusion matrix is a visual representation that documents the outcomes of classification predictions within the pattern recognition domain. The assessment of a classification models' performance is a crucial task, and a key indicator of this is the correlation between the predicted class and the true class attributes of the sample data. A confusion matrix can be utilized to derive evaluation metrics such as accuracy, specificity, and sensitivity. Binary classification problems involve categorizing samples into two distinct groups, namely positive and negative. The classification evaluation tool known as the confusion matrix comprises four distinct categories: true positives (*TP*), true negatives (*TN*), false positives (*FP*), and false negatives (*FN*).

The Accuracy (ACC) metric is a widely employed classification evaluation measure that assesses a classifier's capacity to accurately recognize samples. The accuracy range is bounded by the interval [0, 1]. A higher accuracy value, closer to 1, indicates superior classification performance of the classifier. The methodology for computing accuracy is as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{15}$$

The term "specificity" (SPE) pertains to the likelihood of correctly identifying an individual who is truly negative as negative. The metric assesses the proficiency of a classifier in identifying negative samples, thereby indicating its efficacy in identifying negative individuals. The method for calculating specificity is as follows:

$$Specificity = \frac{TN}{TN + FP} \tag{16}$$

The MCC metric quantifies the degree of association between diagnostic outcomes and true outcomes. In the context of balanced data, an increase in both the ACC and MCC scores indicates an improvement in the quality of predictions. However, in the case of unbalanced data, MCC serves as a more precise indicator of the predictor's quality compared to ACC. The procedure for computing MCC is as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{17}$$

F-Measure is the weighted average of precision (P) and recall (R). Precision refers to the proportion of actual positives among those classified as positive; recall is an indicator of coverage, that is, the number of cases classified as positive:

$$P = \frac{TP}{TP + FP} \tag{18}$$

$$R = \frac{TP}{TP + FN} \tag{19}$$

$$F = \frac{\left(\alpha^2 + 1\right) P * R}{\alpha^2 (P + R)}, \alpha = 1 \tag{20}$$

Next, the swarm intelligence optimization algorithms used in the feature selection experiment include BCCRIME, BRIME, BPSO, bMFO, bALO, BSSA, bMVO, and bCS. Table 4 shows the parameter values set for each algorithm, which are no different from the original values of the algorithm. Furthermore, owing to the characteristics of the feature selection technique, the algorithm's dimensionality is consistently established as the number of features present in the dataset, while the algorithm's population size is consistently established as 20. Two of the optimization algorithms among the set of eight, namely bCS and BMFO, are discretized based on the original algorithm. Algorithms such as BSSA and bALO have been referenced in the literature by other scholars. Furthermore, the experiment was conducted on a Windows Server 2008R2 operating system in order to ensure uniformity of conditions across all experiments. The fundamental components of this apparatus comprise an Intel Xeon (R) CPUE5-2660v3 (2.60 GHz) and 16 GB of RAM, in conjunction with Matlab2017b to execute the code.

Table 4. Parameter settings for the optimization algorithms.

Algorithms	**BCCRIME**	**bMFO**	**BSSA**	**bMFO**
Values	W = 5	W = 5	~	a = 2; b = 1
Algorithms	bALO	bMVO	BPSO	bCS
Values	~	Max = 1; Min = 0.2	wMax = 0.9; wMin = 0.2	pa = 0.25

5.2.2. Microseismic and Blast Dataset Experiment

This section aimed to assess the efficacy of the BCCRIME-FKNN model in practical earthquake prediction by utilizing real-world microseismic and blasting data for classification prediction. This study aimed to evaluate the efficacy of the BCCRIME and FKNN combination by comparing it with three alternative predictors, namely KELM, KNN, and MLP. The comparison was conducted using microseismic and blasting datasets. The experiment aimed to demonstrate the performance disparity between BCCRIME-FKNN and conventional methods. To achieve this, a comparative analysis was conducted between BCCRIME and four classical methods, namely CART and RandomF, among others.

We executed all tests in this paper in accordance with fair comparison principles, aligned with rules employed in previous research [75,76]. A fair comparison involves using consistent metrics, datasets, and evaluation criteria throughout all methods or models being assessed to guarantee impartial results. Hence, the algorithmic configurations and code were sourced from the MATLAB defaults. Furthermore, to demonstrate the superiority of BCCRIME in comparison to other analogous algorithms, the present study conducted a comparative analysis between BCCRIME and seven other swarm intelligence algorithms, namely BRIME, BPSO, bMFO, bALO, BSSA, bMVO, and bCS. In order to comprehensively demonstrate the dependability of the prediction outcomes, this section employed four evaluation techniques, namely, accuracy, specificity, MCC, and F-value, to more precisely assess the processing capability of the BCCRIME-FKNN model on actual data.

Using various classifiers in conjunction with BCCRIME is likely to yield varying classification outcomes. In order to evaluate the efficacy of the BCCRIME and FKNN combination, the former was integrated with three additional classifiers and subjected to a comparative experiment. The findings are illustrated in Figure 7. The superior performance of BCCRIME-FKNN in terms of accuracy, specificity, MCC, and F-value is evident when compared to other predictors. Additionally, the stability of the FKNN model is discernible through the utilization of box plots. On the contrary, the amalgamation of BCCRIME with KELM and MLP exhibits inadequate precision and lacks robustness. Thus, it can be inferred that the amalgamation of BCCRIME and FKNN is highly appropriate.

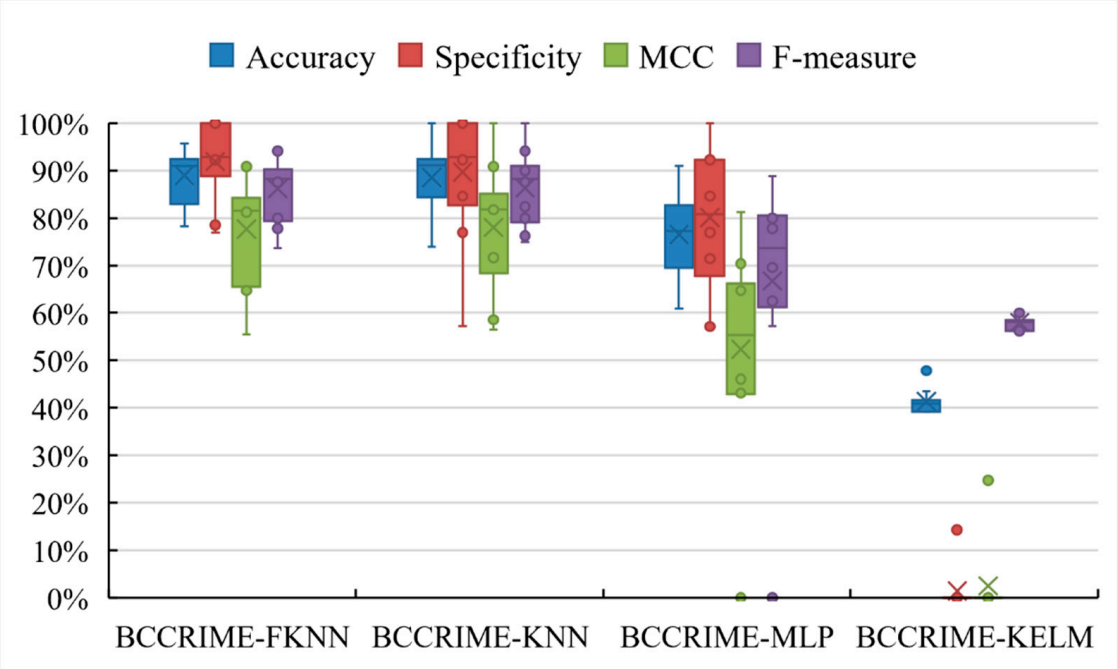


Figure 7. Results of five different classifiers based on BCCRIME.

This paper conducted a comparative experiment between BCCRIME-FKNN and classical classification methods to demonstrate the performance advantage of the former. Figure 8 displays the box plot outcomes of the conducted experiment. Evidently, BCCRIME-FKNN exhibited a substantial edge over alternative classification techniques. Within the set of algorithms considered, RandomF exhibited sporadic instances of favorable outcomes, yet its consistency was notably inferior to that of BCCRIME-FKNN. Conversely, the overall efficacies of CART and ELMforFS were generally subpar. The experimental findings indicate that BCCRIME-FKNN exhibited a significant advantage over traditional classification techniques.

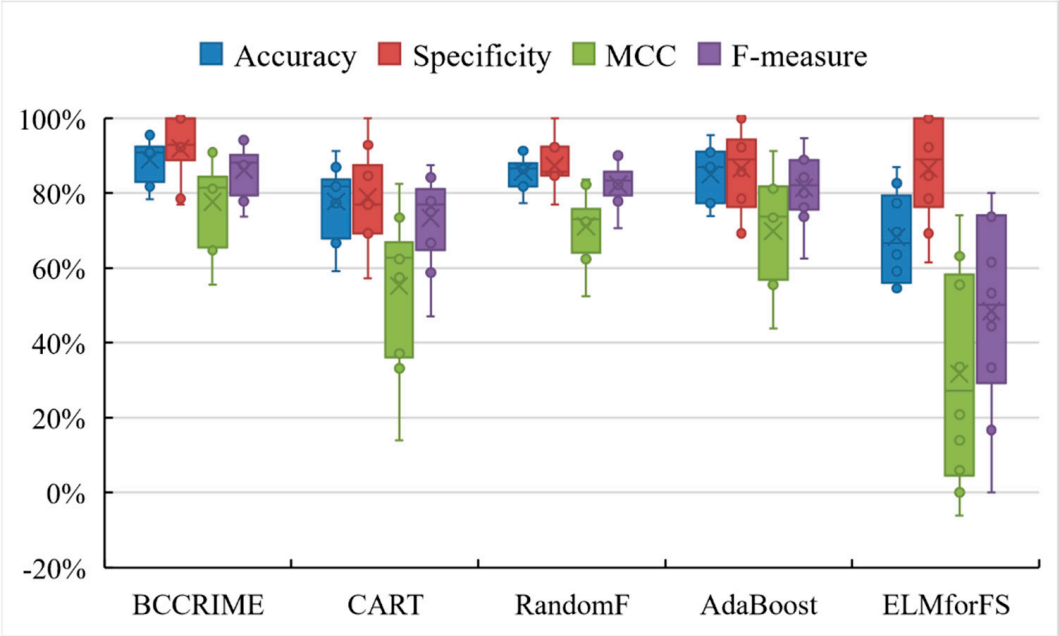


Figure 8. Comparison results of BCCRIME−FKNN with other classical methods.

The aforementioned experiments provide evidence that the amalgamation of FKNN and swarm intelligence algorithms yields superior performance. This study aimed to evaluate the efficacy of the BCCRIME-FKNN classification prediction model in comparison to other well-known swarm intelligence algorithms, such as BRIME, BPSO, bMFO, bALO, BSSA, bMVO, and bCS. The objective was to determine whether BCCRIME-FKNN outperforms these algorithms in terms of classification performance. A graphical representation of the comparison experiment's outcomes is depicted in Figure 9 through the utilization of a box plot. The data from BCCRIME exhibited the highest values across all four classification evaluation indicators, suggesting that the BCCRIME-FKNN model possesses robust predictive capabilities relative to similar algorithms and is well-suited for forecasting microseismic and blasting issues. In contrast to the original MFO algorithm, ICCRIME-FKNN exhibited superior accuracy and enhanced stability. The BCCRIME model-based comprehensive classification effect exhibited superior accuracy and stability in relation to precision and error. The BCCRIME model-based selection exhibited superior accuracy for both negative and positive cases, particularly in terms of specificity. Regarding Matthew's Correlation Coefficient (MCC), the BCCRIME model exhibited the highest correlation between the predicted and true values. The BCCRIME model exhibited a stronger comprehensive classification performance as indicated by its F-value.

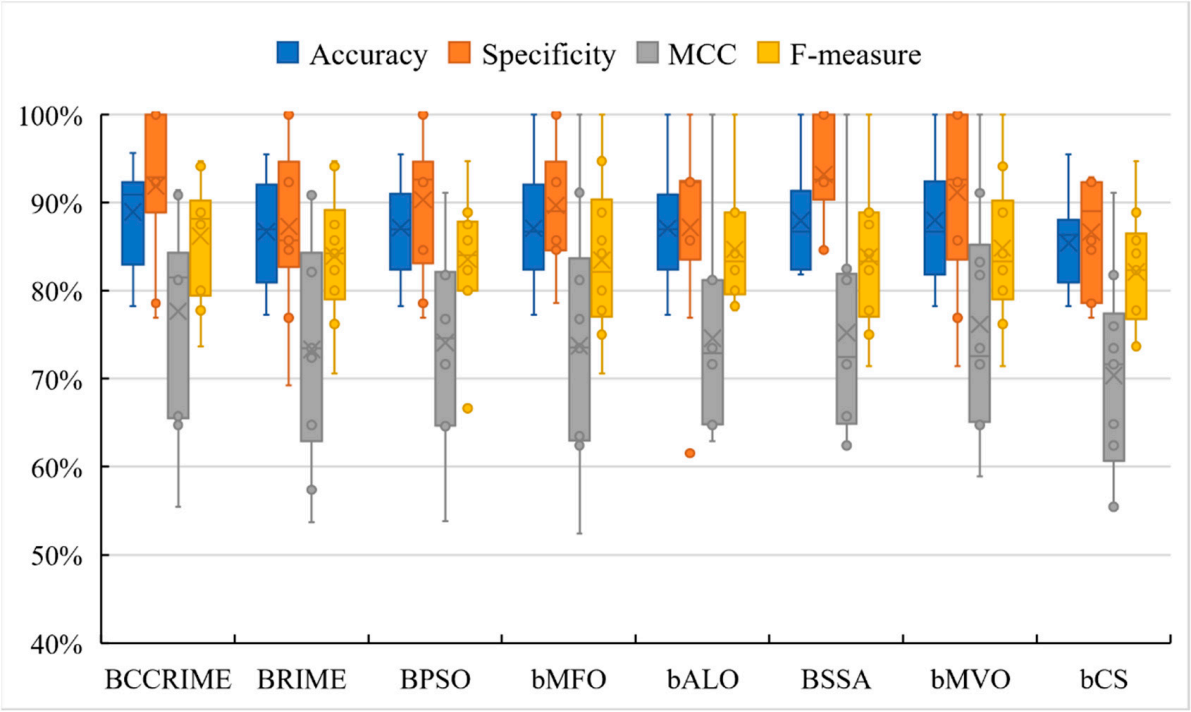


Figure 9. Comparison of BCCRIME and seven algorithms on four evaluation criteria.

In order to bolster the legitimacy of the comparative study, the present paper employed the Friedman test technique to authenticate and order the findings obtained from the experiment, which are presented in Table 5. Upon examination of the testing methodology, it can be concluded that BCCRIME-FKNN exhibited a high degree of stability, ranking first among the tested methods. Thus, based on the aforementioned experiments, it can be demonstrated that the BCCRIME-FKNN model is highly appropriate for aiding in the prediction of microseismic and blasting topics and is capable of efficiently categorizing microseismic and blasting datasets.

Table 5. Verification of the experimental results by using the Friedman test method.

Method		BCCRIME	BRIME	BPSO	bMFO	bALO	BSSA	bMVO	bCS
Accuracy	Avg	3.65	4.6	4.5	4.4	4.65	4.45	4.4	5.35
	Rank	**1**	6	5	2	7	4	2	8
Specificity	Avg	3.55	4.8	4.35	4.6	5	3.75	4.1	5.85
	Rank	**1**	6	4	5	7	2	3	8
MCC	Avg	3.75	4.6	4.55	4.2	4.7	4.6	4.4	5.2
	Rank	**1**	5	4	2	7	5	3	8
F-measure	Avg	3.5	4.5	4.5	4.55	4.55	4.65	4.6	5.15
	Rank	**1**	2	2	4	4	7	6	8

The experimental data from the CCI dataset based on the BCCRIME-FKNN model are presented in Table 6. The initial column denotes the label of ten-fold cross-validation, while the second column represents the count of selected features (SSFS). The remaining columns indicate the accuracy, sensitivity, MCC, and F-value employed in the aforementioned analysis. The results of the ten-fold cross-validation indicate an accuracy value of 90.91%, a specificity value of 0.9286, an MCC value of 0.8147, and an F-value of 0.8819. This finding demonstrates the efficacy of integrating BCCRIME and FKNN as a microseismic and blasting classifier model.

Table 6. Detailed results obtained by BCCRIME.

Fold	SSFS	Accuracy	Specificity	MCC	F-Measure
#1	91	0.909	0.923	0.812	0.889
#2	81	0.909	1.000	0.821	0.875
#3	83	0.955	1.000	0.909	0.941
#4	57	0.909	0.923	0.812	0.889
#5	82	0.957	0.929	0.914	0.947
#6	89	0.818	0.769	0.647	0.800
#7	79	0.833	0.929	0.657	0.778
#8	63	0.913	0.929	0.818	0.889
#9	87	0.783	0.786	0.555	0.737
#10	80	0.909	1.000	0.821	0.875
AVG	~	**0.909**	**0.929**	**0.815**	**0.882**
STD	~	**0.058**	**0.082**	**0.118**	**0.069**

6. Conclusions and Future Directions

This paper proposed a RIME algorithm based on a vertical and horizontal cross-search strategy (CCRIME) to improve the quality of the solution obtained by the RIME algorithm and further enhance its search capability. By constructing a binary version of CCRIME, the key parameters in FKNN were optimized using binary conversion methods. In addition, this paper also proposed a BCCRIME based on discrete CCRIME, which converts the solution after searching into real numbers containing only 0 and 1 through the S-shaped function conversion method for solving feature selection problems. This paper studied the performance of CCRIME from multiple angles using 30 benchmark functions in IEEE CEC2017, mainly by conducting basic algorithm comparison experiments and advanced variant algorithm comparison experiments. Then, this paper also used collected microseismic and blasting data for classification prediction to verify the ability of the BCCRIME-FKNN model to handle real data, using four evaluation methods (the accuracy, specificity, MCC, and F-value) to display the reliability of the prediction results comprehensively, and it conducted ten-fold cross-validation on the experimental results. In summary, this paper proposed an effective method to solve the real-time rock mass stability monitoring problem during deep well mineral resource exploitation. By using the CCRIME algorithm combined with the FKNN model, this paper successfully identified microseismic

and blasting signals and achieved high recognition accuracy. This research provides new ideas and methods for real-time monitoring of rock mass stability during deep well mineral resource exploitation.

In future research, it is possible to try to combine CCRIME with other optimization algorithms to improve its performance. It is also possible to explore the use of other evaluation methods to evaluate the reliability of the prediction results and conduct a more in-depth analysis of the experimental results.

Author Contributions: Conceptualization, W.Z.; Data curation, Z.L. and A.A.H.; Formal analysis, Z.L., A.A.H., S.W. and Y.Z.; Funding acquisition, S.W. and Y.Z.; Investigation, Z.L., A.A.H., H.C. and Y.Z.; Methodology, S.W. and H.C.; Resources, Z.L.; Software, W.Z.; Supervision, H.C.; Validation, W.Z. and H.C.; Writing—original draft, W.Z. and A.A.H.; Writing—review and editing, S.W. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (52204263), Natural Science Foundation of Zhejiang Province (LZ22F020005); MRC (MC_PC_17171); Royal Society (RP202G0230); BHF (AA/18/3/34220); Hope Foundation for Cancer Research (RM60G0680); GCRF (P202PF11); Sino-UK Industrial Fund (RP202G0289); LIAS (P202ED10, P202RE969); Data Science Enhancement Fund (P202RE237); Fight for Sight (24NN201); Sino-UK Education Fund (OP202006); and BBSRC (RM32G0178B8).

Data Availability Statement: The data involved in this study are all public data, which can be downloaded through public channels.

Acknowledgments: This work was supported in part by the: National Natural Science Foundation of China (52204263), Natural Science Foundation of Zhejiang Province (LZ22F020005); MRC (MC_PC_17171); Royal Society (RP202G0230); BHF (AA/18/3/34220); Hope Foundation for Cancer Research (RM60G0680); GCRF (P202PF11); Sino-UK Industrial Fund (RP202G0289); LIAS (P202ED10, P202RE969); Data Science Enhancement Fund (P202RE237); Fight for Sight (24NN201); Sino-UK Education Fund (OP202006); and BBSRC (RM32G0178B8).

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

Appendix A

Table A1. The means and standard deviations obtained during the experiment.

	F1		F2		F3	
	AVG	**STD**	**AVG**	**STD**	**AVG**	**STD**
CCRIME	4.2352×10^{3}	4.7608×10^{3}	2.1640×10^{2}	3.5195×10^{1}	3.0911×10^{2}	5.3019×10^{0}
RIME	8.9768×10^{3}	5.8909×10^{3}	1.1918×10^{3}	1.6094×10^{3}	3.0186×10^{2}	9.5716×10^{-1}
MVO	1.2080×10^{4}	6.8991×10^{3}	9.2436×10^{2}	1.2608×10^{3}	3.0037×10^{2}	1.5597×10^{-1}
BA	5.3849×10^{5}	3.6639×10^{5}	$\mathbf{2.0000 \times 10^{2}}$	$\mathbf{1.3410 \times 10^{-4}}$	3.0009×10^{2}	8.3596×10^{-2}
HHO	1.0454×10^{7}	2.0591×10^{6}	6.3435×10^{11}	1.0304×10^{12}	4.9178×10^{3}	1.8832×10^{3}
PSO	1.3465×10^{8}	1.7454×10^{7}	4.2792×10^{13}	4.5075×10^{13}	6.4025×10^{2}	5.2610×10^{1}
SSA	$\mathbf{3.9871 \times 10^{3}}$	4.9271×10^{3}	2.0535×10^{2}	1.8872×10^{1}	$\mathbf{3.0000 \times 10^{2}}$	$\mathbf{1.0612 \times 10^{-8}}$
WOA	3.3518×10^{6}	2.2122×10^{6}	6.8389×10^{21}	2.8707×10^{22}	1.6825×10^{5}	5.5436×10^{4}
JAYA	5.4373×10^{9}	1.0080×10^{9}	2.7325×10^{31}	1.2670×10^{32}	4.2467×10^{4}	8.1926×10^{3}
PO	3.7297×10^{7}	8.5935×10^{7}	2.2738×10^{38}	1.0759×10^{39}	5.1376×10^{4}	1.1204×10^{4}
SFS	2.4036×10^{8}	2.9173×10^{8}	1.1897×10^{24}	6.4968×10^{24}	2.5396×10^{4}	7.9311×10^{3}
SMA	7.7115×10^{3}	7.9786×10^{3}	2.0000×10^{2}	4.5202×10^{-3}	3.0002×10^{2}	1.7494×10^{-2}
HGS	5.9547×10^{3}	$\mathbf{3.7800 \times 10^{3}}$	2.4649×10^{2}	1.1641×10^{1}	1.5381×10^{3}	3.5435×10^{3}

Table A1. *Cont.*

	F4		F5		F6	
	AVG	STD	AVG	STD	AVG	STD
CCRIME	4.8159×10^{2}	2.2211×10^{1}	$\mathbf{5.6016 \times 10^{2}}$	1.5706×10^{1}	$\mathbf{6.0000 \times 10^{2}}$	$\mathbf{4.5903 \times 10^{-7}}$
RIME	4.9089×10^{2}	2.3365×10^{1}	5.8617×10^{2}	2.0042×10^{1}	6.0031×10^{2}	2.3598×10^{-1}
MVO	4.8857×10^{2}	$\mathbf{3.6765 \times 10^{0}}$	5.8572×10^{2}	2.8558×10^{1}	6.0790×10^{2}	6.4185×10^{0}
BA	4.7796×10^{2}	3.2553×10^{1}	8.0505×10^{2}	4.8012×10^{1}	6.7252×10^{2}	8.1846×10^{0}
HHO	5.2696×10^{2}	3.7392×10^{1}	7.2639×10^{2}	3.9574×10^{1}	6.6215×10^{2}	5.2602×10^{0}
PSO	4.7087×10^{2}	2.9010×10^{1}	7.2702×10^{2}	3.5931×10^{1}	6.5308×10^{2}	1.5342×10^{1}
SSA	4.8848×10^{2}	2.3276×10^{1}	6.1671×10^{2}	3.6354×10^{1}	6.2735×10^{2}	7.2336×10^{0}
WOA	5.5205×10^{2}	3.8969×10^{1}	7.6790×10^{2}	5.3069×10^{1}	6.6996×10^{2}	8.9938×10^{0}
JAYA	7.6986×10^{2}	5.6320×10^{1}	7.3242×10^{2}	$\mathbf{1.4973 \times 10^{1}}$	6.2064×10^{2}	2.0354×10^{0}
PO	$\mathbf{5.1868 \times 10^{2}}$	1.5551×10^{2}	5.8202×10^{2}	5.0596×10^{1}	6.1731×10^{2}	2.1509×10^{1}
SFS	6.2382×10^{2}	8.9232×10^{1}	6.8124×10^{2}	4.0719×10^{1}	6.2350×10^{2}	9.7230×10^{0}
SMA	4.8708×10^{2}	5.3709×10^{0}	5.9261×10^{2}	2.7473×10^{1}	6.0108×10^{2}	7.0865×10^{-1}
HGS	4.8118×10^{2}	2.1795×10^{1}	6.1429×10^{2}	3.0497×10^{1}	6.0159×10^{2}	3.0552×10^{0}
	F7		F8		F9	
	AVG	STD	AVG	STD	AVG	STD
CCRIME	$\mathbf{7.8480 \times 10^{2}}$	$\mathbf{1.5077 \times 10^{1}}$	$\mathbf{8.6683 \times 10^{2}}$	1.7838×10^{1}	$\mathbf{9.1086 \times 10^{2}}$	$\mathbf{2.7764 \times 10^{1}}$
RIME	8.1285×10^{2}	1.9469×10^{1}	8.8054×10^{2}	2.6002×10^{1}	1.3026×10^{3}	4.1007×10^{2}
MVO	8.3459×10^{2}	3.6224×10^{1}	8.9459×10^{2}	2.8450×10^{1}	1.8553×10^{3}	1.8613×10^{3}
BA	1.6226×10^{3}	1.7681×10^{2}	1.0674×10^{3}	5.5317×10^{1}	1.4521×10^{4}	4.7355×10^{3}
HHO	1.2007×10^{3}	7.4765×10^{1}	9.5793×10^{2}	2.0453×10^{1}	6.4783×10^{3}	7.8248×10^{2}
PSO	9.1867×10^{2}	1.6189×10^{1}	9.9796×10^{2}	2.2256×10^{1}	5.4552×10^{3}	2.4482×10^{3}
SSA	8.6523×10^{2}	3.9952×10^{1}	9.0912×10^{2}	2.7688×10^{1}	2.6213×10^{3}	1.0113×10^{3}
WOA	1.2327×10^{3}	7.4548×10^{1}	1.0119×10^{3}	4.3306×10^{1}	8.2725×10^{3}	2.9747×10^{3}
JAYA	1.0282×10^{3}	1.7143×10^{1}	1.0290×10^{3}	$\mathbf{1.3246 \times 10^{1}}$	2.9391×10^{3}	5.2085×10^{2}
PO	8.1384×10^{2}	1.8313×10^{1}	9.2549×10^{2}	6.5905×10^{1}	3.4627×10^{3}	1.1328×10^{3}
SFS	9.4721×10^{2}	4.9220×10^{1}	9.4509×10^{2}	2.9632×10^{1}	3.4738×10^{3}	1.1552×10^{3}
SMA	8.3175×10^{2}	2.8146×10^{1}	8.9092×10^{2}	2.9294×10^{1}	2.3501×10^{3}	1.3994×10^{3}
HGS	8.8299×10^{2}	3.4544×10^{1}	9.2650×10^{2}	2.1175×10^{1}	3.6304×10^{3}	1.0597×10^{3}
	F10		F11		F12	
	AVG	STD	AVG	STD	AVG	STD
CCRIME	4.1045×10^{3}	5.9985×10^{2}	$\mathbf{1.1552 \times 10^{3}}$	$\mathbf{2.1785 \times 10^{1}}$	$\mathbf{3.6782 \times 10^{5}}$	$\mathbf{2.8825 \times 10^{5}}$
RIME	$\mathbf{3.6573 \times 10^{3}}$	4.5199×10^{2}	1.1735×10^{3}	3.4298×10^{1}	2.4857×10^{6}	2.1008×10^{6}
MVO	4.3025×10^{3}	6.7601×10^{2}	1.2623×10^{3}	5.5179×10^{1}	3.5501×10^{6}	2.8495×10^{6}
BA	5.6097×10^{3}	7.4497×10^{2}	1.3274×10^{3}	8.3832×10^{1}	1.8731×10^{6}	1.4604×10^{6}
HHO	5.4742×10^{3}	7.5731×10^{2}	1.2357×10^{3}	4.0220×10^{1}	8.9462×10^{6}	5.1257×10^{6}
PSO	5.9792×10^{3}	5.0901×10^{2}	1.2867×10^{3}	3.5054×10^{1}	2.7076×10^{7}	1.0591×10^{7}
SSA	4.6895×10^{3}	6.4525×10^{2}	1.2572×10^{3}	5.0625×10^{1}	2.1004×10^{6}	1.7153×10^{6}
WOA	6.3111×10^{3}	8.8011×10^{2}	1.4670×10^{3}	8.4714×10^{1}	3.7797×10^{7}	2.7709×10^{7}
JAYA	8.0555×10^{3}	$\mathbf{2.6052 \times 10^{2}}$	1.9427×10^{3}	1.6547×10^{2}	1.5809×10^{8}	5.0242×10^{7}
PO	4.4613×10^{3}	1.1401×10^{3}	1.3041×10^{3}	4.7553×10^{2}	2.5781×10^{8}	5.5878×10^{8}
SFS	5.6953×10^{3}	6.0136×10^{2}	1.3524×10^{3}	6.1781×10^{1}	2.3275×10^{7}	1.6640×10^{7}
SMA	4.2151×10^{3}	6.4265×10^{2}	1.2266×10^{3}	5.0459×10^{1}	1.0666×10^{6}	8.7671×10^{5}
HGS	3.8801×10^{3}	3.7826×10^{2}	1.2254×10^{3}	3.3054×10^{1}	7.1680×10^{5}	6.1027×10^{5}

Table A1. *Cont.*

	F13		F14		F15	
	AVG	STD	AVG	STD	AVG	STD
CCRIME	1.5556×10^4	$\mathbf{1.4199 \times 10^4}$	1.5925×10^4	1.6541×10^4	$\mathbf{5.0471 \times 10^3}$	$\mathbf{5.0119 \times 10^3}$
RIME	$\mathbf{1.5035 \times 10^4}$	1.7138×10^4	1.2460×10^4	9.4411×10^3	1.0450×10^4	9.6709×10^3
MVO	9.4989×10^4	6.0163×10^4	7.5535×10^3	5.5593×10^3	1.4348×10^4	1.3115×10^4
BA	3.4185×10^5	1.0445×10^5	5.9855×10^3	$\mathbf{3.1012 \times 10^3}$	1.0087×10^5	5.8172×10^4
HHO	3.4512×10^5	1.6145×10^5	3.8293×10^4	5.4199×10^4	5.2613×10^4	3.1858×10^4
PSO	4.2939×10^6	1.4517×10^6	9.7861×10^3	5.8728×10^3	4.7090×10^5	2.1419×10^5
SSA	1.3604×10^5	1.0059×10^5	$\mathbf{5.3642 \times 10^3}$	3.9091×10^3	6.4894×10^4	3.9840×10^4
WOA	1.4138×10^5	9.5296×10^4	8.6245×10^5	8.7017×10^5	7.9425×10^4	5.8177×10^4
JAYA	6.4069×10^6	4.6682×10^6	7.2025×10^4	3.3109×10^4	4.0494×10^6	3.1648×10^6
PO	2.4867×10^8	4.1083×10^8	4.2708×10^5	4.0938×10^5	1.2277×10^5	6.1289×10^5
SFS	4.9386×10^5	2.8899×10^5	4.6489×10^4	3.8792×10^4	2.0956×10^4	1.1440×10^4
SMA	4.0304×10^4	2.7035×10^4	3.5754×10^4	1.1732×10^4	3.0833×10^4	1.3257×10^4
HGS	2.7342×10^4	2.6194×10^4	4.5396×10^4	3.4197×10^4	2.0004×10^4	1.5963×10^4
	F16		F17		F18	
	AVG	STD	AVG	STD	AVG	STD
CCRIME	$\mathbf{2.2136 \times 10^3}$	2.5078×10^2	$\mathbf{1.8206 \times 10^3}$	$\mathbf{5.9286 \times 10^1}$	$\mathbf{1.0194 \times 10^5}$	$\mathbf{7.4067 \times 10^4}$
RIME	2.3114×10^3	2.6071×10^2	2.0797×10^3	1.6901×10^2	2.5958×10^5	1.9356×10^5
MVO	2.4394×10^3	2.5265×10^2	2.0807×10^3	1.7179×10^2	1.5999×10^5	1.0959×10^5
BA	3.5780×10^3	4.0794×10^2	2.9109×10^3	3.6237×10^2	1.7767×10^5	1.4956×10^5
HHO	3.1501×10^3	4.2626×10^2	2.4808×10^3	2.9478×10^2	1.0207×10^6	1.1682×10^6
PSO	2.8850×10^3	2.6918×10^2	2.2511×10^3	1.8126×10^2	1.8792×10^5	1.1956×10^5
SSA	2.4204×10^3	2.7047×10^2	2.0334×10^3	1.6308×10^2	1.5527×10^5	9.9610×10^4
WOA	3.6317×10^3	4.7076×10^2	2.4345×10^3	2.2944×10^2	2.9139×10^6	2.9975×10^6
JAYA	3.4847×10^3	$\mathbf{1.4418 \times 10^2}$	2.3268×10^3	9.4815×10^1	1.6687×10^6	7.4539×10^5
PO	3.1960×10^3	6.9761×10^2	2.7776×10^3	2.1281×10^2	4.4167×10^6	6.9188×10^6
SFS	2.6619×10^3	3.5815×10^2	2.0352×10^3	1.2525×10^2	8.2484×10^5	6.7687×10^5
SMA	2.4119×10^3	3.1437×10^2	2.1563×10^3	1.9440×10^2	3.6249×10^5	3.5788×10^5
HGS	2.7071×10^3	2.8874×10^2	2.2062×10^3	2.1884×10^2	1.9029×10^5	1.6169×10^5
	F19		F20		F21	
	AVG	STD	AVG	STD	AVG	STD
CCRIME	$\mathbf{9.5736 \times 10^3}$	$\mathbf{1.0082 \times 10^4}$	$\mathbf{2.1692 \times 10^3}$	1.0123×10^2	$\mathbf{2.3578 \times 10^3}$	1.3366×10^1
RIME	1.5689×10^4	1.3709×10^4	2.3715×10^3	1.4845×10^2	2.3895×10^3	2.2081×10^1
MVO	2.0597×10^4	1.7907×10^4	2.3264×10^3	1.3923×10^2	2.3893×10^3	2.3475×10^1
BA	6.5008×10^5	1.9533×10^5	2.8926×10^3	1.9027×10^2	2.6432×10^3	6.9212×10^1
HHO	2.6299×10^5	1.8066×10^5	2.6595×10^3	1.7370×10^2	2.5271×10^3	4.6083×10^1
PSO	1.4020×10^6	7.3947×10^5	2.6821×10^3	1.4144×10^2	2.5329×10^3	3.7926×10^1
SSA	2.8477×10^5	1.2392×10^5	2.3643×10^3	1.1834×10^2	2.4076×10^3	2.8586×10^1
WOA	2.4239×10^6	2.0654×10^6	2.7622×10^3	1.8807×10^2	2.5863×10^3	6.6913×10^1
JAYA	9.1962×10^5	1.0108×10^6	2.5928×10^3	$\mathbf{8.4174 \times 10^1}$	2.5196×10^3	$\mathbf{1.2270 \times 10^1}$
PO	2.3313×10^7	3.6420×10^7	2.7411×10^3	1.7903×10^2	2.3656×10^3	2.1137×10^1
SFS	3.0252×10^4	2.4268×10^4	2.3835×10^3	1.3154×10^2	2.4383×10^3	2.6660×10^1
SMA	4.0426×10^4	1.9482×10^4	2.4319×10^3	1.9788×10^2	2.3835×10^3	2.2863×10^1
HGS	1.3613×10^4	1.3642×10^4	2.5060×10^3	1.9690×10^2	2.4220×10^3	3.6591×10^1

Table A1. *Cont.*

	F22		F23		F24	
	AVG	STD	AVG	STD	AVG	STD
CCRIME	$\mathbf{2.3000 \times 10^3}$	$\mathbf{2.9252 \times 10^{-13}}$	$\mathbf{2.7118 \times 10^3}$	$\mathbf{1.5448 \times 10^1}$	$\mathbf{2.8903 \times 10^3}$	2.3188×10^1
RIME	4.1442×10^3	1.4998×10^3	2.7371×10^3	2.3874×10^1	2.9404×10^3	3.0717×10^1
MVO	5.3506×10^3	1.2360×10^3	2.7356×10^3	2.7037×10^1	2.8965×10^3	$\mathbf{2.1527 \times 10^1}$
BA	7.1667×10^3	6.4661×10^2	3.3318×10^3	1.6747×10^2	3.3301×10^3	1.4862×10^2
HHO	5.8838×10^3	2.4256×10^3	3.1616×10^3	1.1375×10^2	3.4562×10^3	1.3046×10^2
PSO	5.6505×10^3	2.5817×10^3	3.1309×10^3	1.3014×10^2	3.1925×10^3	8.7834×10^1
SSA	4.2964×10^3	2.0629×10^3	2.7496×10^3	2.3815×10^1	2.9154×10^3	3.3122×10^1
WOA	6.4126×10^3	2.0489×10^3	3.0203×10^3	8.6712×10^1	3.1479×10^3	8.8336×10^1
JAYA	2.7871×10^3	7.7640×10^1	2.9759×10^3	2.7951×10^1	3.1246×10^3	2.3540×10^1
PO	4.2630×10^3	1.3865×10^3	2.8868×10^3	1.2524×10^2	3.2995×10^3	8.3349×10^1
SFS	2.4666×10^3	1.0963×10^2	2.8647×10^3	4.6289×10^1	3.0356×10^3	4.5109×10^1
SMA	5.8160×10^3	8.7287×10^2	2.7456×10^3	2.5447×10^1	2.9211×10^3	2.4033×10^1
HGS	5.2952×10^3	1.1299×10^3	2.7719×10^3	3.8856×10^1	3.0209×10^3	4.8575×10^1
	F25		F26		F27	
	AVG	STD	AVG	STD	AVG	STD
CCRIME	$\mathbf{2.8871 \times 10^3}$	$\mathbf{1.9408 \times 10^0}$	$\mathbf{4.1816 \times 10^3}$	3.0686×10^2	3.2146×10^3	$\mathbf{7.6152 \times 10^0}$
RIME	2.8927×10^3	1.5237×10^1	4.5570×10^3	5.3615×10^2	3.2242×10^3	1.2182×10^1
MVO	2.8887×10^3	1.0184×10^1	4.5202×10^3	4.1884×10^2	3.2133×10^3	1.2663×10^1
BA	2.9108×10^3	2.2000×10^1	9.0807×10^3	2.8195×10^3	3.4683×10^3	1.6138×10^2
HHO	2.9114×10^3	2.1235×10^1	6.6312×10^3	1.8435×10^3	3.3578×10^3	1.5524×10^2
PSO	2.8996×10^3	2.1281×10^1	4.8801×10^3	1.9824×10^3	3.2021×10^3	1.0908×10^2
SSA	2.8972×10^3	2.1530×10^1	4.5765×10^3	7.5915×10^2	3.2319×10^3	1.4168×10^1
WOA	2.9527×10^3	3.2256×10^1	7.4628×10^3	1.4801×10^3	3.3319×10^3	6.3936×10^1
JAYA	2.9703×10^3	2.4551×10^1	6.4704×10^3	1.0823×10^3	3.3443×10^3	2.6258×10^1
PO	2.8954×10^3	1.1156×10^1	5.7900×10^3	1.4064×10^3	$\mathbf{3.3742 \times 10^3}$	7.0381×10^1
SFS	2.9618×10^3	2.3297×10^1	5.1548×10^3	1.2769×10^3	3.3221×10^3	3.2351×10^1
SMA	2.8882×10^3	6.9775×10^0	4.5521×10^3	$\mathbf{2.3921 \times 10^2}$	3.2142×10^3	1.1031×10^1
HGS	2.8893×10^3	1.1222×10^1	4.8307×10^3	6.3208×10^2	3.2267×10^3	1.3656×10^1
	F28		F29		F30	
	AVG	STD	AVG	STD	AVG	STD
CCRIME	3.1889×10^3	4.3821×10^1	$\mathbf{3.5239 \times 10^3}$	$\mathbf{1.3486 \times 10^2}$	$\mathbf{8.1509 \times 10^3}$	$\mathbf{3.1391 \times 10^3}$
RIME	3.2232×10^3	3.1430×10^1	3.6854×10^3	1.5061×10^2	1.7204×10^4	1.1755×10^4
MVO	3.2059×10^3	4.2508×10^1	3.7721×10^3	1.8749×10^2	8.9953×10^5	7.5090×10^5
BA	$\mathbf{3.1333 \times 10^3}$	5.1445×10^1	4.8474×10^3	3.8282×10^2	1.3214×10^6	9.9215×10^5
HHO	3.2534×10^3	2.8765×10^1	4.4460×10^3	3.1369×10^2	1.7637×10^6	8.4530×10^5
PSO	3.2477×10^3	2.1766×10^1	4.2880×10^3	2.5551×10^2	3.0728×10^6	1.0512×10^6
SSA	3.1958×10^3	6.5395×10^1	3.8934×10^3	2.3205×10^2	1.2307×10^6	7.7586×10^5
WOA	3.2992×10^3	3.0927×10^1	4.7666×10^3	3.9102×10^2	8.9236×10^6	6.4166×10^6
JAYA	3.5656×10^3	5.4568×10^1	4.5121×10^3	1.3845×10^2	1.3755×10^7	4.3802×10^6
PO	3.8115×10^3	$\mathbf{5.3452 \times 10^2}$	4.5510×10^3	3.5897×10^2	4.4642×10^7	7.7357×10^7
SFS	3.3704×10^3	4.8728×10^1	4.0549×10^3	2.4728×10^2	7.6159×10^5	5.3012×10^5
SMA	3.2411×10^3	4.1023×10^1	3.7588×10^3	1.5385×10^2	1.5936×10^4	4.8792×10^3
HGS	3.2060×10^3	3.9670×10^1	3.7887×10^3	2.1151×10^2	5.3275×10^4	9.6043×10^4

Table A2. The mean and standard deviation obtained on each function.

	F1		**F2**		**F3**	
	AVG	**STD**	**AVG**	**STD**	**AVG**	**STD**
CCRMIME	2.9771×10^{3}	3.8917×10^{3}	$\mathbf{2.5406 \times 10^{2}}$	$\mathbf{1.9078 \times 10^{2}}$	3.0759×10^{2}	4.0478×10^{0}
CDLOBA	5.5074×10^{3}	5.6453×10^{3}	3.8938×10^{13}	2.1254×10^{14}	8.1427×10^{2}	1.2827×10^{3}
GACO	8.3026×10^{3}	7.2579×10^{3}	6.8740×10^{11}	2.1754×10^{12}	1.0466×10^{3}	9.7701×10^{2}
HGWO	7.9888×10^{9}	1.5486×10^{9}	1.1912×10^{34}	4.0275×10^{34}	7.8849×10^{4}	5.8962×10^{3}
EWOA	4.7411×10^{3}	5.9455×10^{3}	1.4645×10^{13}	6.7222×10^{13}	2.8210×10^{3}	1.9165×10^{3}
CLSGMFO	6.0602×10^{3}	6.6649×10^{3}	7.8325×10^{12}	2.6801×10^{13}	3.6090×10^{3}	2.5337×10^{3}
LGCMFO	7.7790×10^{3}	7.6606×10^{3}	3.6873×10^{12}	7.8929×10^{12}	7.3198×10^{3}	3.1921×10^{3}
CGSCA	1.4432×10^{10}	2.4865×10^{9}	3.2546×10^{35}	1.0980×10^{36}	4.2376×10^{4}	7.5248×10^{3}
RDWOA	7.1219×10^{6}	1.1871×10^{7}	1.1044×10^{16}	3.2277×10^{16}	2.0186×10^{4}	8.1217×10^{3}
ACWOA	5.9692×10^{9}	2.4554×10^{9}	5.2797×10^{33}	1.5160×10^{34}	4.9167×10^{4}	1.0378×10^{4}
GCHHO	$\mathbf{2.7602 \times 10^{3}}$	$\mathbf{3.4855 \times 10^{3}}$	2.2403×10^{7}	1.0442×10^{8}	5.7462×10^{2}	2.3346×10^{2}
LSCA	7.8562×10^{7}	1.1356×10^{8}	5.5566×10^{23}	2.1222×10^{24}	6.1157×10^{3}	2.5955×10^{3}
MGSMA	5.9801×10^{3}	5.7166×10^{3}	2.8944×10^{2}	2.8045×10^{2}	$\mathbf{3.0006 \times 10^{2}}$	$\mathbf{2.2915 \times 10^{-2}}$
	F4		F5		F6	
	AVG	STD	AVG	STD	AVG	STD
CCRMIME	4.8076×10^{2}	2.5159×10^{1}	$\mathbf{5.6346 \times 10^{2}}$	1.7012×10^{1}	$\mathbf{6.0000 \times 10^{2}}$	$\mathbf{3.1492 \times 10^{-4}}$
CDLOBA	$\mathbf{4.7172 \times 10^{2}}$	3.7479×10^{1}	8.6452×10^{2}	7.2789×10^{1}	6.6989×10^{2}	8.7201×10^{0}
GACO	4.8384×10^{2}	1.7228×10^{1}	6.1940×10^{2}	6.7666×10^{1}	6.0055×10^{2}	7.1058×10^{-1}
HGWO	9.1588×10^{2}	7.9238×10^{1}	7.4874×10^{2}	$\mathbf{1.5709 \times 10^{1}}$	6.3615×10^{2}	3.3548×10^{0}
EWOA	4.9190×10^{2}	3.5247×10^{1}	6.7997×10^{2}	4.2813×10^{1}	6.1813×10^{2}	8.3166×10^{0}
CLSGMFO	4.9381×10^{2}	3.0035×10^{1}	6.6178×10^{2}	3.1982×10^{1}	6.1748×10^{2}	7.9104×10^{0}
LGCMFO	4.9227×10^{2}	2.6894×10^{1}	6.4510×10^{2}	3.7981×10^{1}	6.1269×10^{2}	8.5660×10^{0}
CGSCA	1.6769×10^{3}	2.9093×10^{2}	7.9477×10^{2}	1.7393×10^{1}	6.5242×10^{2}	6.9915×10^{0}
RDWOA	5.1466×10^{2}	2.3748×10^{1}	7.0341×10^{2}	5.7872×10^{1}	6.1446×10^{2}	6.1420×10^{0}
ACWOA	1.2930×10^{3}	5.6474×10^{2}	7.9527×10^{2}	2.5487×10^{1}	6.6905×10^{2}	6.6192×10^{0}
GCHHO	4.9563×10^{2}	2.6817×10^{1}	7.0812×10^{2}	3.2537×10^{1}	6.5106×10^{2}	6.7242×10^{0}
LSCA	5.1775×10^{2}	2.9188×10^{1}	5.6561×10^{2}	1.6692×10^{1}	6.0525×10^{2}	1.0405×10^{0}
MGSMA	4.9273×10^{2}	$\mathbf{1.2200 \times 10^{1}}$	5.7690×10^{2}	1.9233×10^{1}	6.0234×10^{2}	1.2835×10^{0}
	F7		F8		F9	
	AVG	STD	AVG	STD	AVG	STD
CCRMIME	$\mathbf{7.8507 \times 10^{2}}$	$\mathbf{1.4574 \times 10^{1}}$	$\mathbf{8.6245 \times 10^{2}}$	1.5059×10^{1}	$\mathbf{9.0321 \times 10^{2}}$	$\mathbf{4.8120 \times 10^{0}}$
CDLOBA	2.6311×10^{3}	2.8591×10^{2}	1.1049×10^{3}	5.7198×10^{1}	9.7052×10^{3}	2.2021×10^{3}
GACO	9.0893×10^{2}	3.4676×10^{1}	8.7577×10^{2}	6.0635×10^{1}	9.3424×10^{2}	4.6311×10^{1}
HGWO	1.0481×10^{3}	2.0918×10^{1}	9.9848×10^{2}	$\mathbf{1.4387 \times 10^{1}}$	3.4898×10^{3}	4.5305×10^{2}
EWOA	9.6225×10^{2}	8.1269×10^{1}	9.5290×10^{2}	2.8131×10^{1}	4.9629×10^{3}	1.5459×10^{3}
CLSGMFO	9.0992×10^{2}	6.7021×10^{1}	9.3012×10^{2}	2.9225×10^{1}	3.4950×10^{3}	1.1974×10^{3}
LGCMFO	8.8148×10^{2}	5.0390×10^{1}	9.1856×10^{2}	2.6165×10^{1}	3.3518×10^{3}	1.2290×10^{3}
CGSCA	1.1520×10^{3}	3.5148×10^{1}	1.0593×10^{3}	1.7754×10^{1}	6.2451×10^{3}	1.3031×10^{3}
RDWOA	9.7258×10^{2}	6.3834×10^{1}	9.8154×10^{2}	4.0161×10^{1}	4.8079×10^{3}	1.6862×10^{3}
ACWOA	1.2478×10^{3}	5.2664×10^{1}	1.0078×10^{3}	2.3374×10^{1}	7.4632×10^{3}	1.1871×10^{3}
GCHHO	1.1016×10^{3}	9.2047×10^{1}	9.5053×10^{2}	2.6860×10^{1}	4.7696×10^{3}	6.6516×10^{2}
LSCA	8.3007×10^{2}	1.8972×10^{1}	8.7683×10^{2}	1.5135×10^{1}	1.1749×10^{3}	1.9735×10^{2}
MGSMA	8.2578×10^{2}	3.3544×10^{1}	8.8091×10^{2}	2.0732×10^{1}	1.1461×10^{3}	6.7840×10^{2}

Table A2. *Cont.*

	F10		F11		F12	
	AVG	STD	AVG	STD	AVG	STD
CCRMIME	$\mathbf{3.8117 \times 10^3}$	5.2835×10^2	$\mathbf{1.1633 \times 10^3}$	$\mathbf{2.8766 \times 10^1}$	$\mathbf{2.7691 \times 10^5}$	$\mathbf{1.9984 \times 10^5}$
CDLOBA	5.5379×10^3	5.7501×10^2	1.3083×10^3	5.8515×10^1	4.4602×10^5	3.1554×10^5
GACO	6.5253×10^3	1.9945×10^3	1.2229×10^3	4.8555×10^1	3.0329×10^5	1.0720×10^6
HGWO	6.6390×10^3	4.2684×10^2	4.6482×10^3	1.2339×10^3	5.6026×10^8	1.6895×10^8
EWOA	4.8231×10^3	4.7699×10^2	1.2233×10^3	4.1341×10^1	2.1335×10^6	1.4432×10^6
CLSGMFO	4.8368×10^3	5.6463×10^2	1.2365×10^3	7.6741×10^1	7.0704×10^5	7.4305×10^5
LGCMFO	4.6106×10^3	6.0161×10^2	1.2460×10^3	6.7025×10^1	7.8606×10^5	6.2101×10^5
CGSCA	8.0676×10^3	$\mathbf{2.9599 \times 10^2}$	2.3249×10^3	3.1854×10^2	1.4256×10^9	3.0369×10^8
RDWOA	5.0815×10^3	3.8035×10^2	1.2469×10^3	3.7450×10^1	3.3508×10^6	1.8090×10^6
ACWOA	6.4246×10^3	1.0186×10^3	3.1402×10^3	7.5725×10^2	7.2940×10^8	6.5601×10^8
GCHHO	5.1476×10^3	6.7906×10^2	1.2513×10^3	5.6921×10^1	8.8002×10^5	6.4579×10^5
LSCA	4.2380×10^3	5.4411×10^2	1.2222×10^3	2.9309×10^1	7.7729×10^6	1.1618×10^7
MGSMA	3.9371×10^3	6.7970×10^2	1.2016×10^3	4.2831×10^1	2.8029×10^6	1.9721×10^6
	F13		F14		F15	
	AVG	STD	AVG	STD	AVG	STD
CCRMIME	1.9203×10^4	1.8060×10^4	1.8427×10^4	1.7300×10^4	6.5933×10^3	9.1034×10^3
CDLOBA	1.4924×10^5	1.2217×10^5	$\mathbf{5.4402 \times 10^3}$	$\mathbf{3.6314 \times 10^3}$	1.3205×10^5	7.8172×10^4
GACO	2.8287×10^4	2.2608×10^4	4.2342×10^4	3.0497×10^4	1.8187×10^4	1.4375×10^4
HGWO	2.9173×10^8	1.4361×10^8	8.9393×10^5	6.3664×10^5	1.2588×10^7	1.4146×10^7
EWOA	1.8263×10^4	1.8810×10^4	4.7627×10^4	3.5783×10^4	1.4240×10^4	1.2351×10^4
CLSGMFO	2.0590×10^5	8.1130×10^5	4.7557×10^4	4.4149×10^4	9.9112×10^3	1.1563×10^4
LGCMFO	4.8387×10^4	3.7624×10^4	3.3747×10^4	3.3235×10^4	$\mathbf{6.3294 \times 10^3}$	$\mathbf{6.0796 \times 10^3}$
CGSCA	4.8903×10^8	1.6879×10^8	1.7939×10^5	1.3243×10^5	8.9434×10^6	7.9523×10^6
RDWOA	1.1308×10^4	$\mathbf{1.0248 \times 10^4}$	1.5974×10^5	1.9727×10^5	1.1445×10^4	1.0108×10^4
ACWOA	3.0391×10^7	2.2999×10^7	8.7553×10^5	7.0608×10^5	5.4222×10^6	3.7688×10^6
GCHHO	$\mathbf{1.0167 \times 10^4}$	1.1361×10^4	3.6069×10^4	2.8784×10^4	6.4598×10^3	8.2371×10^3
LSCA	2.2019×10^5	3.1553×10^5	4.3856×10^4	2.8626×10^4	5.2436×10^4	2.5261×10^4
MGSMA	4.8976×10^4	2.3486×10^4	9.5905×10^3	5.9505×10^3	1.2512×10^4	1.2589×10^4
	F16		F17		F18	
	AVG	STD	AVG	STD	AVG	STD
CCRMIME	2.1716×10^3	2.3029×10^2	$\mathbf{1.8274 \times 10^3}$	8.4514×10^1	1.2741×10^5	7.8150×10^4
CDLOBA	3.4241×10^3	5.0212×10^2	2.8982×10^3	3.7417×10^2	$\mathbf{1.2717 \times 10^5}$	$\mathbf{6.9361 \times 10^4}$
GACO	2.3329×10^3	4.8979×10^2	2.0020×10^3	1.7979×10^2	4.2048×10^5	3.1105×10^5
HGWO	3.4609×10^3	$\mathbf{1.7020 \times 10^2}$	2.4012×10^3	1.5799×10^2	1.8729×10^6	1.7400×10^6
EWOA	2.7465×10^3	2.1533×10^2	2.2874×10^3	1.8493×10^2	5.2977×10^5	5.0531×10^5
CLSGMFO	2.7489×10^3	3.0724×10^2	2.2256×10^3	2.5319×10^2	2.8972×10^5	3.4577×10^5
LGCMFO	2.7294×10^3	3.3492×10^2	2.2124×10^3	2.2146×10^2	2.1992×10^5	1.6580×10^5
CGSCA	3.7311×10^3	2.3719×10^2	2.5298×10^3	1.4165×10^2	3.4331×10^6	2.0187×10^6
RDWOA	2.7950×10^3	3.4434×10^2	2.2498×10^3	2.0527×10^2	5.5457×10^5	4.0176×10^5
ACWOA	3.9732×10^3	3.7958×10^2	2.5963×10^3	2.5811×10^2	2.2597×10^6	2.3980×10^6
GCHHO	2.7611×10^3	2.5749×10^2	2.3727×10^3	2.6181×10^2	3.0817×10^5	3.8868×10^5
LSCA	$\mathbf{2.1555 \times 10^3}$	2.0794×10^2	1.8621×10^3	$\mathbf{7.9278 \times 10^1}$	3.4819×10^5	2.0724×10^5
MGSMA	2.2591×10^3	2.8176×10^2	2.0462×10^3	1.9448×10^2	2.6570×10^5	2.0806×10^5

Table A2. *Cont.*

	F19		F20		F21	
	AVG	STD	AVG	STD	AVG	STD
CCRMIME	$\mathbf{5.7654 \times 10^3}$	$\mathbf{3.6009 \times 10^3}$	$\mathbf{2.1974 \times 10^3}$	$\mathbf{9.0494 \times 10^1}$	$\mathbf{2.3608 \times 10^3}$	1.6479×10^1
CDLOBA	6.4713×10^4	1.9239×10^4	2.9308×10^3	2.0193×10^2	2.6199×10^3	6.5362×10^1
GACO	2.1033×10^4	1.9265×10^4	2.2878×10^3	1.7765×10^2	2.3864×10^3	6.2962×10^1
HGWO	1.1124×10^7	1.3201×10^7	2.6869×10^3	1.2288×10^2	2.5142×10^3	1.3690×10^1
EWOA	1.2400×10^4	1.3969×10^4	2.5675×10^3	2.0623×10^2	2.4683×10^3	3.3169×10^1
CLSGMFO	8.8112×10^3	1.0998×10^4	2.5395×10^3	1.9434×10^2	2.4174×10^3	3.0286×10^1
LGCMFO	6.3336×10^3	4.2901×10^3	2.4549×10^3	1.8338×10^2	2.4048×10^3	4.9898×10^1
CGSCA	2.4060×10^7	1.4414×10^7	2.6037×10^3	1.4489×10^2	2.5680×10^3	1.4304×10^1
RDWOA	1.2841×10^4	1.5447×10^4	2.4676×10^3	1.6746×10^2	2.4996×10^3	5.3795×10^1
ACWOA	1.3257×10^7	2.2435×10^7	2.6273×10^3	1.6787×10^2	2.5858×10^3	3.5601×10^1
GCHHO	6.4727×10^3	3.6347×10^3	2.5354×10^3	1.7368×10^2	2.4884×10^3	3.1171×10^1
LSCA	9.9915×10^4	1.8824×10^5	2.2952×10^3	1.2181×10^2	2.3628×10^3	$\mathbf{1.2373 \times 10^1}$
MGSMA	1.8252×10^4	1.8883×10^4	2.3078×10^3	1.6238×10^2	2.3828×10^3	2.3238×10^1
	F22		F23		F24	
	AVG	STD	AVG	STD	AVG	STD
CCRMIME	$\mathbf{2.3002 \times 10^3}$	$\mathbf{7.5765 \times 10^{-1}}$	2.7146×10^3	1.8745×10^1	2.8912×10^3	2.1952×10^1
CDLOBA	7.2180×10^3	1.5423×10^3	3.2169×10^3	1.1733×10^2	3.2950×10^3	1.3644×10^2
GACO	6.8353×10^3	2.6363×10^3	2.7251×10^3	4.8884×10^1	2.9666×10^3	6.5059×10^1
HGWO	3.2650×10^3	2.7913×10^2	2.9044×10^3	$\mathbf{1.3808 \times 10^1}$	3.0679×10^3	2.3411×10^1
EWOA	5.3100×10^3	1.7810×10^3	2.8449×10^3	5.3846×10^1	3.0390×10^3	4.4861×10^1
CLSGMFO	2.3005×10^3	1.3387×10^0	2.7926×10^3	4.1302×10^1	2.9565×10^3	3.6295×10^1
LGCMFO	2.3008×10^3	1.3722×10^0	2.7723×10^3	3.7745×10^1	2.9293×10^3	2.4733×10^1
CGSCA	3.8366×10^3	2.2173×10^2	2.9923×10^3	2.8842×10^1	3.1492×10^3	2.9612×10^1
RDWOA	6.1073×10^3	1.3758×10^3	2.8667×10^3	4.8515×10^1	3.1539×10^3	9.6235×10^1
ACWOA	5.0314×10^3	2.3208×10^3	3.0489×10^3	8.3863×10^1	3.2190×10^3	7.8071×10^1
GCHHO	3.9292×10^3	2.0856×10^3	2.9362×10^3	6.4931×10^1	3.0966×10^3	8.5676×10^1
LSCA	5.4429×10^3	7.5623×10^2	$\mathbf{2.7116 \times 10^3}$	1.4187×10^1	$\mathbf{2.8746 \times 10^3}$	$\mathbf{9.7497 \times 10^0}$
MGSMA	4.1048×10^3	1.6632×10^3	2.7269×10^3	2.1013×10^1	2.8979×10^3	2.0308×10^1
	F25		F26		F27	
	AVG	STD	AVG	STD	AVG	STD
CCRMIME	2.8908×10^3	1.0318×10^1	4.2227×10^3	4.0093×10^2	3.2139×10^3	8.3073×10^0
CDLOBA	2.9245×10^3	3.4812×10^1	9.9323×10^3	1.3851×10^3	3.4762×10^3	1.2027×10^2
GACO	$\mathbf{2.8877 \times 10^3}$	$\mathbf{1.5572 \times 10^0}$	4.4404×10^3	5.0299×10^2	3.2215×10^3	1.1659×10^1
HGWO	3.0841×10^3	3.1846×10^1	6.0578×10^3	1.9583×10^2	3.3109×10^3	2.4109×10^1
EWOA	2.9056×10^3	2.2671×10^1	5.3412×10^3	8.7735×10^2	3.2479×10^3	2.2766×10^1
CLSGMFO	2.8937×10^3	1.7510×10^1	4.0969×10^3	1.3376×10^3	3.3108×10^3	7.3289×10^1
LGCMFO	2.8944×10^3	1.6987×10^1	$\mathbf{3.8900 \times 10^3}$	1.3627×10^3	3.2882×10^3	3.0538×10^1
CGSCA	3.2842×10^3	1.2604×10^2	6.8564×10^3	9.9596×10^2	3.3889×10^3	4.4544×10^1
RDWOA	2.9097×10^3	1.9742×10^1	5.6800×10^3	1.0261×10^3	3.2432×10^3	1.8363×10^1
ACWOA	3.1722×10^3	1.0847×10^2	7.4175×10^3	1.0116×10^3	3.4468×10^3	9.1493×10^1
GCHHO	2.8994×10^3	1.7365×10^1	5.5697×10^3	1.3955×10^3	3.2581×10^3	2.4280×10^1
LSCA	2.9172×10^3	1.6585×10^1	4.3309×10^3	$\mathbf{1.4284 \times 10^2}$	3.2098×10^3	$\mathbf{5.6140 \times 10^0}$
MGSMA	2.8886×10^3	1.0030×10^1	4.2849×10^3	4.5094×10^2	$\mathbf{3.2096 \times 10^3}$	1.0493×10^1

Table A2. *Cont.*

	F28		F29		F30	
	AVG	STD	AVG	STD	AVG	STD
CCRMIME	$\mathbf{3.1902 \times 10^3}$	5.4065×10^1	$\mathbf{3.4875 \times 10^3}$	1.2193×10^2	$\mathbf{8.5378 \times 10^3}$	$\mathbf{3.7952 \times 10^3}$
CDLOBA	3.2266×10^3	7.5174×10^1	5.1292×10^3	6.2273×10^2	1.9462×10^5	1.3876×10^5
GACO	3.2140×10^3	4.6149×10^1	3.6432×10^3	1.8583×10^2	1.2848×10^4	7.6514×10^3
HGWO	3.6098×10^3	3.5984×10^1	4.4660×10^3	1.5490×10^2	7.0910×10^7	3.7095×10^7
EWOA	3.2222×10^3	2.5979×10^1	4.0159×10^3	2.4819×10^2	2.4362×10^4	1.9051×10^4
CLSGMFO	3.2272×10^3	4.2567×10^1	3.8841×10^3	2.3364×10^2	7.9959×10^4	1.4662×10^5
LGCMFO	3.2173×10^3	2.5110×10^1	3.8060×10^3	2.4541×10^2	3.1086×10^4	7.0755×10^4
CGSCA	3.9139×10^3	1.4622×10^2	4.8157×10^3	2.1270×10^2	8.0984×10^7	2.5901×10^7
RDWOA	3.2605×10^3	2.4724×10^1	3.9396×10^3	2.3394×10^2	2.2020×10^4	2.0967×10^4
ACWOA	3.7377×10^3	1.9915×10^2	4.7696×10^3	3.5580×10^2	5.1487×10^7	2.7099×10^7
GCHHO	3.2150×10^3	2.1697×10^1	4.0339×10^3	2.3675×10^2	1.2335×10^4	5.3795×10^3
LSCA	3.2587×10^3	3.6188×10^1	3.5809×10^3	$\mathbf{8.9066 \times 10^1}$	9.9897×10^5	7.5200×10^5
MGSMA	3.2219×10^3	$\mathbf{1.8604 \times 10^1}$	3.5900×10^3	1.4622×10^2	3.9460×10^4	2.6819×10^4

References

1. Liu, W.; Zhou, H.; Zhang, S.; Zhao, C. Variable Parameter Creep Model Based on the Separation of Viscoelastic and Viscoplastic Deformations. *Rock Mech. Rock Eng.* **2023**, *56*, 4629–4645. [CrossRef]
2. Xu, Z.; Li, X.; Li, J.; Xue, Y.; Jiang, S.; Liu, L.; Luo, Q.; Wu, K.; Zhang, N.; Feng, Y. Characteristics of source rocks and genetic origins of natural gas in deep formations, Gudian Depression, Songliao Basin, NE China. *ACS Earth Space Chem.* **2022**, *6*, 1750–1771. [CrossRef]
3. Li, B.; Li, N.; Wang, E.; Li, X.; Zhang, Z.; Zhang, X.; Niu, Y. Discriminant Model of Coal Mining Microseismic and Blasting Signals Based on Waveform Characteristics. *Shock Vib.* **2017**, *2017*, 6059239. [CrossRef]
4. Yin, H.; Zhang, G.; Wu, Q.; Yin, S.; Soltanian, M.R.; Thanh, H.V.; Dai, Z. A Deep Learning-Based Data-Driven Approach for Predicting Mining Water Inrush From Coal Seam Floor Using Microseismic Monitoring Data. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 4504815. [CrossRef]
5. Kan, J.; Dou, L.; Li, J.; Song, S.; Zhou, K.; Cao, J.; Bai, J. Discrimination of microseismic events in coal mine using multifractal method and moment tensor inversion. *Fractal Fract.* **2022**, *6*, 361. [CrossRef]
6. Xiong, H.; Zhang, Z.; Yang, J.; Yin, Z.Y.; Chen, X. Role of inherent anisotropy in infiltration mechanism of suffusion with irregular granular skeletons. *Comput. Geotech.* **2023**, *162*, 105692. [CrossRef]
7. Zhao, J.; Wang, G.; Pan, J.-S.; Fan, T.; Lee, I. Density peaks clustering algorithm based on fuzzy and weighted shared neighbor for uneven density datasets. *Pattern Recognit.* **2023**, *139*, 109406. [CrossRef]
8. Lu, D.; Yue, Y.; Hu, Z.; Xu, M.; Tong, Y.; Ma, H. Effective detection of Alzheimer's disease by optimizing fuzzy K-nearest neighbors based on salp swarm algorithm. *Comput. Biol. Med.* **2023**, *159*, 106930. [CrossRef]
9. Liu, C.; Chen, W.; Zhang, T. Wavelet-Hilbert transform based bidirectional least squares grey transform and modified binary grey wolf optimization for the identification of epileptic EEGs. *Biocybern. Biomed. Eng.* **2023**, *43*, 442–462. [CrossRef]
10. Zhang, Q.; Huang, A.; Shao, L.; Wu, P.; Heidari, A.A.; Cai, Z.; Liang, G.; Chen, H.; Alotaibi, F.S.; Mafarja, M.; et al. A machine learning framework for identifying influenza pneumonia from bacterial pneumonia for medical decision making. *J. Comput. Sci.* **2022**, *65*, 101871. [CrossRef]
11. Hu, J.; Liu, Y.; Heidari, A.A.; Bano, Y.; Ibrohimov, A.; Liang, G.; Chen, H.; Chen, X.; Zaguia, A.; Turabieh, H. An effective model for predicting serum albumin level in hemodialysis patients. *Comput. Biol. Med.* **2022**, *140*, 105054. [CrossRef] [PubMed]
12. Xu, X.; Ding, S.; Wang, Y.; Wang, L.; Jia, W. A fast density peaks clustering algorithm with sparse search. *Inf. Sci.* **2021**, *554*, 61–83. [CrossRef]
13. Wan, M.; Chen, X.; Zhan, T.; Xu, C.; Yang, G.; Zhou, H. Sparse fuzzy two-dimensional discriminant local preserving projection (SF2DDLPP) for robust image feature extraction. *Inf. Sci.* **2021**, *563*, 1–15. [CrossRef]
14. Mohapatra, S.; Mohapatra, P. Fast random opposition-based learning Golden Jackal Optimization algorithm. *Knowl.-Based Syst.* **2023**, *275*, 110679. [CrossRef]
15. Li, S.; Chen, H.; Chen, Y.; Xiong, Y.; Song, Z. Hybrid Method with Parallel-Factor Theory, a Support Vector Machine, and Particle Filter Optimization for Intelligent Machinery Failure Identification. *Machines* **2023**, *11*, 837. [CrossRef]
16. Zhang, K.; Wang, Z.; Chen, G.; Zhang, L.; Yang, Y.; Yao, C.; Wang, J.; Yao, J. Training effective deep reinforcement learning agents for real-time life-cycle production optimization. *J. Pet. Sci. Eng.* **2022**, *208*, 109766. [CrossRef]
17. Xing, J.; Zhao, H.; Chen, H.; Deng, R.; Xiao, L. Boosting Whale Optimizer with Quasi-Oppositional Learning and Gaussian Barebone for Feature Selection and COVID-19 Image Segmentation. *J. Bionic Eng.* **2023**, *20*, 797–818. [CrossRef]

18. Yang, X.; Zhao, D.; Yu, F.; Heidari, A.A.; Bano, Y.; Ibrohimov, A.; Liu, Y.; Cai, Z.; Chen, H.; Chen, X. Boosted machine learning model for predicting intradialytic hypotension using serum biomarkers of nutrition. *Comput. Biol. Med.* **2022**, *147*, 105752. [CrossRef]
19. Chen, J.; Cai, Z.; Chen, H.; Chen, X.; Escorcia-Gutierrez, J.; Mansour, R.F.; Ragab, M. Renal Pathology Images Segmentation Based on Improved Cuckoo Search with Diffusion Mechanism and Adaptive Beta-Hill Climbing. *J. Bionic Eng.* **2023**, *20*, 2240–2275. [CrossRef]
20. Qi, A.; Zhao, D.; Yu, F.; Heidari, A.A.; Wu, Z.; Cai, Z.; Alenezi, F.; Mansour, R.F.; Chen, H.; Chen, M. Directional mutation and crossover boosted ant colony optimization with application to COVID-19 X-ray image segmentation. *Comput. Biol. Med.* **2022**, *148*, 105810. [CrossRef]
21. Dong, R.; Sun, L.; Ma, L.; Heidari, A.A.; Zhou, X.; Chen, H. Boosting Kernel Search Optimizer with Slime Mould Foraging Behavior for Combined Economic Emission Dispatch Problems. *J. Bionic Eng.* **2023**, 1–33. [CrossRef]
22. Dong, R.; Chen, H.; Heidari, A.A.; Turabieh, H.; Mafarja, M.; Wang, S. Boosted kernel search: Framework, analysis and case studies on the economic emission dispatch problem. *Knowl.-Based Syst.* **2021**, *233*, 107529. [CrossRef]
23. Deng, W.; Zhang, X.; Zhou, Y.; Liu, Y.; Zhou, X.; Chen, H.; Zhao, H. An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems. *Inf. Sci.* **2022**, *585*, 441–453. [CrossRef]
24. Hua, Y.; Liu, Q.; Hao, K.; Jin, Y. A Survey of Evolutionary Algorithms for Multi-Objective Optimization Problems With Irregular Pareto Fronts. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 303–318. [CrossRef]
25. Wu, S.-H.; Zhan, Z.-H.; Zhang, J. SAFE: Scale-adaptive fitness evaluation method for expensive optimization problems. *IEEE Trans. Evol. Comput.* **2021**, *25*, 478–491. [CrossRef]
26. Li, J.-Y.; Zhan, Z.-H.; Wang, C.; Jin, H.; Zhang, J. Boosting data-driven evolutionary algorithm with localized data generation. *IEEE Trans. Evol. Comput.* **2020**, *24*, 923–937. [CrossRef]
27. Fan, C.; Hu, K.; Feng, S.; Ye, J.; Fan, E. Heronian mean operators of linguistic neutrosophic multisets and their multiple attribute decision-making methods. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 1550147719843059. [CrossRef]
28. Cui, W.-H.; Ye, J. Logarithmic similarity measure of dynamic neutrosophic cubic sets and its application in medical diagnosis. *Comput. Ind.* **2019**, *111*, 198–206. [CrossRef]
29. Fan, C.; Fan, E.; Hu, K. New form of single valued neutrosophic uncertain linguistic variables aggregation operators for decision-making. *Cogn. Syst. Res.* **2018**, *52*, 1045–1055. [CrossRef]
30. Ye, J.; Cui, W. Modeling and stability analysis methods of neutrosophic transfer functions. *Soft Comput.* **2020**, *24*, 9039–9048. [CrossRef]
31. Liu, J.; Wei, J.; Heidari, A.A.; Kuang, F.; Zhang, S.; Gui, W.; Chen, H.; Pan, Z. Chaotic simulated annealing multi-verse optimization enhanced kernel extreme learning machine for medical diagnosis. *Comput. Biol. Med.* **2022**, *144*, 105356. [CrossRef] [PubMed]
32. Hu, J.; Han, z.; Heidari, A.A.; Shou, Y.; Ye, H.; Wang, L.; Huang, X.; Chen, H.; Chen, Y.; Wu, P. Detection of COVID-19 severity using blood gas analysis parameters and Harris hawks optimized extreme learning machine. *Comput. Biol. Med.* **2022**, *142*, 105166. [CrossRef] [PubMed]
33. Su, H.; Zhao, D.; Heidari, A.A.; Liu, L.; Zhang, X.; Mafarja, M.; Chen, H. RIME: A physics-based optimization. *Neurocomputing* **2023**, *532*, 183–214. [CrossRef]
34. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-Verse Optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [CrossRef]
35. Yang, X.-S. A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74.
36. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]
37. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume1944, pp. 1942–1948.
38. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]
39. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
40. Ahmadianfar, I.; Asghar Heidari, A.; Noshadian, S.; Chen, H.; Gandomi, A.H. INFO: An Efficient Optimization Algorithm based on Weighted Mean of Vectors. *Expert Syst. Appl.* **2022**, *195*, 116516. [CrossRef]
41. Ahmadianfar, I.; Asghar Heidari, A.; Gandomi, A.H.; Chu, X.; Chen, H. RUN Beyond the Metaphor: An Efficient Optimization Algorithm Based on Runge Kutta Method. *Expert Syst. Appl.* **2021**, *181*, 115079. [CrossRef]
42. Tu, J.; Chen, H.; Wang, M.; Gandomi, A.H. The Colony Predation Algorithm. *J. Bionic Eng.* **2021**, *18*, 674–710. [CrossRef]
43. Salimi, H. Stochastic Fractal Search: A powerful metaheuristic algorithm. *Knowl.-Based Syst.* **2015**, *75*, 1–18. [CrossRef]
44. Chen, H.; Li, C.; Mafarja, M.; Heidari, A.A.; Chen, Y.; Cai, Z. Slime mould algorithm: A comprehensive review of recent variants and applications. *Int. J. Syst. Sci.* **2022**, *54*, 204–235. [CrossRef]
45. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [CrossRef]
46. Yang, Y.; Chen, H.; Heidari, A.A.; Gandomi, A.H. Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Syst. Appl.* **2021**, *177*, 114864. [CrossRef]

47. Yong, J.; He, F.; Li, H.; Zhou, W. A Novel Bat Algorithm based on Collaborative and Dynamic Learning of Opposite Population. In Proceedings of the 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design (CSCWD), Nanjing, China, 9–11 May 2018; pp. 541–546.
48. Li, C.; Hou, L.; Pan, J.; Chen, H.; Cai, X.; Liang, G. Tuberculous pleural effusion prediction using ant colony optimizer with grade-based search assisted support vector machine. *Front. Neuroinform.* **2022**, *16*, 1078685. [CrossRef]
49. Zhu, A.; Xu, C.; Li, Z.; Wu, J.; Liu, Z. Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC. *J. Syst. Eng. Electron.* **2015**, *26*, 317–328. [CrossRef]
50. Tu, J.; Chen, H.; Liu, J.; Heidari, A.A.; Zhang, X.; Wang, M.; Ruby, R.; Pham, Q.-V. Evolutionary biogeography-based whale optimization methods with communication structure: Towards measuring the balance. *Knowl.-Based Syst.* **2021**, *212*, 106642. [CrossRef]
51. Xu, Y.; Chen, H.; Heidari, A.A.; Luo, J.; Zhang, Q.; Zhao, X.; Li, C. An Efficient Chaotic Mutative Moth-flame-inspired Optimizer for Global Optimization Tasks. *Expert Syst. Appl.* **2019**, *129*, 135–155. [CrossRef]
52. Xu, Y.; Chen, H.; Luo, J.; Zhang, Q.; Jiao, S.; Zhang, X. Enhanced Moth-flame optimizer with mutation strategy for global optimization. *Inf. Sci.* **2019**, *492*, 181–203. [CrossRef]
53. Kumar, N.; Hussain, I.; Singh, B.; Panigrahi, B.K. Single sensor-based MPPT of partially shaded PV system for battery charging by using cauchy and gaussian sine cosine optimization. *IEEE Trans. Energy Convers.* **2017**, *32*, 983–992. [CrossRef]
54. Chen, H.; Yang, C.; Heidari, A.A.; Zhao, X. An efficient double adaptive random spare reinforced whale optimization algorithm. *Expert Syst. Appl.* **2020**, *154*, 113018. [CrossRef]
55. Elhosseini, M.A.; Haikal, A.Y.; Badawy, M.; Khashan, N. Biped robot stability based on an A–C parametric Whale Optimization Algorithm. *J. Comput. Sci.* **2019**, *31*, 17–32. [CrossRef]
56. Song, S.; Wang, P.; Heidari, A.A.; Wang, M.; Zhao, X.; Chen, H.; He, W.; Xu, S. Dimension decided Harris hawks optimization with Gaussian mutation: Balance analysis and diversity patterns. *Knowl.-Based Syst.* **2021**, *215*, 106425. [CrossRef]
57. Wu, S.; Mao, P.; Li, R.; Cai, Z.; Heidari, A.A.; Xia, J.; Chen, H.; Mafarja, M.; Turabieh, H.; Chen, X. Evolving fuzzy k-nearest neighbors using an enhanced sine cosine algorithm: Case study of lupus nephritis. *Comput. Biol. Med.* **2021**, *135*, 104582. [CrossRef]
58. Ren, L.; Heidari, A.A.; Cai, Z.; Shao, Q.; Liang, G.; Chen, H.-L.; Pan, Z. Gaussian kernel probability-driven slime mould algorithm with new movement mechanism for multi-level image segmentation. *Measurement* **2022**, *192*, 110884. [CrossRef]
59. Jiang, R.; Dai, F.; Liu, Y.; Li, A. A novel method for automatic identification of rock fracture signals in microseismic monitoring. *Measurement* **2021**, *175*, 109129. [CrossRef]
60. Jiang, R.; Dai, F.; Liu, Y.; Wei, M. An automatic classification method for microseismic events and blasts during rock excavation of underground caverns. *Tunn. Undergr. Space Technol.* **2020**, *101*, 103425. [CrossRef]
61. Li, B.; Li, N.; Wang, E.; Li, X.; Niu, Y.; Zhang, X. Characteristics of coal mining microseismic and blasting signals at Qianqiu coal mine. *Environ. Earth Sci.* **2017**, *76*, 722. [CrossRef]
62. Li, X.; Li, Z.; Wang, E.; Liang, Y.; Li, B.; Chen, P.; Liu, Y. Pattern recognition of mine microseismic and blasting events based on wave fractal features. *Fractals-Complex Geom. Patterns Scaling Nat. Soc.* **2018**, *26*, 1850029. [CrossRef]
63. Peng, P.; He, Z.; Wang, L. Automatic Classification of Microseismic Signals Based on MFCC and GMM-HMM in Underground Mines. *Shock Vib.* **2019**, *2019*, 5803184. [CrossRef]
64. Shan, W.; Qiao, Z.; Heidari, A.A.; Gui, W.; Chen, H.; Teng, Y.; Liang, Y.; Lv, T. An efficient rotational direction heap-based optimization with orthogonal structure for medical diagnosis. *Comput. Biol. Med.* **2022**, *146*, 105563. [CrossRef] [PubMed]
65. Hu, J.; Heidari, A.A.; Zhang, L.; Xue, X.; Gui, W.; Chen, H.; Pan, Z. Chaotic diffusion-limited aggregation enhanced grey wolf optimizer: Insights, analysis, binarization, and feature selection (Intelligent Systems, impact factor:8.709). *Int. J. Intell. Syst.* **2021**, *37*, 4864–4927. [CrossRef]
66. Venkata Rao, R. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int. J. Ind. Eng. Comput.* **2016**, *7*, 19–34. [CrossRef]
67. Askari, Q.; Younas, I.; Saeed, M. Political Optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowl.-Based Syst.* **2020**, *195*, 105709. [CrossRef]
68. Eddaly, M.; Jarboui, B.; Siarry, P. Combinatorial particle swarm optimization for solving blocking flowshop scheduling problem. *J. Comput. Des. Eng.* **2016**, *3*, 295–311. [CrossRef]
69. Hu, K.; Ye, J.; Fan, E.; Shen, S.; Huang, L.; Pi, J. A novel object tracking algorithm by fusing color and depth information based on single valued neutrosophic cross-entropy. *J. Intell. Fuzzy Syst.* **2017**, *32*, 1775–1786. [CrossRef]
70. Liang, Z.; Zhang, J.; Feng, L.; Zhu, Z. A hybrid of genetic transform and hyper-rectangle search strategies for evolutionary multi-tasking. *Expert Syst. Appl.* **2019**, *138*, 112798. [CrossRef]
71. Yu, H.; Cheng, X.; Chen, C.; Heidari, A.A.; Liu, J.; Cai, Z.; Chen, H. Apple leaf disease recognition method with improved residual network. *Multimed. Tools Appl.* **2022**, *81*, 7759–7782. [CrossRef]
72. Zhang, H.; Liu, T.; Ye, X.; Heidari, A.A.; Liang, G.; Chen, H.; Pan, Z. Differential evolution-assisted salp swarm algorithm with chaotic structure for real-world problems. *Eng. Comput.* **2022**, *39*, 1735–1769. [CrossRef]
73. Qiao, K.; Liang, J.; Yu, K.; Yuan, M.; Qu, B.; Yue, C. Self-adaptive resources allocation-based differential evolution for constrained evolutionary optimization. *Knowl.-Based Syst.* **2022**, *235*, 107653. [CrossRef]

74. Liang, J.; Ban, X.; Yu, K.; Qu, B.; Qiao, K. Differential evolution with rankings-based fitness function for constrained optimization problems. *Appl. Soft Comput.* **2021**, *113*, 108016. [CrossRef]
75. Liu, Q.Y.; Li, D.Q.; Tang, X.S.; Du, W. Predictive Models for Seismic Source Parameters Based on Machine Learning and General Orthogonal Regression Approaches. *Bull. Seismol. Soc. Am.* **2023**. [CrossRef]
76. Tie, Y.; Rui, X.; Shi-Hui, S.; Zhao-Kai, H.; Jin-Yu, F. A real-time intelligent lithology identification method based on a dynamic felling strategy weighted random forest algorithm. *Pet. Sci.* 2023, *in press*. [CrossRef]

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