Swarm and Evolutionary Computation 97 (2025) 102023

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

An improved genetic algorithm and clone selection optimization-based gated recurrent unit networks for earthquake magnitude prediction

Wen Zhou [a], Xinchun Yi [a], Changyi Li [a], Zhiwei Ye [a,*], Qiyi He [a], Xiuwen Gong [b], Qiao Lin [c]

[a] School of Computer Science, Hubei University of Technology, Wuhan, 430068, China
[b] School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia
[c] School of Computer Science, University of Nottingham, Nottingham, NG7 2RD, United Kingdom

ARTICLE INFO

Dataset link: https://github.com/123fggv/experimental-data

Keywords:
Earthquake prediction
Genetic algorithm
Gated recurrent units
Clone selection optimization
AETA

ABSTRACT

Earthquake magnitude prediction is a vital rendezvous for human safety, economic and property losses. The earthquake occurrence process represents a highly complex nonlinear problem. Meanwhile, artificial intelligence methods have emerged as automated and intelligent frameworks for addressing magnitude prediction challenges. However, these approaches ignore redundant features and have lower prediction accuracy. Genetic Algorithms (GA) excel in feature selection and Gated Recurrent Units (GRU) have strong time series prediction capabilities. Therefore, we propose a novel earthquake magnitude prediction method, named Improved GA and a Clone Selection Optimization-based GRU (IGA-CSOGRU). First, an improved GA with generation gap strategy is presented to enhance the feature selection capability of time-series data in prediction models. Second, GRU is implemented as the core prediction model. To optimize its hyperparameters, a novel approach combining Latin hypercube sampling with adaptive mutation CSO is introduced, thereby enhancing prediction performance. Finally, to validate the performance of the proposed IGA-CSOGRU, a novel earthquake magnitude prediction dataset is constructed, which is acquired from the self-developed Acoustic & Electromagnetics to AI (AETA) platform. Evaluation metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 were used for assessment. The proposed IGA-CSOGRU model demonstrates significant performance improvements across all datasets, achieving an average RMSE reduction of 5%–7% compared to all baseline methods, highlighting the model's superior capability in handling challenging time series prediction tasks. The implementation code supporting the findings of this study is available at https://github.com/123fggv/Earthquake-prediction.

1. Introduction

The earthquake is among the worst natural disasters and can cause other natural disasters, such as floods and tsunamis [1,2]. The global earthquake data demonstrate that there are more than 5 million earthquakes every year according to crustal movement and other reasons, and tens of thousands of earthquakes every day. Among these earthquake events, although the vast majority of events are too small on the Richter scale for people to feel, the medium–large earthquakes can be fatal and result in severe economic and material losses. Earthquakes are a force of nature with a strong randomness; therefore, earthquake magnitude prediction has always been a difficult problem. Meanwhile, improving the accuracy of earthquake magnitude prediction is an important research topic.

Earthquake magnitude prediction can be broadly divided into two categories, (1) seismic precursors-based methods; and (2) historic earthquake data-based methods [3]. This paper focuses on the former one.

For seismic precursor-based tasks, each instance is associated with the observed earthquake signals, and this task always has large-scale seismic data. Therefore, many artificial intelligence methods have been used to predict the magnitude of earthquakes. Berchich et al. [4] proposed an earthquake magnitude prediction method based on Recurrent Neural Network (RNN) algorithms, employing Long-Short-Term Memory (LSTM), GRU, and their hybrid model LSTM-GRU for earthquake prediction. This approach takes advantage of the strong long-term memory capabilities of LSTM and GRU, which can capture long-term trends and patterns that precede earthquakes, thus reducing overfitting on training data. In the following, Berchich et al. [5] introduced an attention-based LSTM network to predict the magnitudes of large earthquakes. This method demonstrates superior capability in capturing critical information from long-sequence data, effectively managing sequences of varying lengths and complexities. To deal with the massive

* Corresponding author.
E-mail address: hgcsyzw@hbut.edu.cn (Z. Ye).

https://doi.org/10.1016/j.swevo.2025.102023
Received 14 February 2025; Received in revised form 20 May 2025; Accepted 29 May 2025
Available online 20 June 2025

amount of seismic precursors, Ye et al. [6] presented an Elite Genetic Algorithm-based LSTM (EGA-LSTM) for earthquake prediction. This method introduces the EGA for feature selection to address feature redundancy in earthquake magnitude prediction. By selecting important features, model performance is significantly improved. Furthermore, the EGA exhibits a fast convergence rate, allowing it to find optimal parameter configurations in a relatively short time, thus accelerating the training process of the earthquake prediction model. Unfortunately, these methods suffer from three major limitations.

- **Low accuracy**: The performance of training models and redundant seismic data limits the accuracy of earthquake magnitude methods, especially for large-scale seismic data. For instance, the feature selection method EGA may converge prematurely to a certain local optimal solution, thus losing the opportunity to further search for the global optimal solution. This is usually caused by the selection mechanism in the algorithm being overly inclined to retain excellent individuals, resulting in a reduction in population diversity.
- **Low generalization ability**: The GRU model exhibits a high dependence on data quality and hyperparameter tuning, resulting in significant variations and instability in prediction performance across different variants of seismic datasets and hyperparameters.
- **High costs and poor efficiency**: The complexity of GRU leads to high computational costs for training, especially when dealing with large-scale seismic data, resulting in poor prediction efficiency.

Hence, it is nontrivial to solve the above-mentioned drawbacks by focusing on two aspects: improving the quality of seismic data and proposing an effective earthquake magnitude prediction method.

This study systematically studies how to efficiently obtain optimal features and learn the appropriate hyperparameters to solve the earthquake magnitude prediction task. To achieve our goal, we propose a novel Improved Genetic Algorithm and a Clone Selection Optimization-based Gated Recurrent Unit network (IGA-CSOGRU) for earthquake magnitude prediction. In addition, the generation gap strategy is introduced to IGA. This process first randomly selects a parent individual and checks the diversity on the basis of the Hamming distance between the randomly selected individual and the current partner, and the generation gap strategy can get population update, thus enhancing the effectiveness of feature selection. This is particularly beneficial when dealing with the complex and large datasets commonly found in earthquake data, addressing the issue where spatio-temporal models may fail to capture the significant features and potentially overlook critical ones [7].

Recall that CSO has shown good performance in hyperparameter optimization [8]. Motivated by this, we propose a novel CSOGRU to obtain the optimal hyperparameter. GRU has a simpler structure with fewer parameters compared to LSTM, which will be more conducive to model training. An improved CSO was introduced to perform GRU hyperparameter optimization. Latin Hypercube Sampling (LHS) is applied for CSO population initialization to generate uniformly distributed samples that address the issue of nonuniform sampling and susceptibility to local optima in the original CSO. This approach enhances the diversity of the population and enables a more comprehensive exploration of the solution space, thereby reducing the likelihood of becoming trapped in local optima. Additionally, the fixed mutation rate in the original CSO leads to ineffective exploration of the search space in the early stages and lacks fine-grained local search capabilities when approaching the global optimum. To address this issue, we propose an adaptive mutation strategy that adaptively adjusts the mutation rate at different search stages. That is, for poorer solutions, a larger mutation rate is given to increase exploration capabilities, while for better solutions, a smaller mutation rate is applied for fine-tuning. This improves the local search capabilities of the algorithm and the accuracy of the prediction.

Lastly, a novel earthquake magnitude prediction dataset is constructed to validate the performance of the proposed seismic magnitude prediction method, which is based on our self-developed AETA and China Earthquake Networks Center (CENC). In this study, we propose a novel approach that combines IGA feature selection with a GRU model, optimized through the CSO algorithm. This method has been evaluated with various Evolutionary Algorithms (EAs) and Machine Learning (ML) techniques as benchmarks, and four ML evaluation metrics are implemented to rigorously assess the performance of our approach. Experimental results demonstrate that the proposed IGA-CSOGRU framework significantly exceeds the performance of state-of-the-art methods.

The main contributions are as follows.

- A seismic magnitude dataset is constructed. The original dataset was collected by the AETA system and CENC, and our self-developed AETA system is shown in Fig. 1, including seismic instances in China and surrounding areas by detecting electromagnetic and acoustic waves.
- For feature selection task, we propose an IGA methodology using the generation gap strategy. This approach ensures that effective features are better extracted, thus improving the accuracy of earthquake prediction.
- Considering the influence of correctly setting the parameters of the GRU model on the accuracy of seismic prediction, a novel CSO with LHS and adaptive mutation strategy is presented for hyperparameter optimization of the GRU. Therefore, this method can reduce the difficulty of manually adjusting the parameters and improve the generalizability of the GRU model.

The remainder of the study is organized as follows. Section 2 presents the related work. The proposed method is described in Section 3. In Section 4, the IGA-CSOGRU related experiments are described in detail. Conclusions and future work are shown in Section 5.

2. Related work

Numerous researchers have conducted extensive studies on earthquake magnitude prediction tasks, including machine learning-based methods and deep learning-based methods [9,10].

Machine learning-based earthquake prediction approaches are data-driven and non-parametric, requiring fewer prior assumptions [11]. Hence, machine learning has been widely implemented in seismological problems [12,13]. Ikram and Qamar [14] proposed an expert system to predict earthquakes from previous data, which is achieved by applying association rule mining on earthquake data from 1972 to 2013. These associations are polished using predicate-logic techniques to draw stimulating production rules to be used with a rule-based expert system. Moustra et al. [15] used Seismic Electric Signals (SES) to evaluate the performance of artificial neural networks in predicting earthquakes occurring in the region of Greece with the use of different types of input data. Rouet-Leduc et al. [16] applied machine learning to datasets from shear laboratory experiments to identify hidden signals that precede earthquakes, thus predicting earthquake magnitudes. Hoque et al. [17] used artificial neural networks as a prediction model, which have provided accurate earthquake predictions for multiple regions. Salam et al. [18] proposed two hybrid machine learning models for earthquake magnitude prediction. The first model, FPA-ELM, combines the Flower Pollination Algorithm (FPA) with the Extreme Learning Machine (ELM). The second model, FPA-LS-SVM, integrates FPA with the Least Squares Support Vector Machine (LS-SVM). The experimental results demonstrated that the proposed models outperform traditional machine learning models in prediction accuracy. Joshi et al. [19] introduced a novel SeisEML model for earthquake prediction. This model integrates hybrid models, kernel regression algorithms, tree regression algorithms, and regression algorithms. Experimental results indicate that SeisEML significantly enhances the

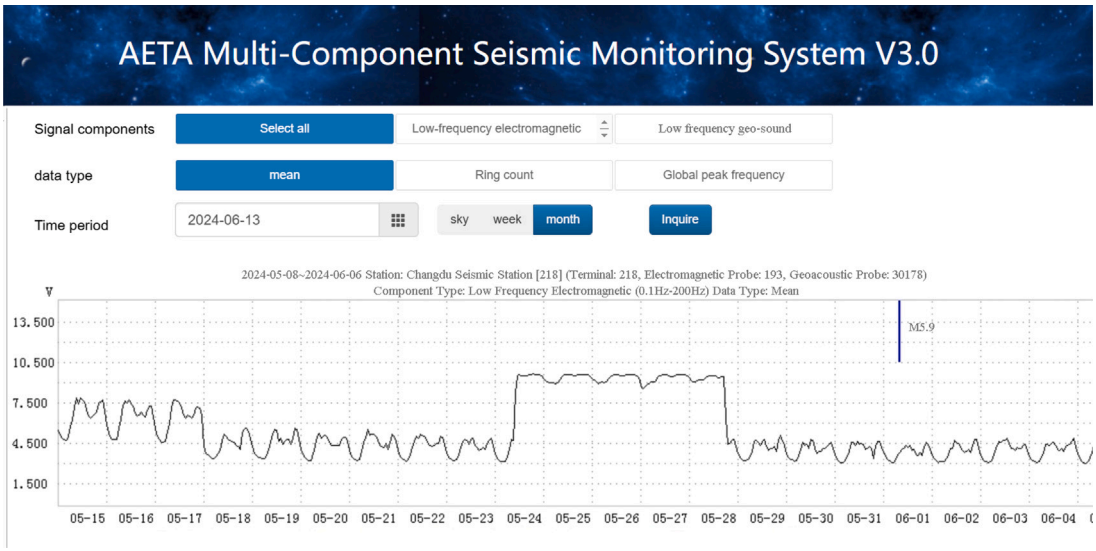


Fig. 1. The sampled signals in AETA.

prediction of peak ground acceleration, thus improving the accuracy of earthquake forecasts. Kubo et al. [20] proposed a hybrid approach combining machine learning and conventional ground motion prediction equations for earthquake prediction. However, its effectiveness is highly dependent on data quality and feature selection, and it exhibits significant limitations in processing time-series data. Zhang et al. [21] introduced a methodology that involves clustering microseismic data prior to model construction, followed by model training to improve prediction accuracy. Experimental results indicate that spatiotemporal clustering analysis of microseismic events significantly improves the model's prediction accuracy. Yuan [22] adopted an earthquake prediction model based on the clustering of global seismic data, utilizing the Space-Time-Magnitude (STM) distance. It has shown good performance in earthquake prediction models. These approaches can accurately predict earthquakes in relevant areas and do not require a lot of computing resources to perform well on small datasets. However, their effectiveness largely depends on the quality of the data and feature selection. Furthermore, these models may exhibit insufficient generalization capabilities when applied to new and unseen data.

Deep learning algorithms have recently made significant progress in solving a wide range of earthquake prediction problems [12,23]. Bao et al. [24] proposed a Convolutional Neural Network (CNN) and designed a three-dimensional feature map that leverages the advantages of both shallow features and high-dimensional information. Experimental results indicate that this method effectively distinguishes seismic magnitudes and improves the accuracy of magnitude prediction to a certain extent. Asim et al. [25] adopted neural networks, recurrent neural networks, random forests, and linear programming boost ensemble classifiers to predict earthquakes. These approaches yielded significant predictive results. To extract spatiotemporal features from seismic data and enhance the feature extraction capability of prediction models for temporal data, Zhang et al. [26] proposed a convolution-based LSTM model for global spatiotemporal earthquake prediction. The results demonstrate that this method effectively predicts earthquakes with higher resolution and accuracy. Vardaan et al. [27] used an RNN with LSTM to model seismic sequences and used the trained model for earthquake prediction. The experimental results indicate that the proposed method significantly improves performance compared to traditional deep learning approaches. Puthran [28] proposed a hybrid seismic prediction model based on a combination of CNN and GRU for spatio-temporal analysis. The results show that this method has excellent superiority in predicting major earthquake events, achieving an accuracy of up to 98.67%. These approaches outperform traditional prediction methods and have become the most mainstream approach currently. However, the large volume of seismic data and the difficulty in tuning the hyperparameters of the prediction models can affect the accuracy of seismic predictions.

Machine learning and deep learning based works for earthquake prediction are summarized in Table 1. In earthquake prediction tasks, traditional approaches like association rule mining [14] and basic neural networks [15,17] provide valuable frameworks but heavily depend on historical patterns and data enhancement. More advanced hybrid models like FPA-ELM [18] and SeisEML [19] improved prediction accuracy but still suffered from data quality dependencies. Recent deep learning approaches using CNN [24], RNN [25,27], and hybrid architectures like CNN-GRU [28] have shown improved spatiotemporal feature extraction and generalization capabilities, however they face challenges in hyperparameter tuning and interpretability. Our proposed IGA-CSOGRU model addresses some of these limitations by optimizing redundant feature information and achieving improved RMSE on self-collected earthquake datasets, though interpretability remains a challenge. To improve the accuracy of earthquake magnitude prediction approaches, it is essential to address issues such as feature redundancy, time-series data selection, and hyperparameter tuning. Therefore, we propose an IGA-CSOGRU model that effectively addresses these challenges and significantly improves the accuracy of earthquake magnitude predictions. In addition, the IGA is implemented to enhance the feature extraction capability of time-series data in prediction models. The GRU is adopted as the prediction model, and the CSO is used for hyperparameter optimization of the GRU, to improve the prediction performance.

3. Approach

In this section, the proposed IGA-CSOGRU earthquake magnitude prediction model is described in detail. The model combines IGA and the CSOGRU model to improve the accuracy and efficiency of earthquake prediction. IGA is used to optimize the selection of input features, reduce redundancy, and improve the quality of seismic data. Meanwhile, the CSOGRU module is able to effectively capture time-dependent and complex patterns in time-series data, which are crucial for accurate earthquake magnitude prediction. Additionally, during the iterative processes of both IGA and CSO algorithms, individual fitness values serve as the evaluation metric for solution quality. Lower fitness values indicate higher solution quality and closer proximity to the optimal solution of the optimization problem. This inverse relationship between fitness values and solution quality aligns with our research objective of minimizing prediction error, where lower fitness values correspond to higher prediction accuracy of the model.

Table 1
Summary of machine learning and deep learning based works.

Work	Proposed approach	Results	Merits	Limitations
[14]	Association rule mining with predicate-logic	Earthquake occurrence prediction	- Production rules were improved - System reliability was increased	- Required the data enhancement - High dependency on historical patterns
[15]	Artificial Neural Networks with SES	Earthquake occurrence prediction	- SES data utilization was improved - Strong generalization	- Required the data enhancement - Sensitive to signal quality
[16]	ML on shear laboratory experiments	Earthquake magnitude prediction	- Signal detection was enhanced - Precursor identification was increased	- Required the data enhancement
[17]	Artificial neural networks	Earthquake occurrence prediction	- Multi-region accuracy was improved - Model generalization was enhanced	- Required the data enhancement - Network structure optimization needed
[18]	FPA-ELM and FPA-LS-SVM hybrid models	Earthquake magnitude prediction	- Hybrid model accuracy was improved - Optimization efficiency was increased	- Required the data enhancement
[19]	SeisEML hybrid model	Peak ground acceleration prediction	- Peak ground acceleration accuracy was improved - Model integration efficiency was enhanced	- Required the data enhancement - Parameter sensitivity
[20]	Hybrid ML with ground motion equations	Ground motion prediction	- Traditional method integration was improved - Ground motion accuracy was enhanced	- Data quality dependent - Limited feature selection
[21]	Clustering-based ML with microseismic data	Ground motion prediction	- Spatiotemporal analysis was improved - Clustering efficiency was enhanced	-Clustering quality dependent
[22]	STM distance-based clustering	Earthquake occurrence prediction	- STM feature utilization was improved - Clustering performance was enhanced - Global data analysis was increased	- Required the data enhancement - Distance metric sensitivity
[24]	CNN with 3D feature map	Earthquake magnitude prediction	- 3D feature extraction was improved - Feature map efficiency was enhanced	- Difficulty of hyperparameter adjustment
[25]	Neural networks, RNN, random forests	Earthquake occurrence prediction	- Multiple classifier performance was improved - Strong generalization	- Difficulty of hyperparameter adjustment
[26]	Convolution-based LSTM	Earthquake magnitude prediction	- Feature extraction was improved - Global prediction accuracy was increased	- Required the data enhancement
[27]	RNN with LSTM	Earthquake magnitude prediction	- Temporal sequence modeling was improved	- Difficulty of hyperparameter adjustment
[28]	Hybrid CNN-GRU	Earthquake magnitude prediction	- Major event accuracy was improved - Strong generalization	- Difficulty of hyperparameter adjustment - Non-interpretability
Ours	IGA-CSOGRU	Earthquake magnitude prediction	- RMSE improvement - Optimizing redundant feature information - Training is conducted using self-collected and processed Earthquake datasets	- Non-interpretability

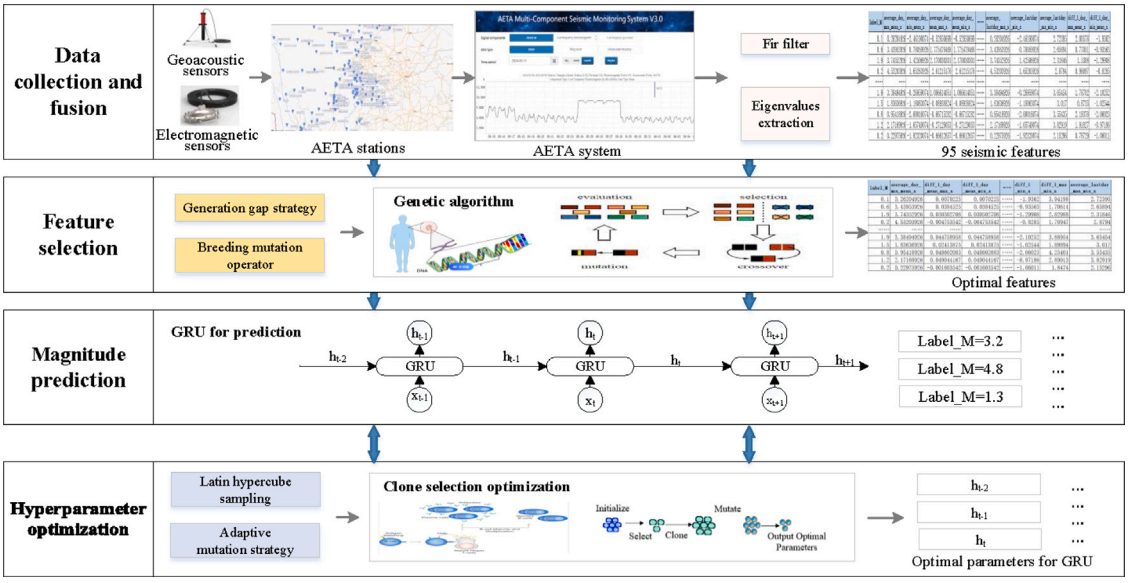


Fig. 2. The overall structure of IGA-CSOGRU.

Algorithm 1 IGA-CSOGRU

Input: Fusion of AETA seismic data X
Output: Magnitude of next day

Maxiter-IGA: The maximum number of iterations of the IGA algorithm
Maxiter-CSO: The maximum number of iterations of the CSO algorithm
$\bar{X}$: The selected optimal feature subset
H: Hyperparameters of GRU model
$\bar{H}$: Optimized GRU model hyperparameters
Initialize population and parameters of IGA
while $t <$ Maxiter-IGA **do**
  Calculate fitness for each individual
  for each individual do
    Roulette selection operation
    Two-point crossover operation
    Binary mutation operator operation
  end for
  Generate new population with generation gap strategy
end while
$\bar{X}$ = best fitness individual
Initialize CSO population by Hypercube Initialization (LHS), Initialize CSO parameters
while $t <$ Maxiter-CSO **do**
  Calculate fitness for each individual
  for each individual do
    Selection operation
    Clone operation
    Perform mutation operations based on the mutation rate dynamically adjusted by fitness
  end for
end while
$\bar{H} = \mathrm{CSO}(H)$
Train GRU using hyperparameters $\bar{H}$
$M = \mathrm{GRU}(\bar{X})$
return M

3.1. Proposed method

The overall structure of the proposed method is shown in Fig. 2. It consists of four basic modules: the data collection and fusion module, the feature extraction module, the seismic prediction module, and the hyperparameter optimization module. The data collection and fusion module uses the data from our self-development AETA system, which is acquired from our self-development geoacoustic sensors and electromagnetic sensors. The feature selection module primarily processes seismic data, removes redundant information, and provides selected features as input to the earthquake prediction module. The seismic prediction module includes a GRU for earthquake magnitude prediction. The hyperparameter optimization phase includes a CSO mechanism for the hyperparameter tuning of the GRU to improve the accuracy of the prediction. The pseudocode of IGA-CSOGRU is described as Algorithm 1.

3.2. Data collection and fusion

Our team has deployed 300 sets of self-designed geoacoustic sensors and electromagnetic sensors around the world. The AETA is a platform to show the 12 indicators acquired from these sensors, including full-frequency electromagnetic (30 kHz), low-frequency electromagnetic (500 Hz), full-frequency ground sound (150 kHz) and low-frequency ground sound (500 Hz). Each type of signal contains 3 indicators: mean, ring count, and peak frequency. The AETA time series [29] data are a set of observation samples $X = (x_1, x_2, \ldots, x_i, \ldots, x_m)$ collected at equal intervals in the time sequence, where $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,j}, \ldots, x_{i,n})$, $x_{i,j}$ represents the observation value of the observation component jth at the ith time point, x_i is a vector of dimensions n. The observed feature is described as F, $F = (F_1, F_2, \ldots, F_j, \ldots, F_n)$, $F_j = (x_{1,j}, x_{2,j}, \ldots, x_{i,j}, \ldots, x_{m,j})$. Therefore, the m observation samples and n observation features can be more intuitively expressed as a matrix in Eq. (1), m is the number of observed samples, n indicates the number of features.

$$\begin{bmatrix} 1 & x_{1,1} & \ldots & x_{1,j} & \ldots & x_{1,n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ i & x_{i,1} & \ldots & x_{i,j} & \ldots & x_{i,n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \\ m & x_{m,1} & \ldots & x_{m,j} & \ldots & x_{m,n} \end{bmatrix} \tag{1}$$

The Finite Impulse Response (FIR) filter and eigenvalues extraction technology are implemented to fuse the observed signals, and 95 features can be observed, 51 features for electromagnetic signals, and 44 features for acoustic signals. However, MW is not detected in AETA. Hence, the Moment Magnitude scale (MW) data are chosen from CENC. As a result, we merge 95 features and MW in a given time. The time interval is short, and the MW prediction in a short period makes less sense. Hence, we select a typical value in each feature to represent a whole day. This means that the size of the original one-day data is 144×96. After the fusion phase, the size becomes 1×96. The fused seismic samples are described as $\hat{X} = (num, x_1, x_2, \ldots, x_i, \ldots, x_m, MW)$, where the num is the number of each sample, and the seismic samples are indicated as Eq. .

$$\begin{bmatrix} 1 & x_{1,1} & \ldots & x_{1,j} & \ldots & x_{1,95} & MW_1 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ i & x_{i,1} & \ldots & x_{i,j} & \ldots & x_{i,95} & MW_i \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \\ m & x_{m,1} & \ldots & x_{m,j} & \ldots & x_{m,95} & MW_m \end{bmatrix} \tag{2}$$

3.3. Feature selection for IGA

Although EGA has been successfully introduced in feature selection, it still performs poorly in the handling of time-series data. Therefore, in the IGA, a generation gap strategy is applied in the population update phase to enhance the feature selection capability of time-series data in prediction models. The IGA framework is described in Fig. 3. The generation gap strategy and diversity maintenance strategy of IGA make it robust and able to consistently find high-quality feature subsets when faced with different types of feature selection problems. When selecting seismic features, the process typically includes the following steps: initializing the population, selecting parents, crossover, mutating operation, selecting children to replace the generation of the

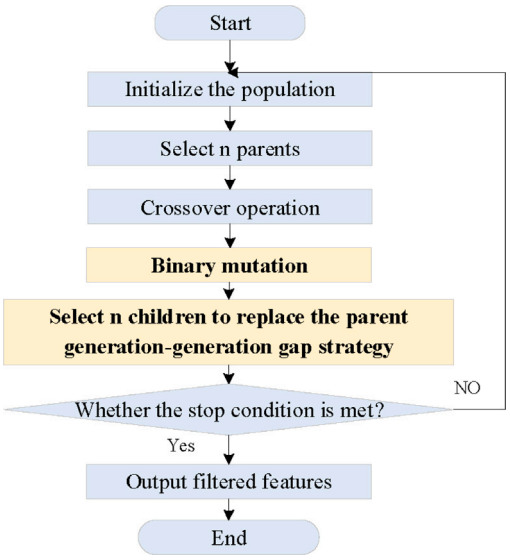


Fig. 3. The framework of IGA.

parents with generation gap strategy, and repeating these steps until the predetermined stopping criteria are met. The final output is the optimal feature subset $\overline{X} = (num, x_1, x_2, \ldots, x_i, \ldots, x_m, MW)$. Crossover is the core of the algorithm. This process first randomly selects a parent individual and checks the diversity on the basis of the Hamming distance between the randomly selected individual and the current partner. If the diversity exceeds a preset threshold, a crossover is performed to produce offspring; otherwise, the current partner is mutated to produce offspring.

In addition, the specific IGA-based seismic feature selection method is shown in Fig. 4. First, the generated seismic feature data is input, then IGA is used for feature selection. The selected feature subset is then input into the GRU prediction model for validation. The RMSE is calculated based on the actual and predicted values output by the GRU model, and then incorporated into Eq. (3) to calculate the fitness value,

$$fg = RMSE * \omega_\alpha + \frac{Selected\,Features}{Total\,features} * \omega_F \tag{3}$$

where ω_α represents the weight controlling the prediction error, ω_F represents the weight controlling the penalty for the number of selected features, "Selected Features" refers to the number of selected features and "Total Features" refers to the total number of available features. Then this fitness value is fed back into the IGA feature selection process. The algorithm terminates and outputs the final feature subset when either the fitness value remains unchanged for 10 consecutive iterations or the maximum iteration count is reached. The final feature subset is then mapped back to the original seismic dataset to form the ultimate seismic feature data, where 1 represents selected seismic data features and 0 indicates redundant seismic data features.

Initialization strategy: Generate the initial population $Population = (p_1, p_2, \ldots, p_i, \ldots, p_N)$, the chromosome is described as p_i, $p_i = (s_{F_1}, s_{F_2}, \ldots, s_{F_i}, \ldots, s_{F_{96}}), s_{F_i} \in [0, 1]$. The s_{F_i} equal to 1 means that the feature F_i is selected; otherwise, it is not selected. The individual encoding rule is binary encoding, and the population generation method is a random method.

Genetic strategy: Roulette selection is used to select individuals with higher fitness to generate offspring; the two-point crossover strategy and binary mutation operator are adopted to generate offspring.

Generation gap strategy: The Generation gap strategy was incorporated into the traditional GA for preserving historical optimal solutions while introducing new genetic information. In this strategy, we determine how many offspring replace the worst individuals in the parent population based on the preset generation gap rate g. After completing crossover and mutation operations, through the generation gap strategy, a portion of the parent population is replaced by offspring. According to the preset generation gap ratio g, high-quality offspring with lower fitness values replace individuals with higher fitness values in the parent population to generate a new population. This approach not only preserves historical optimal solutions but also introduces new optimal individuals, maximizing the algorithm's convergence speed and population diversity. This can be specifically expressed as Eq. (4),

$$\overline{fg}^{t+1} = g \times \overline{fg}_S^{t+1} + (1-g) \times \overline{fg}_s urv^{t+1} \tag{4}$$

where fg_S^{t+1} denotes the expected fitness of the selected individuals, $fg_s urv^{t+1}$ indicates the expected fitness of the survivors.

Finally, the optimal fitness features are selected, which is described as $\overline{X} = (num, x_1, x_2, \ldots, x_i, \ldots, x_m, MW)$.

3.4. GRU for earthquake prediction

GRU is a simpler variant of LSTM. It uses hyperparameters alongside a sliding window time-step mechanism, akin to other deep networks, to meticulously configure and fine-tune the model [30]. The GRU integrates the input gate, forget gate, and output gate structures of the LSTM into an update gate and a reset gate. Using a single update gate, the GRU can achieve forgetting and selection of memories in the neural network. This significantly reduces the number of parameters, making the internal structure simpler, the computation less intensive, and the training process easier compared to the LSTM [31].

The gating mechanism in the GRU allows it to capture information more effectively across different time steps. The GRU can effectively address the long-term dependency problem through its gating mechanism, which enables it to learn and retain long-term information from historical data. This capability is critical for predicting complex phenomena such as earthquakes. The specific principles of GRU are as follows: the update gate determines how much historical and current information is used to update the current implicit state. The update gate at moment t is described as z_t, which represents the update signal for the gating, and z_t is denoted as Eq. (5),

$$z_t = \delta(W_z \cdot [h_{t-1}, x_t]) \tag{5}$$

where the magnitude of z_t determines to what extent the hidden state of the candidate is remembered. h_{t-1} represents the hidden historical state, and x_t denotes the input data at time t. W_z is the weight matrix and σ is the sigmoid function.

The reset gate determines how much historical information is retained. The reset gate at moment t is described as r_t, which is denoted as Eq. (6),

$$r_t = \delta(W_r \cdot [h_{t-1}, x_t]) \tag{6}$$

where a larger reset signal value indicates that more historical information needs to be remembered. W_r is the weight matrix. Under the influence of the update gate z_t and the reset gate r_t, the hidden state candidate and the hidden output state h_t in the current time step, the update is described as Eq. (7),

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t \tag{7}$$

which candidates for the implicit state are described by Eq. (8).

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \cdot h_{t-1}, x_t]) \tag{8}$$

The candidate hidden state is responsible for integrating the informational features of the input data and historical data. This operation is related to the reset signal r_t obtained from the reset gate. The hyperparameters of GRU include hidden size, num layers, learning rate, batch size, seq length, dropout, and weight decay, which are described as $H = (hSize, nLayers, lRate, bSize, sLength, wDecay)$.

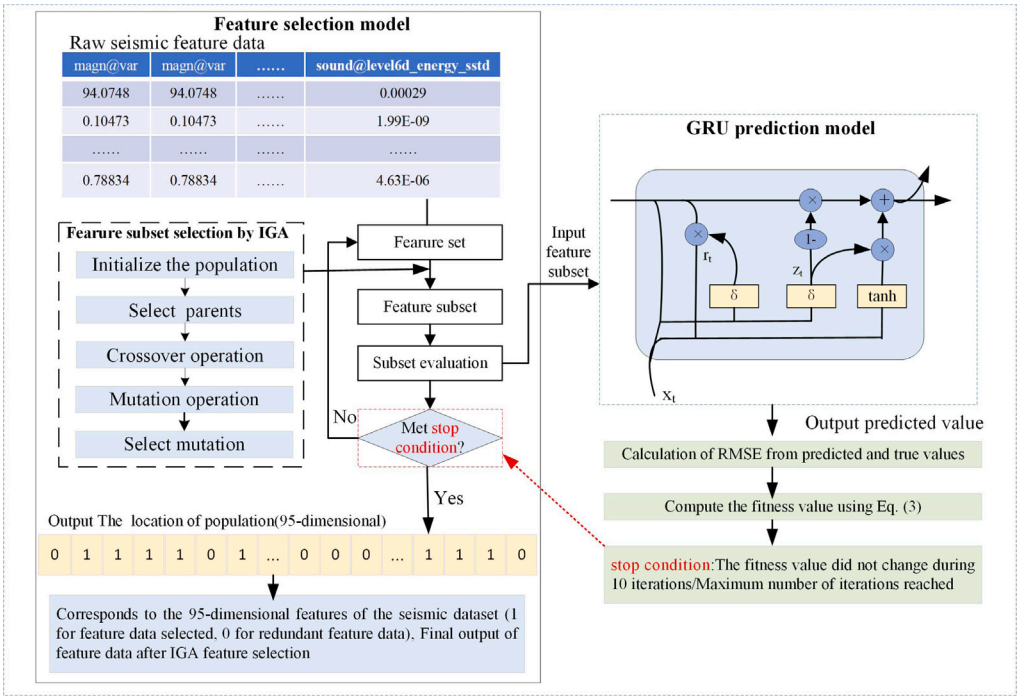


Fig. 4. IGA-based feature selection method for GRU model.

3.5. CSO for hyperparameter optimization

The CSO is one of the bionic intelligent systems inspired by the biological immune system and is a new frontier of research in the field of artificial intelligence. To generate uniformly distributed samples that address the issue of non-uniform sampling and susceptibility to local optima in the original CSO, we introduce LHS for population initialization. This approach can enhance the diversity of the population and enable a more comprehensive exploration of the solution space, thereby reducing the likelihood of becoming trapped in local optima. Additionally, the fixed mutation rate in the original CSO leads to ineffective exploration of the search space in the early stages and lacks fine-grained local search capabilities when approaching the global optimum. To address this issue, we propose an adaptive mutation strategy that adaptively adjusts the mutation rate at different search stages. That is, for poorer solutions, a larger mutation rate is given to increase exploration capabilities, while for better solutions, a smaller mutation rate is applied for fine-tuning. This enhances the algorithm's local search capabilities and improves the accuracy of the solution.

As a popular artificial immune computational model, CSO has been widely applied to many optimization problems [32]. The CSO generates new solutions primarily through the hypermutation operator, which simulates the immune response process. The specific scheme for applying CSO to the GRU hyperparameter tuning is illustrated in Fig. 5. Using CSO to simulate the diversity of the biological immune system helps to explore a wider range of combinations within the hyperparameter search space, leading to the identification of more optimal configurations. Memory cells in CSO can remember historically well-performing parameter combinations. Applying them to the GRU model hyperparameter tuning can expedite the search for optimal hyperparameters, potentially increasing flexibility and efficiency when optimizing the GRU model across different datasets. The general steps of our improved CSO are as follows.

Step 1: Initialization: Randomly initialize a population containing individuals (N), initial population generation using latin hypercube sampling. This step is described as Eq. (9),

$$P_0 = \mathrm{LHS}(\alpha, \beta) \tag{9}$$

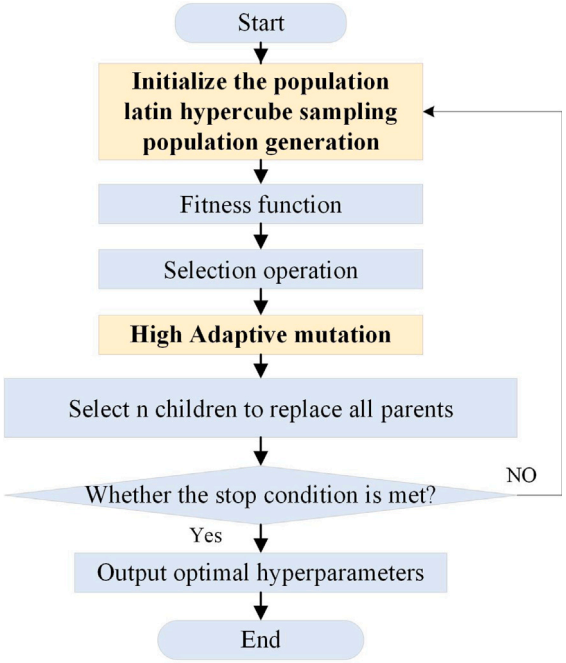


Fig. 5. CSO for hyperparameter optimization.

where α is the number of hyperparameters and β is the size of the population.

Step 2: Assessment: Given a set of patterns to be recognized (P), for each pattern, determine its affinity (degree of match) with each element of the population. For each individual x_i in the population, calculate the fitness value using the fitness function fc. This procedure is described as Eqs. (10) and (11).

$$\mathrm{F} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{10}$$

$$fc_i = \mathrm{F}(x_i) \tag{11}$$

Step 3: Selection and cloning: Select a subset of n individuals with the highest affinity in the population N, and generate the corresponding clones based on their affinity for the antigen. Calculate the adjusted fitness $\tilde{fc}_i$ of each individual based on the maximum fitness value $fc_{\max}$ and the minimum fitness value $fc_{\min}$. This stage is described in Eq. (12).

$$\tilde{fc}_i = \frac{(fc_{\max} - fc_i + e \times 10^{-10})}{(fc_{\max} - fc_{\min} + e \times 10^{-10})} \tag{12}$$

Step 4: High mutation: Calculate the mutation rate P_m of each individual based on its fitness value fc_i, minimum fitness $fc_{\min}$, and maximum fitness $fc_{\max}$. This process is described in Eq. (13),

$$P_m = \mu_{\max} - (\mu_{\max} - \mu_{\min}) \times \tilde{fc}_i \tag{13}$$

where $\mu_{\max}$ is the maximum mutation rate and $\mu_{\min}$ is the minimum mutation rate.

For each cloned individual, generate a mutation mask Mu based on the mutation rate and apply Gaussian noise G. This process is described in Eqs. (14) and (15),

$$Mu = R < P_m \tag{14}$$

$$G = \mathcal{N}(0, P_m) \tag{15}$$

here, R contains random values between 0 and 1, and P_m is the mutation probability. This creates a binary mask determining which elements of an individual will be mutated, where noise G follows a normal distribution with mean 0 and standard deviation P_m, allowing adaptive mutation intensity based on the mutation probability. The mutated individual $\hat{x}$ is given by Eq. (16),

$$\hat{x} = x + Mu \odot G \tag{16}$$

where the mutated individual $\hat{x}$ is created by adding Gaussian noise G to specific positions in the original individual x, as determined by the mask Mu, through element-wise multiplication ($\odot$) achieved.

Step 5: Population update: Select the best individuals from the original population and the mutated cloned individuals to form a new population. Combine all individuals, sort them according to fitness values, and select the top N individuals to ensure that the quality of the population continues to improve. This phase is described as Eq. (17),

$$P_{\text{next}} = \text{Top-K}(P \cup C, \mathrm{N}) \tag{17}$$

where P_{next} represents the next generation population, P is the current population, C is the set of mutated clones, and N is the population size. The Top-K operator selects the N individuals with low fitness values from the unified set of both original and cloned individuals ($P \cup C$). This selection mechanism implements an elitist strategy that preserves the best individuals across generations.

Step 6: Repeat Steps 2–5 until the termination condition is met. Finally, the optimal hyperparameters can be found, which are described as $\overline{H} = (nUnit, lDropout, lRate, nEopch, wDecay)$.

Given that the GRU prediction model involves multiple hyperparameters, manual tuning can be both challenging and time-consuming. Therefore, we implemented the aforementioned CSO hyperparameter optimization method to tune five critical hyperparameters of the GRU model, thereby enhancing its predictive performance. The process begins by partitioning the IGA-selected feature data into a 70% train set and a 30% test set. Subsequently, the five GRU hyperparameters are input into the CSO algorithm for optimization. During the optimization process, there is real-time interaction with the GRU prediction model, where the hyperparameters from each iteration are used to train the GRU model and validate its performance on the validation set to obtain the RMSE. The optimization process terminates and outputs the final optimal GRU hyperparameters for training when either the RMSE remains unchanged for 4 consecutive iterations or the maximum iteration count is reached. The specific process of CSO-based GRU prediction model hyperparameter optimization is illustrated in Fig. 6.

Latin hypercube initialization strategy: To enhance the diversity and uniformity of the initial population, this study incorporates LHS during the population initialization phase, as formally expressed in Eq. (9). The LHS method effectively divides the search space in each dimension into equal intervals, ensuring that each interval in every dimension is sampled exactly once. This approach successfully addresses the limitations of traditional random initialization methods, which often result in individuals clustering in certain regions of the search space, leading to poor coverage and compromised global search capabilities. The LHS initialization strategy ensures a more uniform distribution of the initial population across the entire search space, effectively enhancing the algorithm's global search capability while reducing the risk of premature convergence.

Adaptive mutation strategy: An adaptive mutation strategy has been incorporated into the aforementioned CSO algorithm to further balance global exploration and local exploitation capabilities. Unlike traditional fixed mutation rates, the adaptive mutation rate dynamically adjusts based on the current population's fitness distribution, thereby enhancing search efficiency and solution quality. The relative fitness calculation is expressed in Eq. (12), while the final individual mutation rate is formulated in Eq. (13). As evident from Eqs. (12)–(13), during the early stages of the algorithm, when the population has not undergone sufficient iterative optimization, individuals generally exhibit higher fitness values, resulting in lower relative fitness and consequently higher mutation rates . These elevated mutation rates facilitate broader exploration of the search space by individuals, strengthening the algorithm's global search capability and preventing convergence to local optima. As iterations progress, fitness differentiation among population members increases, with superior individuals gradually achieving higher relative fitness values and correspondingly lower mutation rates, thus enhancing local exploitation capabilities and enabling refined searches around promising solutions. This adaptive mutation strategy effectively addresses the algorithm's search requirements at different stages, improving overall optimization performance.

4. Experiments and analysis

4.1. Data collection

The experimental data of this study are obtained from the AETA platform developed by our team. To make the experimental results more accurate, we select seismic data from six seismic stations in Sichuan Province as our experimental data and perform data fusion and data processing. The six seismic stations in Sichuan Province with frequent earthquakes collected by AETA include Ganzi County Earthquake Prevention and Disaster Reduction Bureau Seismic Station (DS1), Dujiangyan Middle School Seismic Station (DS2), Nanxi Seismic Station in Yulong County, Lijiang (DS3), Qiaojia Seismic Station (DS4), Furong Seismic Station in Pingshan, Yibin (DS5), and Shifang City Earthquake Prevention Seismic Station (DS6). The frequency and magnitude distributions of these six seismic stations are shown in Fig. 7.

4.2. Data fusion

The seismic data collected by AETA with electromagnetic and geoacoustic data is fused to finally obtain a seismic dataset with several important features, where the ratio of the training set to the test set is 7:3. Table 2 shows part of the DS1 data after data fusion and Fig. 8 shows the trend in magnitude levels from 2017 to 2023 for each station after data processing.

The linear regression correlation between the features of our fused dataset and the earthquake magnitude is shown in Fig. 9. The blue dots represent each data point, where the horizontal axis corresponds to the

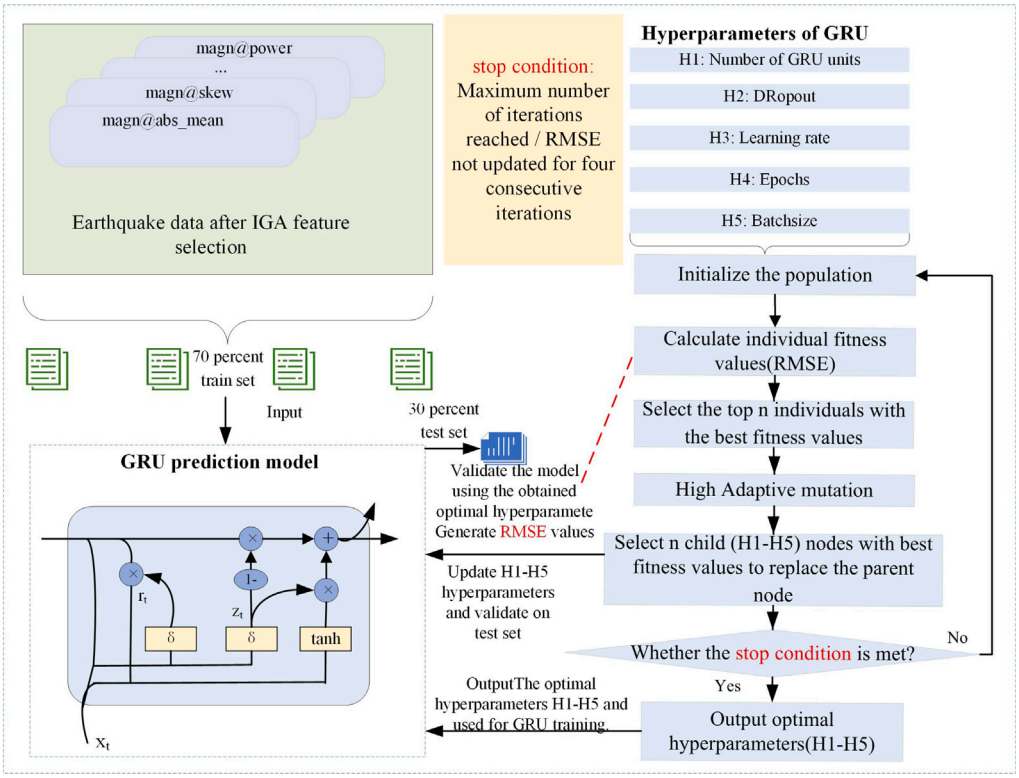


Fig. 6. Hyper-parameters optimization method of GRU model based on CSO.

Table 2
Data characteristics of DS1 after data fusion.

MW	*magn@var*	*magn@skew*	*magn@kurt*	*magn@abs$_{max}$*	...	*sound@var*	*sound@power*	*sound@skew*	*sound@kurt*
1.1	0.6785	0.6785	0.0725	9.9661	...	0.6961	0.0001	3.1188	103.1413
0.9	0.4035	0.4035	0.0863	−0.9274	...	0.0001	0.0001	0.0266	0.0540
2.1	0.4912	0.4912	0.1100	−1.2451	...	0.0001	0.0001	0.0319	0.0518
0.9	0.6444	0.6444	0.1250	−1.1689	...	0.0001	0.0001	0.0210	0.0652
...	...	...	...	...	...	...	...	...	...
1.4	0.6877	0.6877	0.3275	−0.582	...	0.0001	0.0001	0.0189	0.0426
3.3	2.7899	2.7899	0.4124	12.3956	...	0.0002	0.0002	0.0053	−0.0269
0.2	3.1448	3.1448	0.4181	3.4922	...	0.0002	0.0002	0.0018	0.0536
1.5	1.4616	1.4616	0.2271	7.5639	...	0.0002	0.0002	0.0043	−0.0270

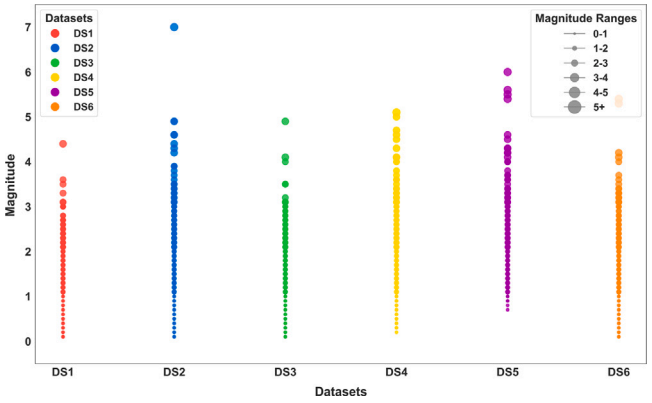


Fig. 7. The frequency and magnitude distributions of these six seismic stations.

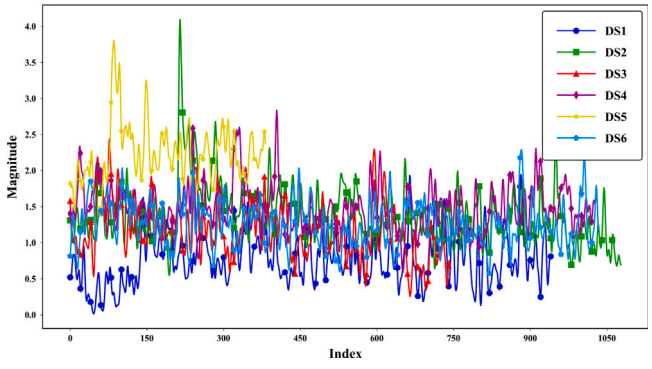


Fig. 8. Trend map of magnitude changes at each station.

value of the feature. When the blue dots are concentrated in a specific direction, it indicates a stronger association between the feature and the magnitude. The red line illustrates the linear relationship between the feature and the magnitude. If the regression line slopes upward, it indicates a positive correlation between the feature and the magnitude; if the line slopes downward, it indicates a negative correlation; and if the regression line is horizontal, it suggests no correlation. The red shaded surface in the figure represents the confidence interval of the linear regression fitting curve, which indicates the uncertainty of the estimated relationship between each seismic feature and the magnitude; the narrower the shaded area, the more accurate the estimation of

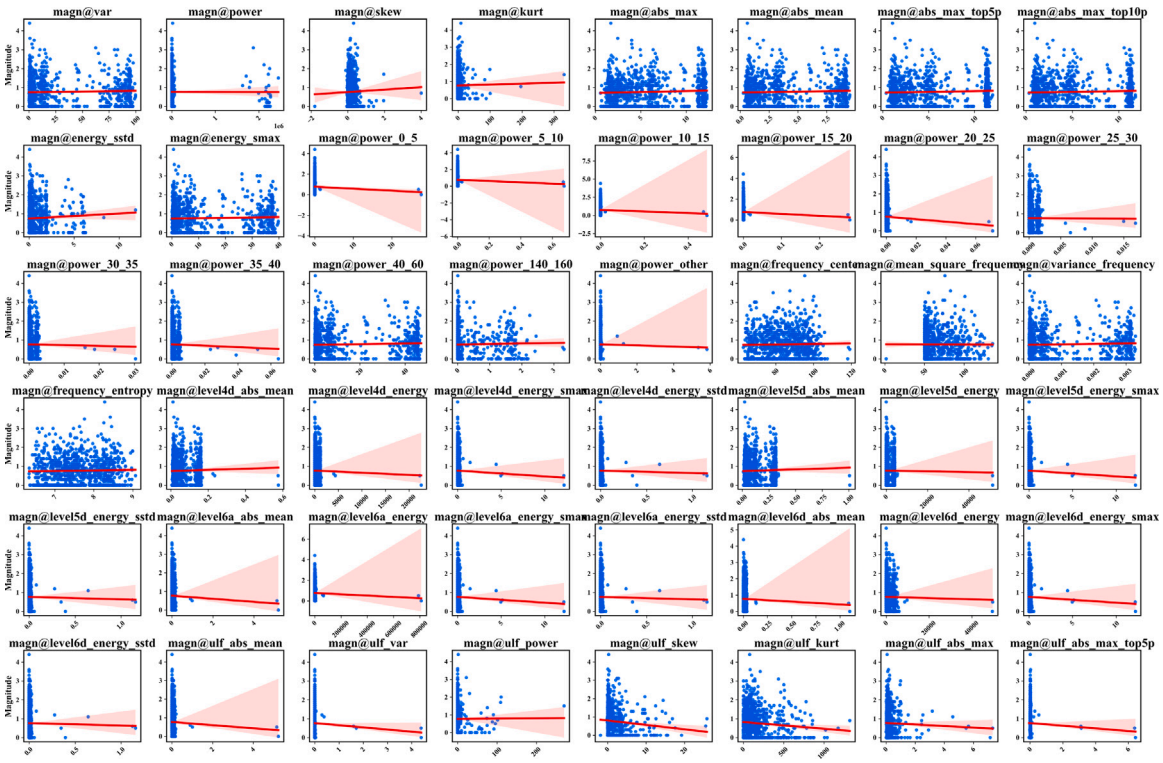


Fig. 9. Linear regression correlation plot of features and magnitude.

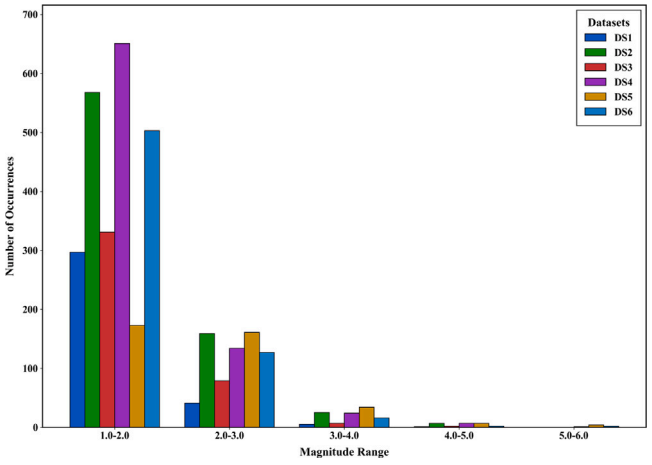


Fig. 10. Distribution of earthquake magnitudes across six datasets.

the relationship between the feature and the magnitude; and the wider the shaded area, the greater the uncertainty of the prediction. From the figure, we can see that most features of the fused dataset exhibit a strong correlation with magnitude, which provides an important basis for us to build earthquake magnitude prediction models in the subsequent stages.

Fig. 10 shows the distribution of earthquake occurrences in different magnitude ranges in the data sets corresponding to six regions. It can be seen from the figure that most earthquake events in all datasets are concentrated in the magnitude range of 1.0 to 2.0, and the number of events decreases significantly as the magnitude increases.

4.3. Assessment indicators

Several evaluation metrics are implemented to evaluate the performance of the proposed method, including the root mean square error (RMSE), the mean absolute error (MAE), the mean squared error (MSE) and the coefficient of determination (R^2) [33]. These metrics are calculated according to Eqs. (18)–(21) and are used to compare the proposed method with other existing methods.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{18}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{19}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{20}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{21}$$

4.4. Feature selection performance analysis

Selecting appropriate features plays a crucial role in the predictive performance of the model. The IGA is introduced for feature selection to improve the accuracy of our earthquake magnitude prediction model. To evaluate the performance of feature selection of this method, we compared it with the Steady-State Genetic Algorithm (Steady-GA), the Stud Genetic Algorithm (Stud-GA), EGA, and the Genetic Algorithm (GA) across six datasets. Performance was assessed using the maximum fitness value (f_{max}), minimum fitness value (f_{min}), average fitness value (f_{avg}), and the standard deviation of the fitness value (f_{std}), all of which were computed after feature selection. The formula for calculating the fitness value is shown in Eq. (3). The number of features selected for each algorithm on DS1 to DS6 is illustrated in Fig. 11. Fig. 12 presents the comparison of f_{max}, f_{min}, f_{avg}, and f_{std} across all datasets. The results indicate that IGA achieved superior feature selection performance in most datasets, striking a balance between model prediction accuracy and feature selection complexity.

Furthermore, to validate the robustness and stability of the IGA algorithm compared to other algorithms, we conducted comprehensive

Table 3
Test functions.

Function name	Formula	Range	Category	f_{min}
F1	$f_1(x) = \sum_{i=1}^{dim} x_i^2$	$[-100, 100]$	Unimodal	0
F2	$f_2(x) = \sum_{i=1}^{dim} \lvert x_i \rvert + \prod_{i=1}^{dim} \lvert x_i \rvert$	$[-10, 10]$	Unimodal	0
F3	$f_3(x) = \sum_{i=1}^{dim} (\sum_{j=1}^{i} x_j)^2$	$[-100, 100]$	Unimodal	0
F4	$f_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le dim\}$	$[-100, 100]$	Unimodal	0
F5	$f_5(x) = \sum_{i=1}^{dim-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]$	Unimodal	0
F6	$f_6(x) = \sum_{i=1}^{dim} (x_i + 0.5)^2$	$[-100, 100]$	Unimodal	0
F7	$f_7(x) = \sum_{i=1}^{dim} i x_i^4 + rand$	$[-1.28, 1.28]$	Unimodal	0
F8	$f_8(x) = \sum_{i=1}^{dim} -x_i \sin(\sqrt{\lvert x_i \rvert})$	$[-500, 500]$	Multimodal	0
F9	$f_9(x) = \sum_{i=1}^{dim} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	Multimodal	0
F10	$f_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{dim}\sum_{i=1}^{dim} x_i^2}) - \exp(\frac{1}{dim}\sum_{i=1}^{dim}\cos(2\pi x_i)) + 20 + e$	$[-32, 32]$	Multimodal	0
F11	$f_{11}(x) = \sum_{i=1}^{dim} \frac{x_i^2}{4000} - \prod_{i=1}^{dim} \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]$	Multimodal	0
F12	$f_{12}(x) = \frac{\pi}{dim}[10\sin^2(\pi y_1) + \sum_{i=1}^{dim-1}(y_i - 1)^2(1 + 10\sin^2(\pi y_{i+1})) + y_n^2] + \sum_{i=1}^{dim} u(x_i, 10, 100, 4)$	$[-50, 50]$	Multimodal	0
F13	$f_{13}(x) = 0.1[\sin^2(3\pi x_1) + \sum_{i=1}^{dim-1}(x_i - 1)^2(1 + \sin^2(3\pi x_{i+1})) + (x_n - 1)^2(1 + \sin^2(2\pi x_n))]$	$[-50, 50]$	Multimodal	0
F14	$f_{14}(x) = \sum_{i=1}^{dim} x_i^2 + \sum_{i=1}^{dim} \sin^2(x_i)$	$[-65.536, 65.536]$	Multimodal	0
F15	$f_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	$[-5, 5]$	Multimodal	0
F16	$f_{16}(x) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$	$[-5, 5]$	Multimodal	0
F17	$f_{17}(x) = \sum_{i=1}^{dim}(x_i^2 - 10\cos(3\pi x_i) + 10) + \sum_{i=1}^{dim}\sin^2(x_i)$	$[-1.28, 1.28]$	Multimodal	0
F18	$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$	$[-2, 2]$	Multimodal	0
F19	$f_{19}(x) = \sum_{i=1}^{dim}(x_i - 1)^2 + \sum_{i=1}^{dim}(x_i + 2)^2$	$[0, 1]$	Multimodal	0
F20	$f_{20}(x) = \sum_{i=1}^{dim} \lvert x_i \rvert^3 + \sum_{i=1}^{dim}\cos^3(x_i)$	$[0, 1]$	Hybrid	0
F21	$f_{21}(x) = \sum_{i=1}^{dim} x_i^2 + \sum_{i=1}^{dim}\cos^2(x_i)$	$[0, 10]$	Hybrid	0
F22	$f_{22}(x) = \sum_{i=1}^{dim} x_i^4 + \sum_{i=1}^{dim}\sin^2(x_i) + \sum_{i=1}^{dim}\cos^2(x_i)$	$[0, 10]$	Hybrid	0
F23	$f_{23}(x) = (\sum_{i=1}^{dim} x_i^2)(\sum_{i=1}^{dim}\sin^2(x_i)) + e^{\frac{1}{dim}\sum_{i=1}^{dim}\lvert x_i \rvert}$	$[0, 10]$	Hybrid	0

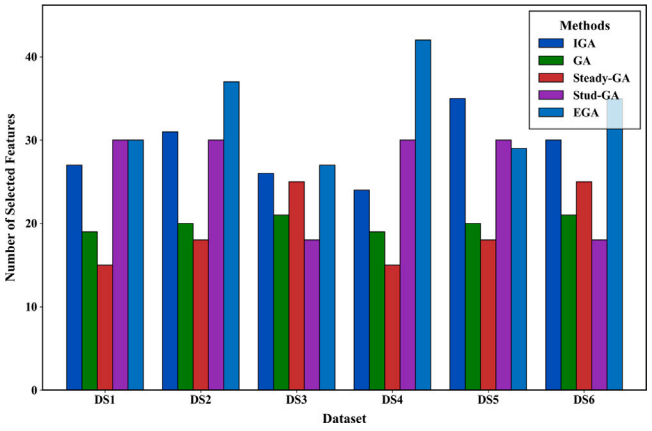


Fig. 11. The number of features across six datasets.

evaluations using 23 benchmark test functions, as detailed in Table 3. The table presents the mathematical expressions, variable ranges, function types, and optimal values for all 23 benchmark functions. Among these, functions F1–F7 are unimodal functions, primarily used to examine the algorithm's global optimization capability; functions F8–F19 are multimodal functions, designed to assess the algorithm's ability to escape local optima; and functions F20–F23 are composite functions, which further evaluate the algorithm's performance in complex search spaces.

The experimental parameters were set with a maximum iteration count of 500 and a population size of 30, which refers to Ref. [34]. The Average Fitness was employed as the evaluation metric. The experimental results and convergence curves are presented in Table 4 and Fig. 13, respectively. As evidenced by Table 4, the IGA algorithm achieved the lowest average fitness values across the majority of test functions, demonstrating its superior capability to converge rapidly toward optimal solutions and indicating robust global search capabilities and exceptional optimization performance. Fig. 13 further illustrates that IGA exhibits faster convergence rates across most test functions, with its best fitness values significantly outperforming those of comparative algorithms. These results comprehensively demonstrate the IGA algorithm's enhanced convergence efficiency and superior optimization capabilities in complex optimization problems.

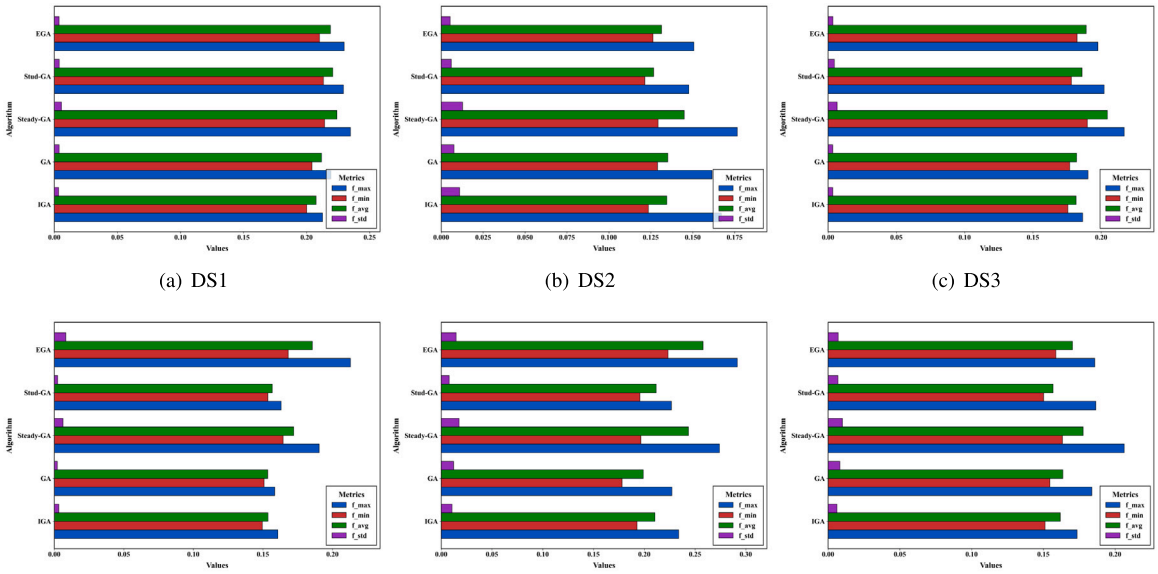
4.5. Hyper-parameters optimizing performance analysis

To validate the effectiveness of CSO in hyperparameter tuning of GRU, we compared its performance with classical hyperparameter tuning methods, including Original CSO(OCSO), Random Search (RS), Grid Search (GS), and Differential Evolution (DE) across six datasets. In the experiments, the experimental results are depicted in Fig. 14. For the metrics RMSE, MSE and MAE, superior performance is indicated by proximity to the origin (center of the radar chart). Conversely, for R^2, a greater distance from the origin signifies better performance. As shown in the figure, the introduction of CSO for hyperparameter tuning in optimizing the GRU prediction model significantly outperforms the other methods on most of the datasets and exhibits a more balanced performance on all evaluation metrics.

4.6. Forecasting performance analysis

4.6.1. Comparison with baseline models and state-of-the-art models

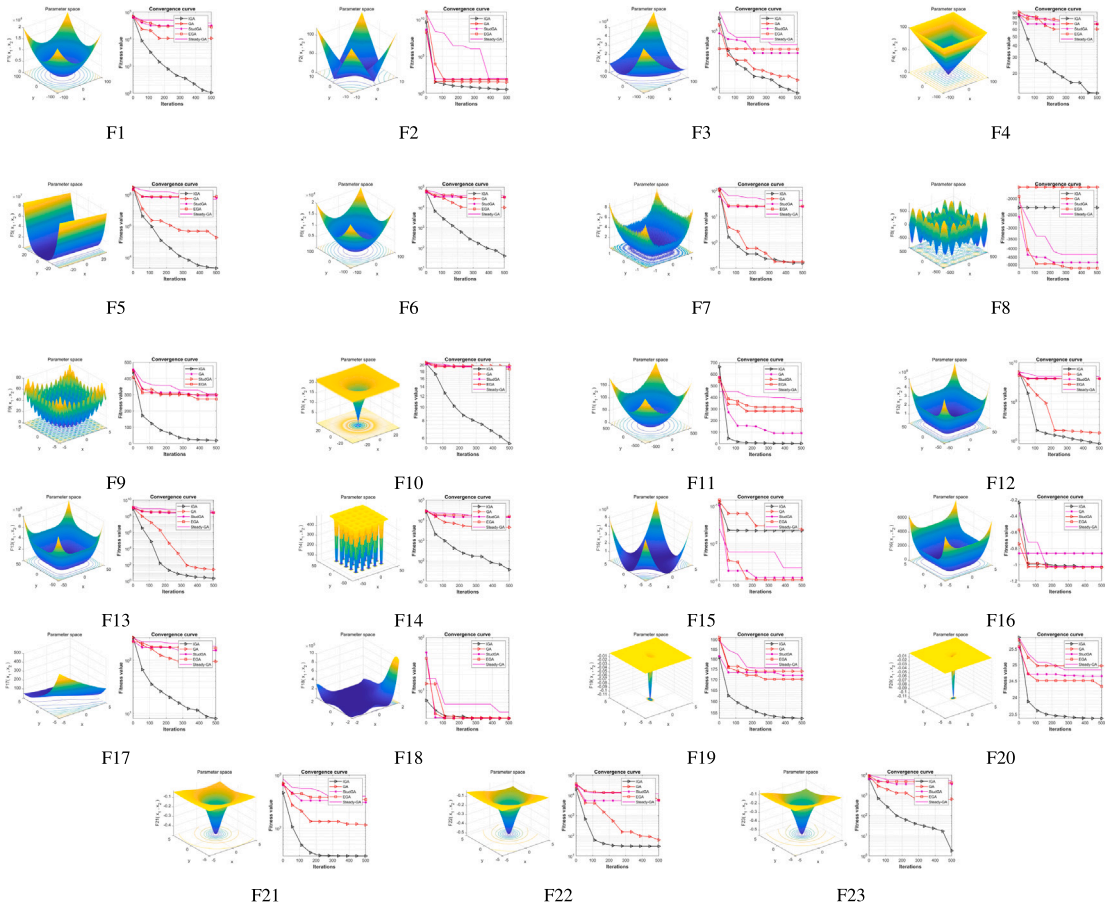
To rigorously evaluate the performance of the proposed IGA-CSOGRU, a comparative analysis against several commonly used machine learning techniques is performed, including Random Forest (RF) [35], Adaboost [36], Linear Regression (LR) and Support Vector Regression (SVR) [37]. To further validate the seismic prediction performance of our proposed method, we chose to use the current state-of-the-art methods LSTM, Bidirectional LSTM (BiLSTM), Bidirectional GRU (BiGRU), EGA-LSTM, Attention-GRU and Attention-LSTM with our



(a) DS1 (b) DS2 (c) DS3

(d) DS4 (e) DS5 (f) DS6

Fig. 12. Comparison of f_{max}, f_{min}, f_{avg}, f_{std} for different algorithms on each dataset.



F1 F2 F3 F4

F5 F6 F7 F8

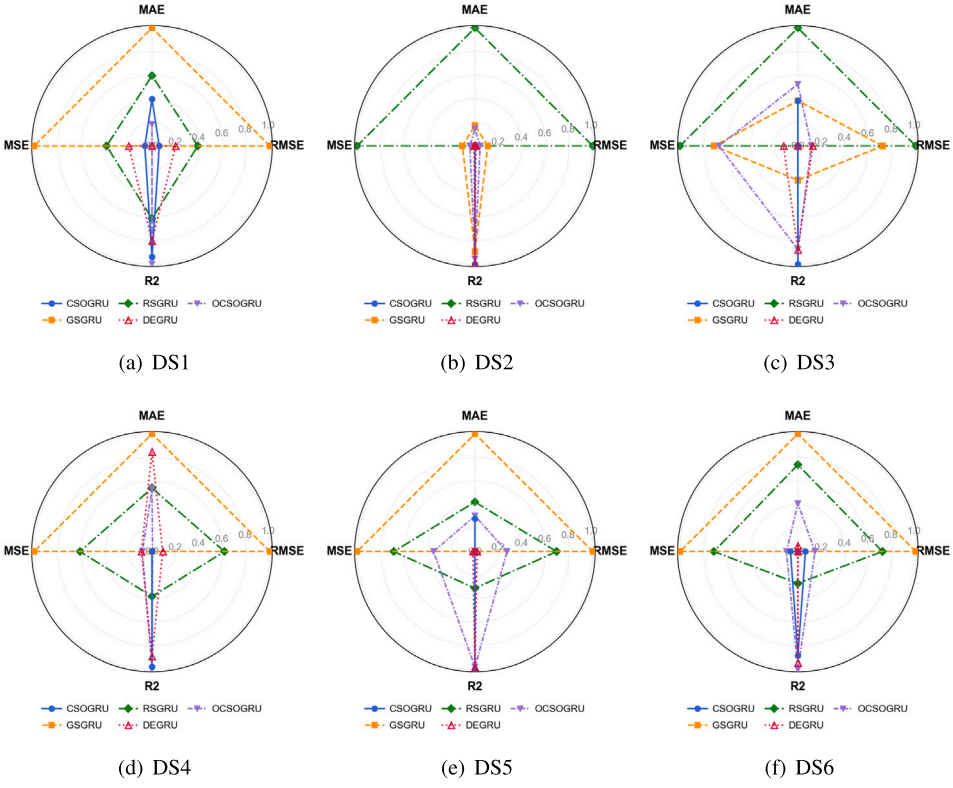
F9 F10 F11 F12

F13 F14 F15 F16

F17 F18 F19 F20

F21 F22 F23

Fig. 13. Convergence curves on each benchmark function.



(a) DS1 (b) DS2 (c) DS3

(d) DS4 (e) DS5 (f) DS6

Fig. 14. Comparison of hyperparameter tuning methods across six datasets.

Table 4
Average fitness of each algorithm under the test function.

Func.	IGA	GA	Stud-GA	EGA	Steady-GA
F1	**1.04473E+02**	1.08010E+04	3.20142E+04	2.78740E+04	3.89448E+04
F2	**3.22656E+00**	3.15612E+01	8.48154E+01	6.89854E+01	1.03571E+02
F3	**8.26873E+03**	1.39152E+04	4.10063E+04	4.79133E+04	5.76149E+04
F4	**1.21738E+01**	6.03641E+01	6.80070E+01	6.85560E+01	7.28157E+01
F5	**2.02174E+03**	1.84168E+05	6.12752E+07	7.65442E+07	4.86564E+07
F6	**4.01628E+01**	9.90361E+03	3.10463E+04	3.18867E+04	3.90789E+04
F7	**1.59007E−01**	1.77699E−01	2.36464E+01	2.51269E+01	4.25520E+01
F8	−2.26633E+03	−1.70623E+03	**−4.85079E+03**	−5.25285E+03	−4.34999E+03
F9	**2.66088E+01**	2.87354E+02	2.87468E+02	2.55419E+02	3.30356E+02
F10	**5.47807E+00**	1.97295E+01	1.92273E+01	1.86650E+01	1.94146E+01
F11	**1.78366E+00**	9.03053E+01	2.81092E+02	2.98528E+02	3.82317E+02
F12	**3.97648E−01**	9.55675E+00	8.26904E+07	8.39407E+07	1.69361E+08
F13	**2.06671E+00**	2.58587E+01	2.45078E+08	2.81499E+08	3.84058E+08
F14	**3.68905E+01**	4.54295E+03	1.44754E+04	1.57968E+04	1.36348E+04
F15	2.19771E−02	2.35722E−02	1.23044E−03	**1.07741E−03**	2.22940E−03
F16	−1.03155E+00	**−8.60051E−01**	−1.03158E+00	−1.03063E+00	−1.02585E+00
F17	**8.00950E+00**	9.45126E+01	1.52680E+02	1.68968E+02	1.83491E+02
F18	**3.00090E+00**	3.05471E+00	3.00187E+00	3.00542E+00	3.93896E+00
F19	**1.52583E+02**	1.74019E+02	1.71979E+02	1.70185E+02	1.75528E+02
F20	**2.33796E+01**	2.49678E+01	2.46489E+01	2.43344E+01	2.48407E+01
F21	**3.00744E+01**	1.15300E+02	3.05718E+02	3.46028E+02	4.00463E+02
F22	**3.00529E+01**	6.20283E+01	5.37228E+03	5.76489E+03	1.17553E+04
F23	**1.81206E+00**	6.35028E+02	3.51485E+03	4.06070E+03	4.95652E+03

proposed IGA-CSOGRU on the data-fused AETA six datasets. The experiments evaluate and analyze the results using RMSE, MAE, MSE and R^2 respectively. In the experiments, the maximum number of iterations of CSO is set to 20 because in the specific experiment we set an early stopping mechanism, that is, the RMSE value does not change in 4 consecutive iterations. Through multiple experiments, we found that the experiment will stop iterating at about 15 iterations. In order to avoid unnecessary computational overhead, we set the maximum number of iterations of CSO to 20. The number of GRU units was varied between 50 and 200, the dropout rate was varied

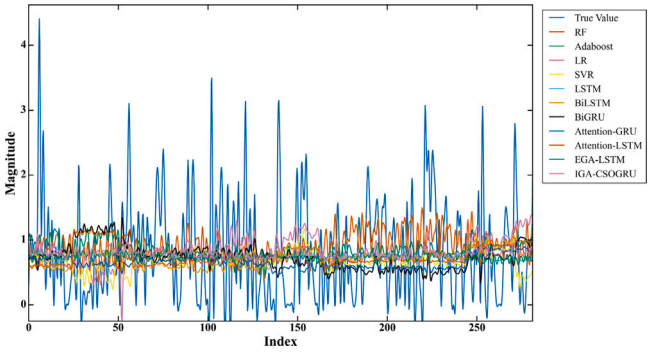


Fig. 15. Predicted versus true value with 10 algorithms curves on DS1.

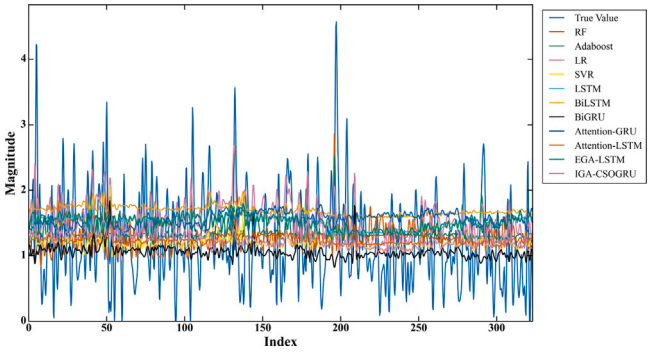


Fig. 16. Predicted versus true value with 10 algorithms curves on DS2.

between 0 and 0.5, the learning rate was varied between 0.0001 and 0.01, and the number of training calendar elements was varied between 30 and 130, the maximum number of iterations of IGA is set to 50, these parameters were set as described in Ref. [6]. The performance of IGA-CSOGRU and other methods in DS1 through DS6 was compared using the metrics RMSE, MAE, MSE, and R^2. The results are shown in Table 5. Smaller RMSE and MSE values indicate better predictive performance, while a R^2 value closer to 1 indicates better model fitness. The results in bold in the table represent the best performance. From the experimental results, it is evident that our proposed model significantly outperforms baseline models and state-of-the-art models in various evaluation metrics. This improvement is attributed to the incorporation of feature selection and parameter optimization within our IGA-CSOGRU predictive model, resulting in a substantial enhancement of the model's predictive capabilities.

The predicted values of the proposed model, baseline models, and the state-of-the-art models are compared with the true values to further validate the predictive performance of the proposed model. The results are shown in Figs. 15 and 16, and the ordinate denotes the predicted earthquake magnitudes. The visual analysis of the figures clearly indicates that the IGA-CSOGRU predictive model exhibits superior accuracy and maintains a high degree of stability.

4.6.2. Model performance analysis

4.6.3. Ablation experiments

In the prediction process of the magnitude of the earthquake, IGA is used to select the characteristics of the seismic data, and CSO is used to regulate the hyperparameters of the GRU prediction model. To verify the role of IGA and CSO in predicting earthquake magnitude, we compare and analyze GRU, IGA-GRU, and IGA-CSOGRU in six datasets, and the experimental results are shown in Table 6. Smaller RMSE and MSE values indicate better predictive performance, while a R^2 value closer to 1 indicates a better model fit. The results in bold in the table represent the best performance. The results of the ablation experiments reveal that the IGA-CSOGRU model achieves optimal results, underscoring the substantial contribution of the IGA for feature selection and the CSO for parameter adjustment to the model's predictive efficacy.

The fitting curves illustrate the performance of the model in the training data. By analyzing the convergence and stability of the curves, we can determine whether the model has effectively learned the features of the training data. Adaptation curves, on the other hand, show changes in model fitness during the optimization process. The optimization efficiency and global search capability of the model can be assessed to examine the trend of these curves, further validating the predictive performance and stability of the model. To validate the predictive performance and stability of the IGA-CSOGRU model, we further examined the fitted curves and the adaptation curves observed during the training process. A high degree of curve fitting reflects the model's ability to precisely capture the intrinsic relationships among data points, resulting in minimal variance between the predicted and actual values. The sustained descent and eventual stabilization of the fitness curve imply that the proposed model has approached or achieved convergence, indicative of robust performance, and also suggest that the hyperparameter tuning of the model has been appropriately conducted. The fitting curves are shown in Fig. 17, and the adaptation curves are shown in Fig. 18.

From the experimental results, it can be observed that the seismic prediction model used for most datasets demonstrates high accuracy and reliability. However, the goodness-of-fit is poor for DS1. This is due to the sparse data points in certain regions of DS1, which prevents the fitted curve from accurately capturing the trend changes in these areas. Observing the fitness value curves on different datasets, it can be seen that the proposed model has a small error between the predicted values and the actual observations. The model accurately reflects the actual conditions of the seismic events. In addition, the smooth and well-performing fitness curves indicate that our proposed model maintains stable predictive performance across different datasets, demonstrating good robustness.

5. Conclusions and future work

In this study, we introduce a novel earthquake prediction method based on a gated recurrent unit with clonal selection and an improved genetic algorithm with generation gap, named IGA-CSOGRU. First, we utilize the genetic algorithm with generation gap for feature selection, identifying features that are highly correlated with earthquake magnitude prediction. Subsequently, to enhance prediction accuracy, we employ a gated recurrent unit model to improve the selection ability of temporal features and incorporate CSO for hyperparameter tuning. To validate the predictive performance and generalizability of the proposed model, we made predictions across different regions and datasets. Experimental results demonstrate that our model significantly outperforms both traditional and advanced prediction models, exhibiting excellent reliability and robustness. The RMSE was reduced by an average of 5%–7% compared to all baseline methods. Specifically, the model improved by 4%–5% over state-of-the-art deep learning methods and by a significant 8%–11% over traditional machine learning methods.

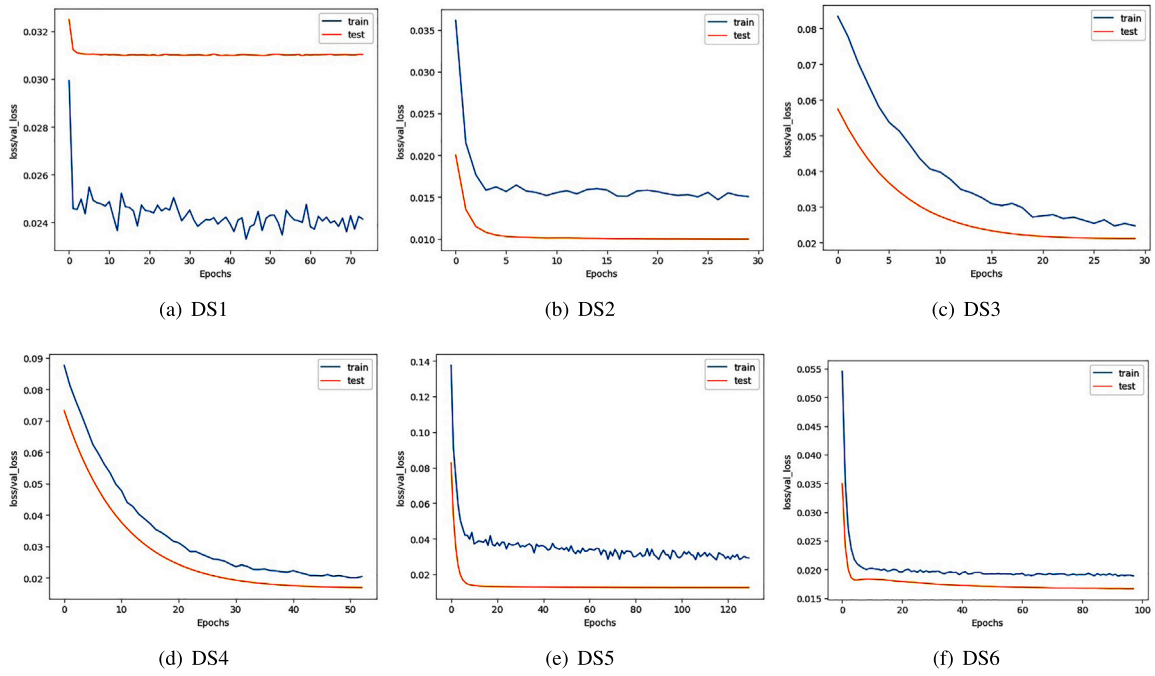
However, our proposed model still exhibits some discrepancies between predicted results and actual values, and due to the scarcity of large earthquake samples, its predictive performance may be affected. In future work, we will focus on improving the model's predictive capabilities and accuracy in large earthquake prediction through a series of explorations.

Table 5
Comparison of the validity of IGA-CSOGRU and ML at each dataset.

Method	Indicators	DS1	DS2	DS3	DS4	DS5	DS6
RF	RMSE	0.182199	0.105068	0.166917	0.135932	0.134371	0.135220
	MAE	0.146128	0.080384	0.140810	0.105125	0.107031	0.107355
	MSE	0.033196	0.011039	0.027861	0.018477	0.018055	0.018284
	R^2	−0.048194	−0.071054	−0.249819	−0.071096	−0.425018	−0.097473
Adaboost	RMSE	0.178550	0.103299	0.164547	0.132970	0.125685	0.137478
	MAE	0.143240	0.081634	0.139573	0.103164	0.101017	0.112092
	MSE	0.031880	0.010670	0.027075	0.017681	0.015796	0.018900
	R^2	−0.006627	−0.035303	−0.214577	−0.024916	−0.246750	−0.134420
LR	RMSE	0.180453	0.111920	0.156888	0.152328	0.151258	0.159278
	MAE	0.143578	0.090630	0.126156	0.115536	0.120531	0.124485
	MSE	0.032563	0.012526	0.024613	0.023203	0.022878	0.025369
	R^2	−0.028199	−0.215325	−0.104139	−0.345051	−0.805700	−0.522725
SVR	RMSE	0.182747	0.101664	0.154402	0.158791	0.126505	0.364691
	MAE	0.142443	0.077536	0.126635	0.109558	0.094272	0.191051
	MSE	0.033396	0.010335	0.023840	0.025214	0.016003	0.132999
	R^2	−0.054508	−0.002785	−0.069430	−0.461618	−0.263074	−6.982825
LSTM	RMSE	0.180263	0.101339	0.153194	0.132099	0.124410	0.132258
	MAE	0.142765	0.078806	0.128053	0.100780	0.088787	0.105028
	MSE	0.032494	0.010269	0.023468	0.017450	0.015478	0.017492
	R^2	−0.026031	0.003614	−0.044476	−0.011541	−0.221585	−0.049914
BiLSTM	RMSE	0.179338	0.101023	0.152339	0.132660	0.116621	0.135922
	MAE	0.140555	0.076651	0.126316	0.100033	0.086342	0.109853
	MSE	0.032162	0.010205	0.023207	0.017598	0.013600	0.018474
	R^2	−0.015524	0.009824	−0.032843	−0.020145	−0.073417	−0.108886
BiGRU	RMSE	0.179463	0.103698	0.153806	0.136514	0.127287	0.135520
	MAE	0.138861	0.076887	0.127350	0.102561	0.099715	0.109200
	MSE	0.032206	0.010753	0.023656	0.018636	0.016202	0.018365
	R^2	−0.016941	−0.043310	−0.052833	−0.080283	−0.278737	−0.102340
Attention-LSTM	RMSE	0.178948	0.107816	0.154911	0.132482	0.136060	0.129741
	MAE	0.141440	0.086974	0.128977	0.100443	0.096774	0.104013
	MSE	0.032022	0.011624	0.023997	0.017551	0.018512	0.016832
	R^2	−0.011116	−0.127830	−0.068018	−0.017410	−0.461071	−0.010338
Attention-GRU	RMSE	0.183642	0.103453	0.166269	0.134429	0.125481	0.137323
	MAE	0.146953	0.079689	0.139371	0.102355	0.098020	0.108912
	MSE	0.033724	0.010702	0.027645	0.018071	0.015745	0.018857
	R^2	−0.064858	−0.038384	−0.230380	−0.047539	−0.242712	0.131862
EGA-LSTM	RMSE	0.178418	0.101125	0.150472	0.131267	0.113444	0.131780
	MAE	0.140621	0.077376	0.121392	0.099404	0.085121	0.106409
	MSE	0.031833	0.010226	0.022642	0.017231	0.012869	0.017366
	R^2	−0.002788	0.007829	−0.007679	−0.001748	−0.010248	−0.045839
IGA-CSOGRU	RMSE	**0.176173**	**0.099904**	**0.145607**	**0.129986**	**0.112129**	**0.129094**
	MAE	**0.140063**	**0.076815**	**0.119320**	**0.098615**	**0.086680**	**0.102648**
	MSE	**0.031037**	**0.009980**	**0.021201**	**0.016896**	**0.012573**	**0.016665**
	R^2	**0.019995**	**0.031634**	**0.048933**	**0.020561**	**0.007682**	**−0.000274**

Table 6
Results of ablation experiment.

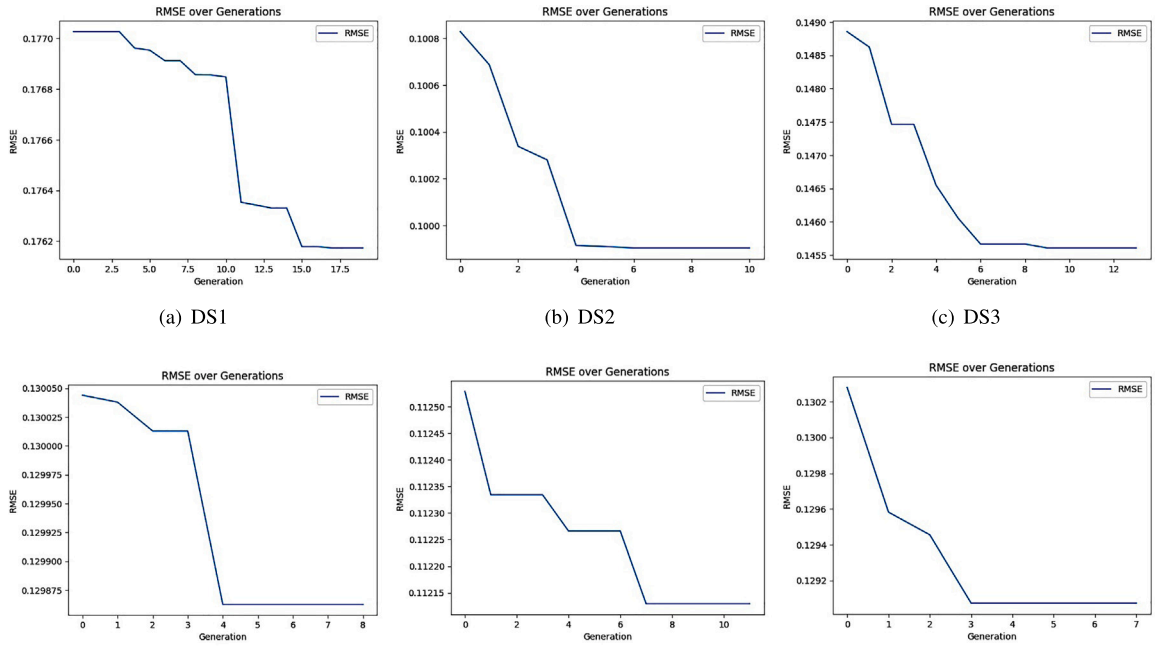
Method	Indicators	DS1	DS2	DS3	DS4	DS5	DS6
GRU	RMSE	0.182223	0.103778	0.157141	0.134875	0.130966	0.136756
	MAE	0.142529	0.082931	0.129117	0.103803	0.092692	0.110298
	MSE	0.033205	0.010770	0.024693	0.018191	0.017152	0.018702
	R^2	−0.048466	−0.044929	−0.107703	−0.054488	−0.353719	−0.122536
IGA-GRU	RMSE	0.179193	0.102011	0.150107	0.131818	0.115741	0.135119
	MAE	0.141902	0.079539	0.124615	0.102578	0.088522	0.108924
	MSE	0.032110	0.010406	0.022532	0.017376	0.013396	0.018257
	R^2	−0.013893	−0.00964	−0.010757	−0.007240	−0.057267	−0.095836
IGA-CSOGRU	RMSE	**0.176173**	**0.099904**	**0.145607**	**0.129986**	**0.112129**	**0.129094**
	MAE	**0.140063**	**0.076815**	**0.119320**	**0.098615**	**0.086680**	**0.102648**
	MSE	**0.031037**	**0.009980**	**0.021201**	**0.016896**	**0.012573**	**0.016665**
	R^2	**0.019995**	**0.031634**	**0.048933**	**0.020561**	**0.007682**	**−0.00027**



(a) DS1 (b) DS2 (c) DS3

(d) DS4 (e) DS5 (f) DS6

Fig. 17. Fitting curves on each dataset.



(a) DS1 (b) DS2 (c) DS3

(d) DS4 (e) DS5 (f) DS6

Fig. 18. Adaptation curves about RMSE on each dataset.

CRediT authorship contribution statement

Wen Zhou: Methodology, Writing – review & editing. **Xinchun Yi:** Writing – review & editing, Writing – original draft, Project administration, Investigation, Methodology, Visualization. **Changyi Li:** Methodology, Investigation. **Zhiwei Ye:** Investigation, Methodology, Writing – review & editing. **Qiyi He:** Data curation, Investigation. **Xiuwen Gong:** Investigation, Methodology. **Qiao Lin:** Methodology, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by NSFC, China http://www.nsfc.gov.cn/ under Grant 62202147, Grant 62376089, and Grant 42201464;

in part by the Hubei University of Technology Green Industrial Technology Leading Program, China under Grant XJ2024000601; and in part by the Hubei Provincial Department of Education Scientific Research Plan Key Project, China under Grant D20231404.

Data availability

The datasets are available at https://github.com/123fggv/experimental-data. For further needs, contact the corresponding author.

References

[1] M.S. Abdalzaher, M.S. Soliman, S.M. El-Hady, A. Benslimane, M. Elwekeil, A deep learning model for earthquake parameters observation in IoT system-based earthquake early warning, IEEE Internet Things J. 9 (11) (2021) 8412–8424, http://dx.doi.org/10.1109/JIOT.2021.3114420.

[2] J. León, M. Ordaz, E. Haddad, I. Araújo, Risk caused by the propagation of earthquake losses through the economy, Nat. Commun. 13 (1) (2022) 2908, http://dx.doi.org/10.1038/s41467-022-30504-3.

[3] A. Panakkat, H. Adeli, Recent efforts in earthquake prediction (1990–2007), Nat. Hazards Rev. 9 (2) (2008) 70–80, http://dx.doi.org/10.1061/(ASCE)1527-6988(2008)9:2(70).

[4] A. Berhich, F.-Z. Belouadha, M.I. Kabbaj, A location-dependent earthquake prediction using recurrent neural network algorithms, Soil Dyn. Earthq. Eng. 161 (2022) 107389, http://dx.doi.org/10.1016/j.soildyn.2022.107389.

[5] A. Berhich, F.-Z. Belouadha, M.I. Kabbaj, An attention-based LSTM network for large earthquake prediction, Soil Dyn. Earthq. Eng. 165 (2023) 107663, http://dx.doi.org/10.1016/j.soildyn.2022.107663.

[6] Z. Ye, W. Lan, W. Zhou, Q. He, L. Hong, X. Yu, Y. Gao, Elite GA-based feature selection of LSTM for earthquake prediction, J. Supercomput. (2024) 1–26, http://dx.doi.org/10.1007/s11227-024-06218-2.

[7] B. Li, Z. Li, J. Chen, Y. Yan, Y. Lv, W. Du, MAST-GNN: A multimodal adaptive spatio-temporal graph neural network for airspace complexity prediction, Transp. Res. Part C: Emerg. Technol. 160 (2024) 104521, http://dx.doi.org/10.1016/j.trc.2024.104521.

[8] W. Luo, X. Lin, T. Zhu, P. Xu, A clonal selection algorithm for dynamic multimodal function optimization, Swarm Evol. Comput. 50 (7) (2019) 100459, http://dx.doi.org/10.1016/j.swevo.2018.10.010.

[9] M.H. Al Banna, K.A. Taher, M.S. Kaiser, M. Mahmud, M.S. Rahman, A.S. Hosen, G.H. Cho, Application of artificial intelligence in predicting earthquakes: state-of-the-art and future challenges, IEEE Access 8 (2020) 192880–192923, http://dx.doi.org/10.1109/ACCESS.2020.3029859.

[10] An automated earthquake classification model based on a new butterfly pattern using seismic signals, Expert Syst. Appl. 238 (2024) 122079, http://dx.doi.org/10.1016/j.eswa.2023.122079.

[11] P. Kavianpour, M. Kavianpour, E. Jahani, A. Ramezani, A CNN-BiLSTM model with attention mechanism for earthquake prediction, J. Supercomput. 79 (17) (2023) 19194–19226, http://dx.doi.org/10.1007/s11227-023-05369-y.

[12] G.C. Beroza, M. Segou, S. Mostafa Mousavi, Machine learning and earthquake forecasting—next steps, Nat. Commun. 12 (1) (2021) 4761, http://dx.doi.org/10.1038/s41467-021-24952-6.

[13] B. Rouet-Leduc, C. Hulbert, N. Lubbers, K. Barros, C.J. Humphreys, P.A. Johnson, Machine learning predicts laboratory earthquakes, Geophys. Res. Lett. 44 (18) (2017) 9276–9282, http://dx.doi.org/10.1002/2017GL074677.

[14] A. Ikram, U. Qamar, Developing an expert system based on association rules and predicate logic for earthquake prediction, Knowl.-Based Syst. 75 (2015) 87–103, http://dx.doi.org/10.1016/j.knosys.2014.11.024.

[15] Artificial neural networks for earthquake prediction using time series magnitude data or seismic electric signals, Expert Syst. Appl. 38 (12) (2011) 15032–15039, http://dx.doi.org/10.1016/j.eswa.2011.05.043.

[16] P. Debnath, P. Chittora, T. Chakrabarti, P. Chakrabarti, Z. Leonowicz, M. Jasinski, R. Gono, E. Jasińska, Analysis of earthquake forecasting in India using supervised machine learning classifiers, Sustainability 13 (2) (2021) 971, http://dx.doi.org/10.3390/su13020971.

[17] A. Hoque, J. Raj, A. Saha, P. Bhattacharya, Earthquake magnitude prediction using machine learning technique, in: Trends in Computational Intelligence, Security and Internet of Things: Third International Conference, ICCISIoT 2020, Tripura, India, December 29-30, 2020, Proceedings 3, 2020, pp. 37–53, http://dx.doi.org/10.1007/978-3-030-66763-4.

[18] M.A. Salam, L. Ibrahim, D.S. Abdelminaam, Earthquake prediction using hybrid machine learning techniques, Int. J. Adv. Comput. Sci. Appl. 12 (5) (2021) 654–6652021, http://dx.doi.org/10.14569/IJACSA.2021.0120578.

[19] A. Joshi, B. Raman, C.K. Mohan, L.R. Cenkeramaddi, Application of a new machine learning model to improve earthquake ground motion predictions, Nat. Hazards 120 (1) (2024) 729–753, http://dx.doi.org/10.1007/s11069-023-06230-4.

[20] H. Kubo, T. Kunugi, W. Suzuki, S. Suzuki, S. Aoi, Hybrid predictor for ground-motion intensity with machine learning and conventional ground motion prediction equation, Sci. Rep. 10 (1) (2020) 11871, http://dx.doi.org/10.1038/s41598-020-68630-x.

[21] P. Zhang, X. Li, J. Chen, Prediction method for mine earthquake in time sequence based on clustering analysis, Appl. Sci. 12 (21) (2022) 11101, http://dx.doi.org/10.3390/app122111101.

[22] R. Yuan, An improved K-means clustering algorithm for global earthquake catalogs and earthquake magnitude prediction, J. Seism. 25 (3) (2021) 1005–1020, http://dx.doi.org/10.1007/s10950-021-09999-8.

[23] P. Kavianpour, M. Kavianpour, A. Ramezani, Deep multi-scale dilated convolution neural network with attention mechanism: a novel method for earthquake magnitude classification, in: 2022 8th Iranian Conference on Signal Processing and Intelligent Systems, ICSPIS, IEEE, 2022, pp. 1–6, http://dx.doi.org/10.1109/ICSPIS56952.2022.10043978.

[24] Z. Bao, J. Zhao, P. Huang, S. Yong, X. Wang, A deep learning-based electromagnetic signal for earthquake magnitude prediction, Sensors 21 (13) (2021) 4434, http://dx.doi.org/10.3390/s21134434.

[25] K. Asim, F. Martínez-Álvarez, A. Basit, T. Iqbal, Earthquake magnitude prediction in hindukush region using machine learning techniques, Nat. Hazards 85 (2017) 471–486, http://dx.doi.org/10.1007/s11069-016-2579-3.

[26] Z. Zhang, Y. Wang, A spatiotemporal model for global earthquake prediction based on convolutional LSTM, IEEE Trans. Geosci. Remote Sens. (2023) http://dx.doi.org/10.1109/TGRS.2023.3302316.

[27] K. Vardaan, T. Bhandarkar, N. Satish, S. Sridhar, R. Sivakumar, S. Ghosh, Earthquake trend prediction using long short-term memory RNN, Int. J. Electr. Comput. Eng. 9 (2) (2019) 1304–1312, http://dx.doi.org/10.11591/ijece.v9i2.pp1304-1312.

[28] R. Puthran, Spatio-temporal analysis of hybrid CNN-gru model for prediction of earthquake for disaster management, Int. J. Intell. Syst. Appl. Eng. 12 (2024) 270–281.

[29] M.W. Li, R.Z. Xu, J. Geng, W.C. Hong, H. Li, A ship motion forecasting approach based on Fourier transform, regularized Bi-LSTM and chaotic quantum adaptive WOA, Ocean Eng. 313 (2024) 119560, http://dx.doi.org/10.1016/j.oceaneng.2024.119560.

[30] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, http://dx.doi.org/10.48550/arXiv.1412.3555, arXiv preprint arXiv:1412.3555.

[31] M. Xia, H. Shao, X. Ma, C.W. De Silva, A stacked GRU-rnn-based approach for predicting renewable energy and electricity load for smart grid operation, IEEE Trans. Ind. Inform. 17 (10) (2021) 7050–7059, http://dx.doi.org/10.1109/TII.2021.3056867.

[32] Y. Peng, B.-L. Lu, Hybrid learning clonal selection algorithm, Inform. Sci. 296 (2015) 128–146, http://dx.doi.org/10.1016/j.ins.2014.10.056.

[33] L. Huang, Y. Wang, H. Bi, G. Zhu, L. Liu, W. Jiang, Initial results of atmospheric weighted mean temperature estimation with Pangu-weather in real-time GNSS PWV retrieval for China, GPS Solut. 29 (1) (2025) 48, http://dx.doi.org/10.1007/s10291-024-01807-3.

[34] Y. Fu, D. Liu, J. Chen, L. He, Secretary bird optimization algorithm: a new metaheuristic for solving global optimization problems, Artif. Intell. Rev. 57 (5) (2024) 123, http://dx.doi.org/10.1007/s10462-024-10729-y.

[35] F. Yang, M. Kefalas, M. Koch, A.V. Kononova, Y. Qiao, T. Bäck, Auto-rep: an automated regression pipeline approach for high-efficiency earthquake prediction using lanl data, in: 2022 14th International Conference on Computer and Automation Engineering, ICCAE, IEEE, 2022, pp. 127–134, http://dx.doi.org/10.1109/ICCAE55086.2022.9762437.

[36] K.M. Asim, A. Idris, T. Iqbal, F. Martínez-Álvarez, Seismic indicators based earthquake predictor system using genetic programming and AdaBoost classification, Soil Dyn. Earthq. Eng. 111 (2018) 1–7, http://dx.doi.org/10.1016/j.soildyn.2018.04.020.

[37] K.M. Asim, A. Idris, T. Iqbal, F. Martínez-Álvarez, Earthquake prediction model using support vector regressor and hybrid neural networks, PloS One 13 (7) (2018) e0199004, http://dx.doi.org/10.1371/journal.pone.0199004.