EAI Endorsed Transactions
on Scalable Information Systems

Research Article EAI.EU

Multi-strategy KOA Algorithm for Optimizing Gated Recurrent Cell Networks in Automatic Writing Scoring Method Design

Longmei Gu[1,*]

[1] General Education School Chongqing Water Resources and Electric Engineering College, Chongqing 402160, China

Abstract

INTORDUCTION: Builds an objective, robust, high-precision automatic scoring method for essays that not only improves the efficiency of exam scoring, but also provides effective feedback to help users improve their writing skills.
OBJECTIVES: Addressing the problems of current automatic writing scoring methods that fail to consider holistic and process features and lack of model accuracy.
METHODS: In this paper, a methodology approach for automatic scoring of writing based on intelligent optimization algorithm to improve recurrent neural network is proposed. Firstly, relevant features are extracted by analyzing the problem and process of automatic writing scoring; then, the gated recurrent unit network is improved by multi-strategy Keplerian optimization algorithm to construct the automatic writing scoring model; finally, the effectiveness and superiority of the proposed method is verified by simulation experiment analysis.
RESULTS: The results show that the scoring method proposed in this paper controls the scoring error within 0.04, which solves the problem of incomplete features and insufficient scoring accuracy of automatic scoring methods for writing.
CONCLUSION: The proposed algorithm can improve the accuracy and real-time performance of automatic scoring of writing questions, but the optimization efficiency needs to be further improved.

Keywords: automatic writing scoring methods, gated recurrent cell networks, Keplerian optimization algorithm, Gauss-Levy flight strategy

Received on 11 July 2023, accepted on 4 March 2024, published on 5 March 2024

doi: 10.4108/eetsis.4859

*Corresponding author. Email: wangyaee@163.com

1. Introduction

National large-scale examinations are an effective way to measure students' performance and select talents, such as the college entrance examination and the four or six levels of college English in China, and TOEFL and IELTS in foreign countries [1]. Writing, as an important part of these large-scale examinations, is mainly used to evaluate students' thinking ability and language knowledge [2]. Currently, large-scale exams consume enormous human, material and financial resources, and manual scoring is subjective and has a large error, especially the essay scoring process is long, and the current scoring method can no longer meet the social needs and the development of education reform [3]. With the rapid development of computer technology, Internet technology and artificial intelligence technology, intelligent theories continue to impact the traditional scoring model, but also bring new opportunities and prospects for the innovation of automatic scoring of essays [4]. Unlike manual scoring, automatic scoring of essays is to estimate the score of essays by constructing features that can reflect words, sentences and subjects [5]. Therefore, the construction of an objective, robust, high-precision automatic scoring method for essays not only improves the efficiency of exam scoring, but also provides effective feedback to help users improve their writing level, which is an urgent problem to be solved in the current work of education reform and innovation, and the improvement of the quality of scoring for automated marking of exams [5].

The development of intelligent technology has led to the development of automatic scoring innovation, which has also attracted the attention of experts in the field of automatic scoring, especially the research of experts and scholars focusing on the field of automatic scoring of writing [6]. Currently, the research on automatic writing scoring methods is mainly carried out from two aspects: automatic writing scoring feature extraction and automatic writing scoring model construction [7]. Writing automatic scoring feature extraction is mainly studied from the aspects of word feature extraction [8], semantic feature extraction [9], and topic feature representation [10]. Literature [11] proposes the matching rate of n-order word elements as the scoring rule, and at the same time introduces the length penalty ratio to solve the problem of high scores for short sentences; literature [12] analyzes the writing scoring features from the perspectives of literal overlap, keywords, semantics, etc., and constructs the writing automatic scoring model. Writing automatic scoring model methods include feature engineering-based machine learning methods, deep learning-based methods, and hybrid model-based methods [13]. Literature [14] uses convolutional neural network to obtain sentence features and proposes an automatic scoring method for composition by fusing topic features; Literature [15] extracts artificial features based on grammatical-syntactic rules for composition scoring and scores the composition by using a support vector machine; Literature [16] constructs an automatic scoring system for composition by extracting a variety of compositional features, such as words, sentences, and so on, and by using the random forest algorithm; Literature [17] extracts the feature set from three aspects, such as word frequency, word size, distribution location, etc., and uses random forest to score the composition; literature [18] uses a hybrid neural network model to identify and analyze the gracefully entering boards in the compositions; literature [19] firstly uses deep neural network to obtain the semantic representation of the compositions, and then, combined with manually extracted compositions features such as the number of misspelled words and the number of characters, it uses the XGboost classifier to predict and analyze the composition score. In response to the above literature analysis, the existing automatic writing scoring methods do not consider comprehensive composition scoring features, lack scoring process features, and the scoring model accuracy is lacking [20].

Intelligent optimization algorithm called modern heuristic algorithm is an algorithm with global optimization performance, versatile and suitable with parallel processing, which has a strict theoretical basis rather than simply relying on the experience of experts, and can find the optimal solution or near optimal solution within a certain time [21]. Recurrent neural network (RNN) is derived from the Hopfield network proposed by Saratha Sathasivam in 1982. Recurrent neural network, which is a fully connected neural network with the addition of backward and forward temporal relationships, can better deal with timing-related problems such as machine translation [22]. The combination of recurrent neural networks, intelligent optimization algorithms makes the accuracy of automatic writing scoring methods improve, which makes the research of automatic writing scoring methods based on intelligent optimization algorithms to improve recurrent neural networks become a hot spot of experts' research [23].

Aiming at the problems existing in the current automatic scoring methods for writing, this paper proposes an automatic scoring method for writing based on the intelligent optimization algorithm to improve the RNN network. The main contributions of this paper are: elaborating the analysis of automatic scoring process of writing questions, extracting the feature model of automatic scoring of writing, combining recurrent neural network and intelligent optimization algorithm, proposing the automatic scoring method of writing based on the optimization of gated recurrent unit network with improved KOA algorithm, and verifying the method of this paper through experiments that the method of this paper has a higher scoring accuracy and real-time performance.

2. Pre-processing of data for automatic scoring of writing questions

2.1. Analysis of the automatic scoring process for writing questions

When using machine learning models for automatic scoring, scoring feature extraction is carried out on the scoring dataset of the corresponding writing question type, and the scoring feature data obtained is used to train the original model to obtain the scoring model of the writing question type [24]. When auto-scoring, the same scoring feature extraction is performed on the text to be scored, and the obtained feature data are input to the corresponding auto-scoring model to obtain the auto-scoring results, the specific process is shown in Figure 1.

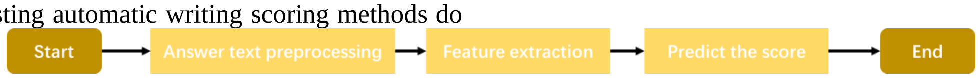


Figure 1 Automated scoring process

2.2. Scoring feature extraction

The automatic scoring features of essays are mainly divided into two categories: non-text features and text features, which are analyzed as shown in Figure 2.

Non-textual features

Sub-textual features mean that the extraction of features does not take into account the meaning of text words, but only counts the features such as word length, lexical properties, the organization of the text and chapter structure, which mainly reflect the quality of the text from the surface features of the text [25]. This section extracts features from two perspectives: lexical features and syntactic features.

a.Lexical features refer to the level extraction features of the words in the text, mainly considering the length of the words, lexical properties, spelling errors and other features, lexical features include the number of words L1, the average length of the words L2, the statistics of the word types L3, and the number of spelling errors words L4;

b. Syntactic features mainly include the length (number of words) of the sentences in the text, sentence rank and other features, specifically, the average length of sentences S1, the height and depth of the syntactic tree S2, the average number of subordinate clauses S3, and the number of sentences S4.

Text features

Text features are the meaning of the words in the article considered when extracting features, including article representation vector, vector space dimensionality reduction, and topic distribution set. In this paper, the topic distribution set is used for article representation, i.e., LDA (Latent Dirichlet Allocation).

Text topic modeling refers to modeling large-scale document corpus with statistical methods to mine the implicit topic information in the documents [26].LDA topic model was proposed by Blei et al. in 2002 as a probabilistic generative model.The LDA features are denoted as T1, and the specific model is as follows:

$$\vartheta_{\tilde{m},k} = \frac{n_{\tilde{m}}^{(k)} + \alpha_k}{\sum_{k=1}^{K} n_{\tilde{m}}^{(k)} + \alpha_k} \tag{1}$$

$\vartheta_{\tilde{m},k}$ $n_{\tilde{m}}^{(k)}$ α_k where denotes the new text topic distribution, and and are LDA hyperparameters.

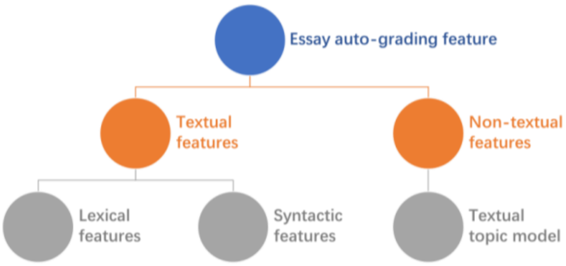


Figure 2. Writing automated scoring signs extraction analysis map

3. Door control cycle unit network

Recurrent Neural Network (RNN) provides an effective solution to the time series prediction problem, but suffers from the problem of gradient explosion and gradient vanishing when dealing with long term time series problems.LSTM and GRU, as the advanced version of RNN, effectively solve the gradient problem of RNN. Compared with LSTM, GRU it has a simpler structure, fewer parameters, and introduces the gate structure, which consists of update gates and reset gates [27].The schematic diagram of GRU network is shown in Fig. 3, and the specific model structure is as follows:

$$r_t = \sigma(W_{hr} h_{t-1} + W_{xr} x_t + b_r) \tag{2}$$

$$\tilde{h}_t = \tanh(W_{r\tilde{h}} \bullet (r_t * h_{t-1}) + W_{x\tilde{h}} x_t + b_{\tilde{h}}) \tag{3}$$

r_t h_{t-1} Where is the reset gate, which determines how much of 's historical memory is retained. $\tilde{h}_t$ is the latest information of the Caddidate hidden layer at the current moment, is the hidden layer information of the cell state at and respectively, , , and are the right to reset. h_{t-1} h_t $t-1$ t $W_{r\tilde{h}}$ $W_{x\tilde{h}}$ W_{xr} W_{hr} b_r $b_{\tilde{h}}$ The information of is the hidden layer information of the cell state at the moment of and respectively, , , , are the weights, , are the biases.

$$z_t = \sigma(W_{hz} h_{t-1} + W_{xz} x_t + b_z) \tag{4}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{5}$$

Where W_{hz} W_{xz} b_z are the weights and is the bias. z_t h_{t-1} h_t is a forgetting gate, the function is to combine the input hidden layer information at the previous moment with the candidate hidden layer information at the current moment to get the output cell hidden layer information . $z_t = 0$ h_{t-1} $z_t = 1$ h_t When , the hidden layer directly outputs the hidden layer information of the previous moment , and when , the candidate hidden layer directly outputs the current hidden layer information .

$$y_t = \sigma(W_{yt} h_t) \tag{6}$$

W_{yt} h_t Where represents the weight of the current hidden layer output in the middle of the final output layer.

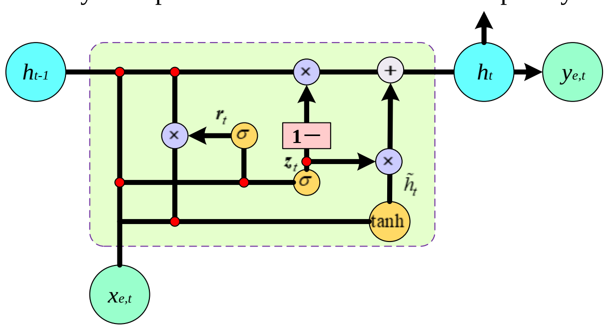


Figure 3. GRU network

4. Multi-Strategy Keplerian Optimization Algorithm

4.1. Keplerian optimization algorithm

The Keplerian optimization algorithm [28] is an intelligent optimization algorithm based on the principle of celestial motion, which simulates the laws of motion of the planets around the sun in order to solve the optimization problem, and the simulation principle is shown in Figure 4.

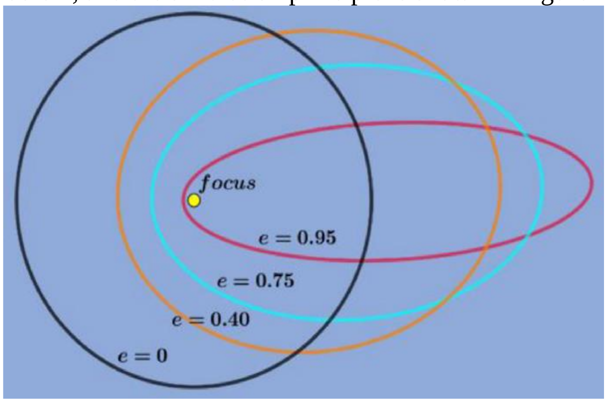


Figure 4. Search mechanism inspired by the Keplerian optimization algorithm

The Kepler Optimization Algorithm (KOA) is a physics-based meta-heuristic algorithm [28] inspired by Kepler's laws of planetary motion that predicts the position and velocity of a planet at any given time. In KOA, each planet and its position is a candidate solution, which is randomly updated during the optimization process with respect to the best solution (Sun) so far.KOA allows for a more efficient exploration and utilization of the search space, since the candidate solutions (planets) behave differently from the Sun at different times.

The optimization search strategy and steps of the KOA algorithm are as follows:

Step 1: Initialize the population.

At the beginning of the optimization process, *the N* planets are randomly initialized with the following initialization formula:

$$X_i^j = X_{i,down}^j + rand \times \left(X_{i,up}^j - X_{i,down}^j \right) \tag{7}$$

X_i^j $X_{i,down}^j$ $X_{i,up}^j$ $rand$ Among them, denotes the *jth* dimensional orbital position of the *ith* planet; and denote the lower and upper limits of the planetary orbital positions respectively; denotes a random number in the range of 0 to 1; the value of *i* ranges from 1 to *N*; and the value of *j* ranges from 1 to *d*. The following are some examples of random numbers.

Step 2: Define gravity.

Gravity is the condition that controls the orbits of the planets around the sun. The closer a planet is to the Sun, the greater its orbital velocity. X_s X_i According to the law of gravity, the gravitational force equation between the optimal solution (the Sun) and the candidate solution (the planet) is as follows:

$$F_{g_i}(t) = e_i \times \mu(t) \times \frac{\bar{M}_s \times \bar{m}_i}{\bar{R}_i^2 + \varepsilon} + r_1 \tag{8}$$

$\bar{M}_s$ $\bar{m}_i$ M_s m_i M_s m_i X_s X_i e_i $\mu(t)$ r_1 ε $\bar{R}_i$ X_s X_i Where and are the normalized values of and respectively; and denote the masses of the optimal solution (Sun) and the candidate solution (planet) ; denotes the eccentricity of the planetary orbits; denotes the gravitational constant, whose computation is related to the number of iterations; is the randomly generated value between 0 and 1, which is used to diversify the values of gravitational force during the optimization process; denotes to prevent the denominator to be small as 0; denotes the normalized value of the Euclidean distance between the optimal solution (Sun) and the candidate solution (planet) . Euclidean distance normalization value, calculated as follows:

$$R_i(t) = \left\| X_s(t) - X_i(t) \right\|_2 = \sqrt{\sum_{j=1}^{d} \left(X_{sj}(t) - X_{ij}(t) \right)^2} \tag{9}$$

X_s X_i The masses of the optimal solution (sun) and the candidate solution (planet) are calculated as follows:

$$M_s = r_2 \frac{fit_s(t) - fit_{worst}(t)}{\sum_{k=1}^{N} \left(fit_k(t) - fit_{worst}(t) \right)} \tag{10}$$

$$m_i = \frac{fit_i(t) - fit_{worst}(t)}{\sum_{k=1}^{N} \left(fit_k(t) - fit_{worst}(t) \right)} \tag{11}$$

$fit_s(t)$ $fit_{worst}(t)$ r_2 where denotes the optimal fitness value for the *tth* iteration, denotes the worst fitness value for the *tth* iteration, and denotes a randomly generated number between 0 and 1 for dispersing the quality values of various planets.

$\mu(t)$ The universal gravitational constant decreases exponentially as the number of iterations increases:

$$\mu(t) = \mu_0 \times \exp\left(-\gamma \frac{t}{T_{\max}} \right) \tag{12}$$

γ μ_0 t $T_{\max}$ Where is a constant, is the initial value, and are the current and maximum iterations respectively.

Step 3: Calculate the speed of the planets.

The orbital speed of a planet around the Sun depends on the position of the planet in relation to the Sun. As the planet moves closer to the Sun, the planet's velocity increases, preventing it from flying towards the Sun because of the Sun's strong pull; as the planet moves away from the Sun, the planet's velocity slows down because the Sun's gravitational pull becomes weaker. The formula for calculating planetary velocity is divided into two parts, the first part is the strategy for bringing planets close to the sun that allows for population diversification, and the second part is the escape strategy that prevents falling into a local optimum, as described below:

$$V_i(t)=\begin{cases} l\times(2r_4X_i-X_b)+\ddot{l}\times(X_a-X_b)+(1-R_{i-norm}(t)) & R_{i-norm}(t)\le 0.5 \\ \quad \times F\times U_1\times r_5\times(X_{up}-X_{down}) & \\ r_4\times L\times(X_a-X_i)+(1-R_{i-norm}(t))\times F\times U_2\times r_5 & else \\ \quad \times(X_{up}-X_{down}) & \end{cases} \tag{13}$$

$$l=U\times H\times L \tag{14}$$

$$L=\left(\mu(t)\times(M_s+m_i)\left|\frac{2}{R_i(t)+\varepsilon}-\frac{1}{a_i(t)+\varepsilon}\right|\right)^{0.5} \tag{15}$$

$$H=r_3\times(1-r_4)+r_4 \tag{16}$$

$$U=\begin{cases}0 & \vec{r}_5\le\vec{r}_6\\ 1 & else\end{cases} \tag{17}$$

$$F=\begin{cases}1 & r_4\le 0.5\\ -1 & else\end{cases} \tag{18}$$

$$\ddot{l}=(1-U)\times\vec{H}\times L \tag{19}$$

$$\vec{H}=r_3\times(1-\vec{r}_5)+\vec{r}_5 \tag{20}$$

$$U_1=\begin{cases}0 & \vec{r}_5\le r_4\\ 1 & else\end{cases} \tag{21}$$

$$U_2=\begin{cases}0 & r_3\le r_4\\ 1 & else\end{cases} \tag{22}$$

$V_i(t)$ $\vec{r}_5$ $\vec{r}_6$ r_3 r_4 X_a X_b $a_i(t)$ where denotes the velocity of the *ith* planet; and denote randomly generated vectors between 0 and 1, and denote randomly generated vectors between 0 and 1, respectively; and denote randomly selected vectors from the population, respectively; and denotes the elliptical orbital half-length axis of the *ith* planet at time t, defined as follows:

$$a_i(t)=r_3\times\left[T_i^2\times\frac{\mu(t)\times(M_s+m_i)}{4\pi^2}\right]^{\frac{1}{3}} \tag{23}$$

$$T_i=|r| \tag{24}$$

T_i r where denotes the orbital period of planet i and denotes normally distributed random values.

$$R_{i-norm}(t)=\frac{R_i(t)-\min(R(t))}{\max(R(t))-\min(R(t))} \tag{25}$$

$R_{i-norm}(t)\le 0.5$ Where, if , the planet is close to the sun, increase the velocity to prevent crashing into the sun. Otherwise decrease the speed.

Step 4: Escape from the local optimum.

In the solar system, most planets will rotate counterclockwise around their own axis near the sun, and then some planets adopt a clockwise rotation pattern. F In order to escape the local optimum, KOA uses the flag to change direction to simulate a clockwise rotation pattern to facilitate a better search of space.

Step 5: Update the positions of the planets.

In order to balance the exploration and exploitation phases, updating the positions of the planets on an existing basis should take into account not only the velocity of the planets, but also the gravitational effect of the Sun. The velocity of the planet represents the optimized exploitation phase, which facilitates the planet to fall into the local optimum when approaching the Sun; the gravitational effect of the Sun represents the optimized exploration phase, which facilitates the planet to capture the optimum value. The updating formula of the planets is as follows:

$$X_i(t+1)=X_i(t)+F\times V_i(t)+\left(F_{g_i}(t)+|r|\right)\times U\times(X_s(t)-X_i(t)) \tag{26}$$

Step 6: Update the information on the distances of the planets from the Sun.

In order to further improve the exploration and exploitation behavior operation of planets, the KOA algorithm utilizes a time-varying strategy to update the distance information between planets and the sun. When the planet is close to the Sun, the KOA focuses on optimizing exploration; when the Sun is far away, the KOA implements exploitation optimization with the following strategy formulation:

$$X_i(t+1) = X_i(t)\times U_1 + (1-U_1)\times\left(\frac{X_i(t)+X_s+X_a}{3} + h\times\left(\frac{X_i(t)+X_s+X_a(t)}{3} - X_b(t)\right)\right) \tag{27}$$

h where can balance exploration and exploitation behavioral operations:

$$h = \frac{1}{e^{\eta r}} \tag{28}$$

$$\eta = (a_2 - 1)\times r_4 + 1 \tag{29}$$

$$a_2 = -1 - 1\times\left(\frac{t\% \frac{T_{\max}}{\overline{T}}}{\frac{T_{\max}}{\overline{T}}}\right) \tag{30}$$

r η a_2 Among them, denotes the randomly selected values of normal distribution; denotes the reduction factor from 1 to -2; denotes the cyclic control parameter, which reduces the change from -1 to -2, and the specific change simulation diagram is shown in Fig. 5.

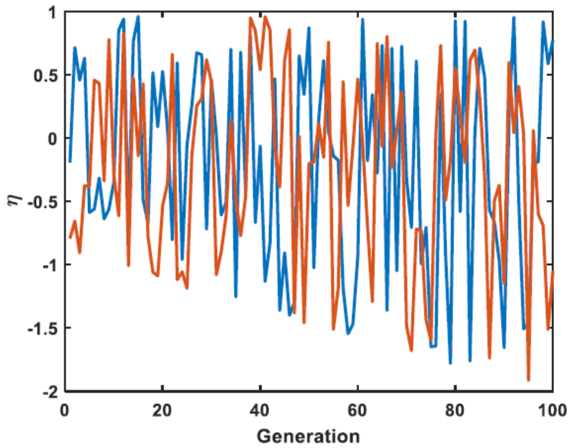


Figure 5. Simulation of parameter variation of Keplerian optimization algorithm

Step 7: Elite Selection Strategy.

To ensure that the optimal value is retained, the KOA employs an elite selection strategy, calculated as follows:

$$X_{i,new}(t+1) = \begin{cases} X_i(t+1) & f\left(X_i(t+1) \le f\left(X_i(t)\right)\right) \\ X_i(t) & else \end{cases} \tag{31}$$

Step 8: Determine whether the KOA algorithm has reached the maximum number of iterations. If it reaches, output the optimal solution at the optimal value; otherwise, restart step 2.

4.2. Improvement strategies

In order to increase the diversity of the KOA algorithm and overcome the problem of the algorithm falling into local optimum, this paper chooses two strategies to improve the KOA algorithm.

Bernoulli mapping strategy

In order to increase the initialization diversity of the KOA algorithm, in this paper, we choose the Bernoulli mapping strategy to initialize the population.Bernoulli mapping is a kind of chaotic mapping strategy, which is mainly used to provide random chaotic sequences.The expression of Bernoulli mapping is as follows:

$$Z_{k+1} = \begin{cases} Z_k/(1+\rho) & Z_k \in (0, 1-\rho] \\ (Z_k - 1 + \rho)/\rho & Z_k \in (1-\rho, 1) \end{cases} \tag{32}$$

Z_k ρ Where, is the kth generation sequence value and is the control quantity, which generally takes the value of 5, the strategy will have better traversal.

Gauss-Levy flight strategy

When updating the positions of the planets, the KOA algorithm can easily fall into local optimality. In order to avoid falling into local optimality, this section introduces the Gaussian-Levy perturbation strategy, and after the Gaussian random variation or Levy flight strategy, the individual positions of the planets are updated as follows:

1) Gaussian randomized wandering strategy

$$X_i(t+1) = X_i(t) + F\times V_i(t) + \left(F_{g_i}(t) + |r|\right)\times randn\cdot\times\left(X_s(t) - X_i(t)\right) \tag{33}$$

$randn$ Where, denotes the random parameters that obey normal distribution. This strategy mainly utilizes the global optimal solution and the individual optimal solution to update the individual position, which accelerates the search speed and search efficiency.

2) Levy Flight Strategy

Levy flights have small flight steps over long periods of time and occasionally produce longer flight steps to increase flight diversity.The specific formulas for the Levy flight model are as follows:

$$X_i(t+1) = X_s + randn\cdot Levy\left(X_i(t)\right) + randn\cdot\left|X_l(t) - X_i(t)\right| \tag{34}$$

$$Levy\left(X_i(t)\right) = \alpha\cdot s\cdot\left(X_s(t) - X_i(t)\right) \tag{35}$$

$X_i(t+1)$ $t+1$ i $Levy(\cdot)$ α $[-1,1]$ s Where denotes the th planetary individual of the th generation, denotes the Levy flight model, denotes the scale factor, which takes the value of ; is the random wandering step, which is calculated as follows:

$$s = \frac{u}{|v|^{1/\beta}} \tag{36}$$

$$\sigma_u = \left[\frac{\Gamma(1+\beta)\cdot \sin(\pi\cdot\beta/2)}{\Gamma((1+\beta)/2)\cdot\beta\cdot 2^{(\beta-1)/2}} \right]^{1/\beta} \tag{37}$$

$$\sigma_v = 1 \tag{38}$$

u v $u \sim N(0,\sigma_u^2)$ $v \sim N(0,\sigma_v^2)$ $\Gamma(\cdot)$ Where, and are parameters obeying normal distribution, i.e. , , are gamma functions.

4.3. Multi-Strategy Improved KOA Algorithm Flow

According to the improvement strategy, the step-by-step flow of the KOA algorithm is shown in Figure 6.

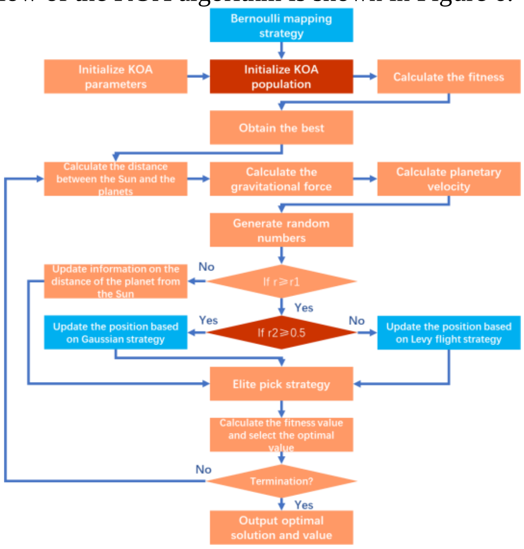


Figure. 6 Flowchart of MSKOA algorithm

5. Process of automatic scoring method for writing question types based on multi-strategy KOA algorithm to improve GRU network

5.1. Decision Variables and Objective Functions

In order to overcome the gradient optimization of gated recurrent units falling into local optima, a multi-strategy KOA algorithm based optimization of gated recurrent unit neural network (MSKOA-GRU) is proposed in this section.

The traditional gated recurrent unit neural network adopts the gradient descent method, which can easily make the optimization of the parameters of the gated recurrent unit neural network fall into the local optimum. l W_{hr} W_{xr} $W_{\tilde{r}h}$ $W_{x\tilde{h}}$ W_{hz} W_{xz} W_{yt} b_r $b_{\tilde{h}}$ b_z $10\times l$ In order to overcome the above problems, the multi-strategy KOA algorithm is used to optimize the GRU neural network parameters, i.e., to optimize the weights and biases of the neural network, and in this paper, we use the real number coding method to encode the parameters, and the coding region is mainly divided into the weight value and bias, The coding dimensions for the 10-dimensional inputs of the GRU neural network units (, , , , , , , , , and) are .

In order to improve the convergence optimization efficiency of the multi-strategy KOA algorithm to optimize the GRU parameters, the root-mean-square error function is used as the objective function of the MSKOA-GRU algorithm, which is calculated as follows:

$$\min RMSE = \sqrt{\left(\sum_{i=1}^{M}(\hat{y}_i - y_i)^2\right)\Big/ M} \tag{39}$$

y_i $\hat{y}_i$ Where is the actual value and is the predicted value.

5.2. Methodological process

Combining the MSKOA algorithm and gated recurrent unit neural network, an automatic scoring method based on the MSKOA algorithm with improved gated recurrent unit neural network is proposed in this section. This automatic scoring model mainly takes the automatic scoring features of writing questions as input and the scoring values as output, and constructs the mapping relationship between the automatic scoring features and the scoring values. The flowchart of the automatic scoring method for writing question types based on the MSKOA algorithm to improve the gated recurrent unit neural network is shown in Figure 7. The specific steps are as follows:

Step 1: Acquire the essay collection data and extract the writing automatic scoring feature samples; pre-process the acquired samples with sparse smoothing data processing method; normalize the raw data with Z-Score method and divide the data into testing set, validation set and training set;

Step 2: The initial parameters of the GRU network are encoded using the MSKOA algorithm based on the Bernoulli mapping strategy, and the algorithm parameters such as the population parameters and the number of iterations are initialized at the same time; the population is initialized, and the objective function value is calculated;

Step 3: Update the planets' velocities and positions based on their current positions and velocities using Kepler's laws and the gravitational formula, Gauss-Levy flight strategy;

Step 4: In each iteration, it is necessary to compare the objective function value of each planet with the objective function value of the current global optimal solution and update the global optimal solution;

Step 5: Judge whether the termination condition is satisfied. If it is satisfied, exit the iteration, output the optimal GRU parameters and execute step 6, otherwise continue to execute step 3;

Step 6: Decode the MSKOA based optimized GRU parameters to obtain the network parameters;

Step 7: Construct the MSKOA-GRU analysis model, train the analysis model using the training set, input the test set into the model, and obtain the evaluation results and error analysis results.

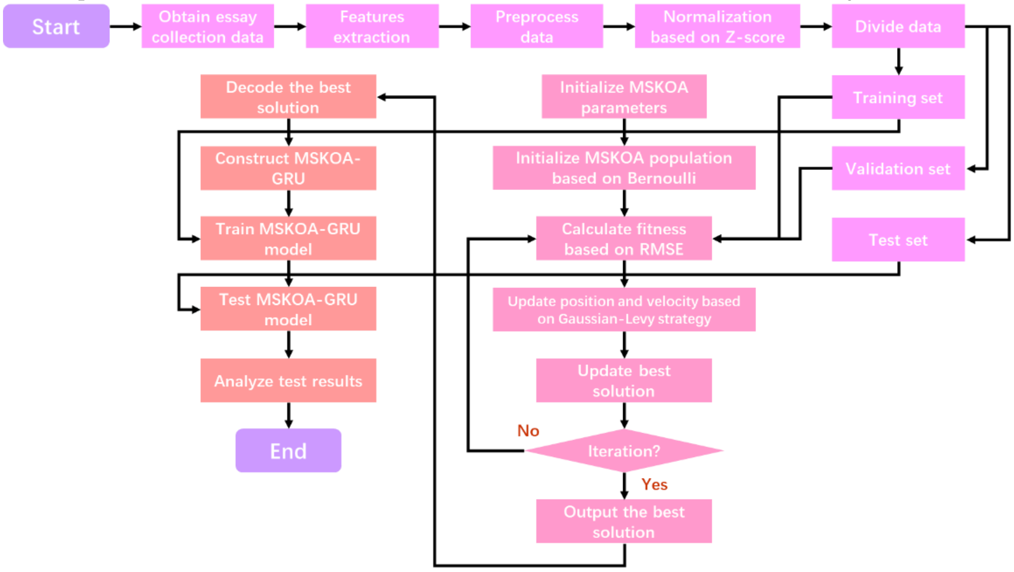


Figure 7. Automatic scoring method for writing questions based on MSKOA-GRU

6. Experiments and Analysis of Results

In order to verify the effectiveness of the automatic scoring method proposed in this paper, the algorithms of LSTM, GRU, KOA-LSTM, MSKOA-LSTM, and KOA-GRU are selected for comparison, and the specific parameter settings of each algorithm are shown in Table 1.The training data, validation data, and test data are mainly from the essay collection data samples, with the training sample being 600, the number of validation samples 36, and the test sample The experimental simulation environment is Windows 10, and the programming language Matlab.

Table 1 Parameter settings for the method of evaluating the effectiveness of English language teaching

arithmetic	parameterization
LSTM	The number of hidden layer nodes is 50 and Adam optimization adjusts the weights
GRU	The number of hidden layer nodes is 50 and Adam optimization adjusts the weights
KOA-LSTM	The number of hidden nodes is 50 and the KOA population is 50
MSKOA-LSTM	The number of cryptic nodes is 50 and the MSKOA population is 50
KOA-GRU	The number of hidden nodes is 50 and the KOA population is 50
MSKOA-GRU	The number of cryptic nodes is 50 and the MSKOA population is 50

6.1. Feature Extraction Analysis

In order to extract the principal components of the automatic scoring features of writing questions, this paper adopts the principal component analysis method to carry out feature extraction analysis, and the specific analysis results are shown in Figure 8. As can be seen from Figure 8, the principal components of the writing automatic scoring features include features such as L2, L1, L4, S2, S1, L3, T1, and so on.

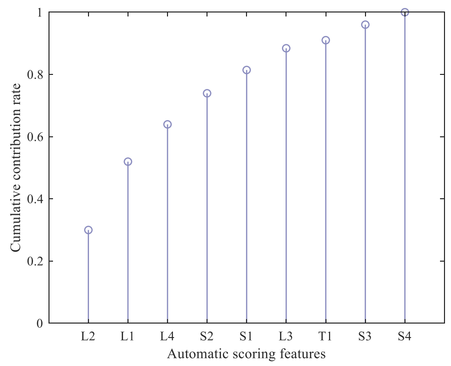
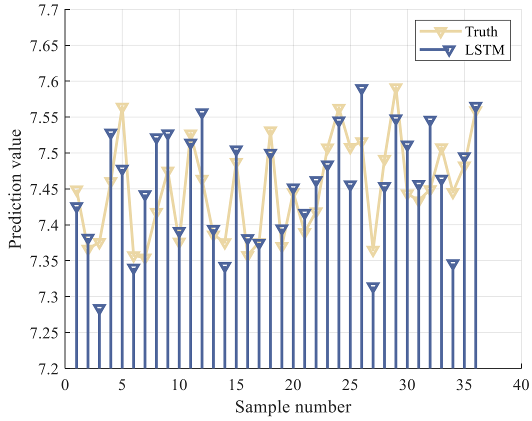


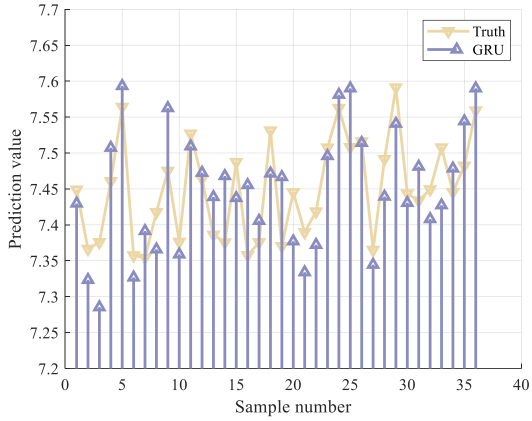
Figure 8. Feature principal component analysis

6.2. Model Performance Analysis and Comparison

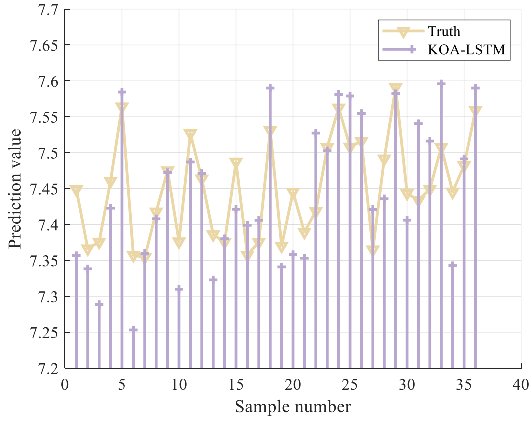
In order to verify the effectiveness and superiority of the automatic writing scoring method based on the MSKOA-GRU algorithm, KOA-GRU was compared with five other models, and the scoring results of each model are shown in Figures 9 and 10. Figure 9 gives the automatic scoring method scoring values for writing based on each algorithm. As can be seen from Figure 9, the automatic scoring method scoring values based on the MSKOA-GRU algorithm are closest to the true values.



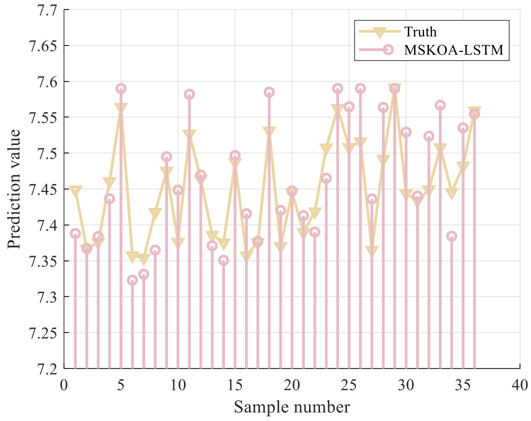
(a) LSTM



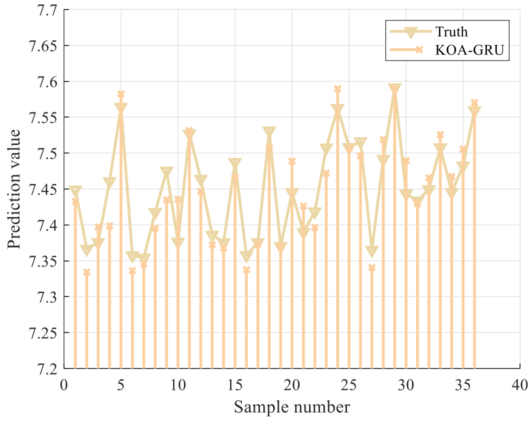
(b) GRU



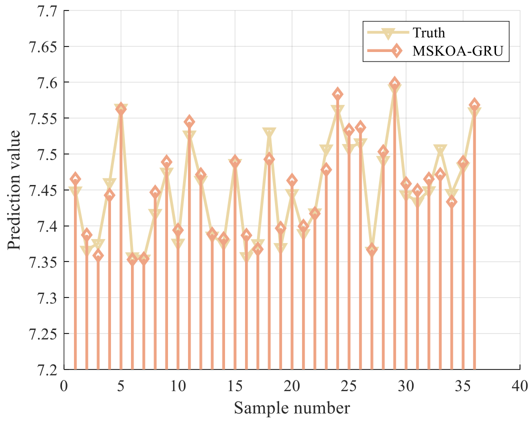
(c) KOA-LSTM



(d) MSKOA-LSTM



(e) KOA-GRU



(f) MSKOA-GRU

Figure 9. Scoring values of automatic writing scoring methods based on different algorithms

The relative error between the automatic scoring values of writing based on each algorithm and the true value is given in Figure 10. From Figure 10, it can be seen that the absolute value of the relative error of the automatic scoring values of writing based on the MSKOA-GRU algorithm is controlled within the range of 0.04, and the absolute value of the relative errors of LSTM, GRU, KOA-LSTM, MSKOA-LSTM, and KOA-GRU are controlled within the range of 0.12, 0.1, 0.13, 0.7, and 0.63, respectively. This shows that the accuracy of automatic writing scoring method based on MSKOA-GRU algorithm is better than other algorithms. In conclusion, the error of the automatic writing scoring method based on the MSKOA-GRU algorithm is minimized overall.

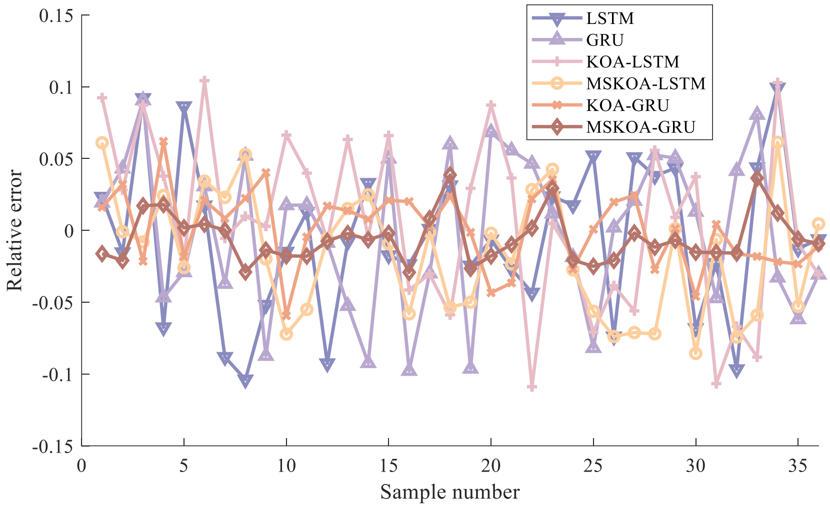
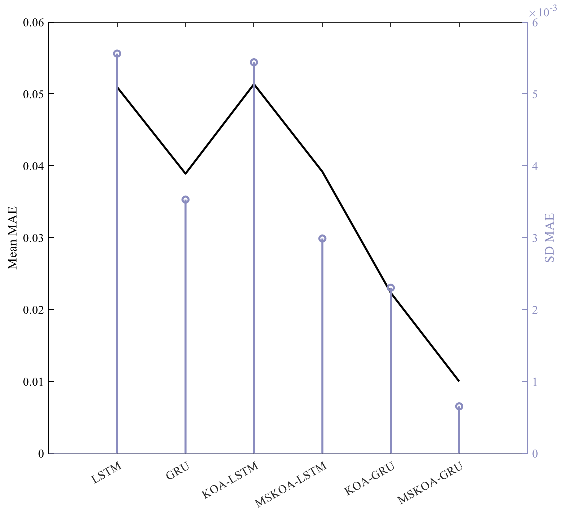
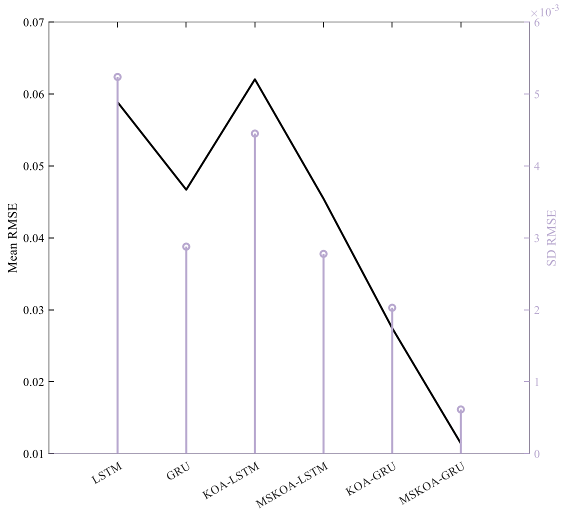


Figure 10 Relative error between automatic scoring values and true values of writing based on each algorithm

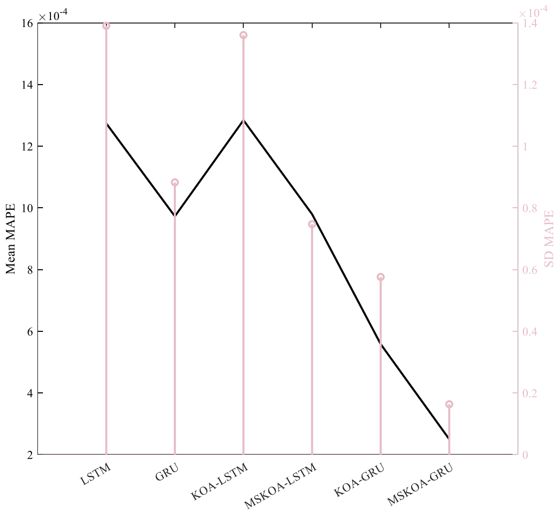
In order to further verify the superiority of the automatic writing scoring method based on the MSKOA-GRU algorithm, MSKOA-GRU was compared with the other five methods in terms of error and timeliness performance, and the results of MAE, RMSE, MAPE, R , and scoring time of the algorithms are shown in Figure 11. Figure 11 gives a comparison of the scoring performance of the automatic writing scoring methods based on each algorithm. From Figure 11, it can be seen that in terms of mean value, the automatic writing scoring methods based on MSKOA-GRU algorithm have the smallest MAE, RMSE, MAPE, the smallest time spent, the largest R , and the rank of 1. In terms of standard deviation, the automatic writing scoring methods based on MSKOA-GRU algorithm have the smallest MAE, RMSE, MAPE, R , and the smallest time spent, and the rank of 1.



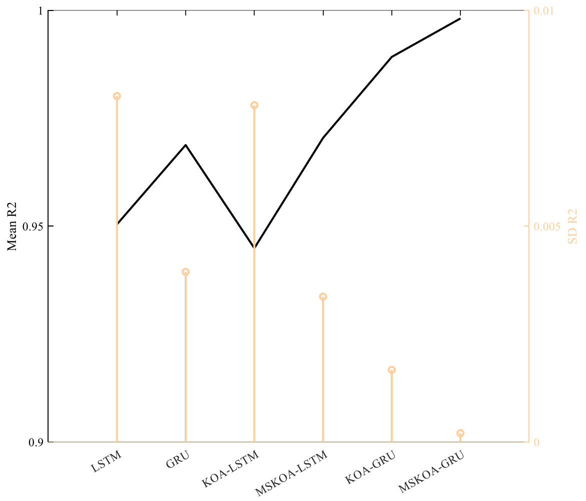
(a) MAE



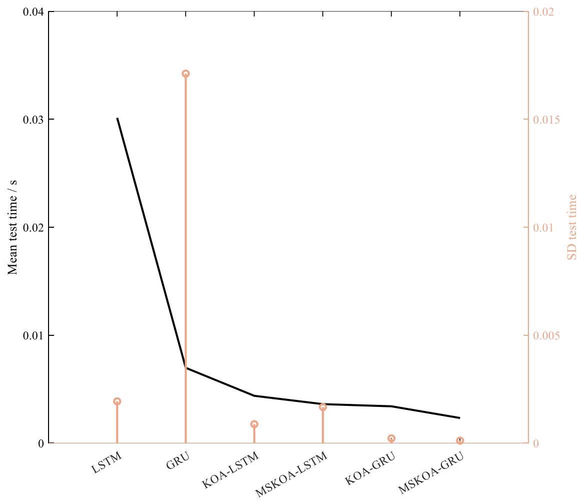
(b) RMSE



(c) MAPE



(d) R^2



(e) Scoring time

Figure 11. Performance comparison of automatic scoring of writing based on each algorithm

7. Conlusion

In order to improve the accuracy and real-time performance of automatic scoring of writing question types, this paper proposes a method for automatic scoring of writing based on multi-strategy KOA algorithm optimization to improve the gated recurrent unit neural network. The method analyzes the automatic scoring problem of writing question types and extracts the scoring features. Combined with the improved Keplerian optimization algorithm, the parameters of the gated recurrent unit neural network are optimized to improve the accuracy and real-time performance of the automatic scoring method. The writing scoring data is utilized for analysis, and the results show that the method proposed in this paper improves the accuracy of automatic scoring under the condition of real-time performance.

Although the algorithm proposed in this paper can improve the accuracy and real-time performance of automatic scoring of writing questions, the optimization efficiency needs to be further improved. In the subsequent research, the GRU will be improved through the attention mechanism to further improve the execution efficiency of the algorithm.

References

[1] Wang H.W. Using Coh-Metrix to automatically assess students' writing proficiency[J]. Overseas English, 2011(14):2.

[2] Spring R , Johnson M .The possibility of improving automated calculation of measures of lexical richness for EFL writing: a comparison of the LCA, NLTK and SpaCy tools[J].System, 2022, 106:102770-.

[3] Dorst A G , Valdez S , Bouman H .Machine translation in the multilingual classroom:How, when and why do humanities students at a Dutch university use machine translation?[J].Translation and Translanguaging in Multilingual Contexts, 2022, 8(1):49-66.

[4] Kashihara K .Automatic Discrimination of Task Difficulty Predicted by Frontal EEG Activity During Working Memory Tasks in Young and Elderly Drivers [J].International Journal of Information Technology & Decision Making, 2022.

[5] Kasparaitis P .Automatic Transliteration of Polish and English Proper Nouns into Lithuanian[J].Inf. Technol. Control. 2023, 52:128-139.

[6] Nakayama Y , Sato M , Okamoto M ,Kondo Y, Tamura M, Minagawa Y. Deep learning-based classification of adequate sonographic images for self-diagnosing deep vein thrombosis[J].PLOS ONE, 2023, 18.

[7] Gondane S .Design & Development of Smart Fire Safety System for Vehicles Using IOT[J]. Engineering Technology, 2023.

[8] Venugopal M , Sharma V K , Sharma K .Web information mining and semantic analysis in heterogeneous unstructured text data using enhanced latent Dirichlet allocation[J].Concurrency and computation: practice and experience, 2023.

[9] Fery L , Dubrulle B , Podvin B ,Pons F, Faranda D. Learning a Weather Dictionary of Atmospheric Patterns Using Latent Dirichlet Allocation[J]. Geophysical Research Letters, 2022(9):49.

[10] Jesus Serrano-Guerrero,Mohammad Bani-Doumi, Romero F P ,Olivas J A.A fuzzy aspect-based approach for recommending hospitals[J].International Journal of Intelligent Systems, 2022, 37(4):2885-2910.

[11] Zhou S , Kan P , Huang Q , Silbernagel J. A guided latent Dirichlet allocation approach to investigate real-time latent topics of Twitter data during Hurricane Laura:[J].Journal of Information Science, 2023, 49(2):465-479.

[12] Ramathulasi T , Rajasekharababu M .Augmented latent Dirichlet allocationmodel viaword embedded clusters for mashup service clustering[J]. Concurrency and computation: practice and experience, 2022.

[13] Kang Y , Ou R , Zhang Y ,Li H, Tian S. PG-CODE:Latent Dirichlet Allocation Embedded Policy Knowledge Graph for Government Department Coordination[J] . . Journal of Tsinghua University: Natural Science English Edition, 2022(004):027.

[14] Shafi N , Chachoo M A .Query intent recognition by

integrating latent dirichlet allocation in conditional random field[J]. of Information Technology, 2023.

[15] Ranjithkumar S , Pandian S C .Fuzzy Based Latent Dirichlet Allocation for Intrusion Detection in Cloud Using ML[J]. Computers, Materials and Continuum (English), 2022(3):17.

[16] Shafi N , Chachoo M A .Query intent recognition by integrating latent dirichlet allocation in conditional random field[J]. of Information Technology, 2023.

[17] Wang H , Hong Z , Hong M .Research on product recommendation based on matrix factorization models fusing user reviews[J].Applied Soft Computing. 2022(123-):123.

[18] Ujlayan A , Sharma M .An analysis of employee skills and potency using machine learning[J]. 2022.

[19] Zeng D , Zhao J , Zhang W ,Zhou Y. User-interactive innovation knowledge acquisition model based on social media[J].Information Processing & Management: Libraries and Information Retrieval Systems and Communication Networks: An International Journal, 2022(3):59.

[20] Jiang L , Zhang T , Huang T .Empirical Research of Hot Topic Recognition and its Evolution Path Method for Scientific and Technological Literature[J] . .Journal of Advanced Computatioanl Intelligence and Intelligent Informatics, 2022(3 TN.156):26.

[21] Zeng L , Zheng Z , Lian X , Zhang K, Zhu M, Zhang K. Intelligent optimization method for the dynamic scheduling of hot metal ladles of one-ladle technology on ironmaking and steelmaking interface in steel plants[J]. Journal of Mineral Metallurgy and Materials:English Edition, 2023, 30(9):1729-1739.

[22] J. Liu, W. X. Zhang, Y. G. Li, Y. Chao Zhang, J. Zhou. Research on line consistency measurement based on twin neural network[J]. Journal of Beijing Institute of Technology, 2022, 42(6):9.

[23] Tian D , Liu H , Chen S ,Li M, Liu C. Human Error Analysis for Hydraulic Engineering: Comprehensive System to Reveal Accident Evolution Process with Text Knowledge[J].Journal of construction engineering and management, 2022.

[24] Zhou Q , Dai H , Hu Z ,Liu Y, Yang G. Accuracy-first and efficiency-first privacy-preserving semantic -aware ranked searches in the cloud[J].International Journal of Intelligent Systems, 2022, 37:9213 - 9244.

[25] Fernandez J , Song Y , Padua M ,Liu P. A Framework for Urban Parks: Using Social Media Data to Assess Bryant Park, New York[J].Landscape journal, 2022(1):41.

[26] Ning W , Liu J , Xiong H .Knowledge discovery using an enhanced latent Dirichlet allocation-based clustering method for solving on-site assembly problems[J].Robotics and Computer-Integrated Manufacturing, 2022, 73(3):102246.

[27] JIANG Jian-Xiang, YANG Ping, GUAN Yu-Da Yang Kang, LIU Lu-Yao. A short-term load forecasting method for industrial customers based on VMD-GRU-EC[J]. Electrical Automation, 2023, 45(5):34-37.

[28] Abdel-Basset M , Mohamed R , Azeem S A A ,Jameel M.Kepler optimization algorithm: a new metaheuristic algorithm inspired by Kepler's laws of planetary motion[J].Knowledge-based systems, 2023.