[1]Nili Guo

Research on Artificial Intelligence-Enabled Animation Script Creation

Journal of Electrical Systems

Abstract: - With the development of artificial intelligence technology, animation script creation has ushered in new opportunities. This study investigates the utilization of an algorithm based on LSTM and attention mechanisms to enhance the automation level of animation script creation. By integrating the advantages of LSTM in capturing temporal sequence dependencies and the superiority of the attention mechanism in processing long sequences, a novel script generation algorithm is proposed. The research results demonstrate significant advantages of this algorithm in terms of coherence, emotional expression, grammatical correctness, and plot coherence of the generated script content. Both in terms of BLEU score and ROUGE score, the algorithm performs exceptionally well after initial iterations and multiple iterations, reaching 0.50 and 0.55 respectively. In comparative experiments regarding grammatical errors, plot jumps, and other aspects, the sentences generated by the research model algorithm significantly outperform other algorithms, exhibiting higher accuracy and coherence. Overall, the algorithm based on LSTM and attention mechanisms can effectively improve the quality of animation script creation, providing new methods and ideas for intelligent script generation, which holds significant theoretical and practical implications.

Keywords: artificial intelligence; LSTM; natural language processing; attention mechanism; text generation

INTRODUCTION

With the rapid development of artificial intelligence technology, the application of intelligent algorithms in various fields has gradually gained popularity. In the film and animation industry, script creation, as a crucial link, directly impacts the overall quality of the work [1]. However, traditional script creation relies heavily on the screenwriter's inspiration and experience, resulting in issues such as long creation cycles and uneven quality. Recently, significant progress has been made in natural language processing using artificial intelligence, providing new ideas and methods for automated script generation [2].

Long Short-Term Memory (LSTM), an effective recurrent neural network, excels in processing time series data. Its advantage in capturing long-distance dependencies makes it an ideal choice for generating coherent text [3]. However, LSTM still faces challenges in information decay and computational efficiency when processing long-sequence text. Currently, text generation models based on LSTM and attention mechanisms have achieved remarkable results in areas such as machine translation, automatic summarization, and dialogue systems [4]. In the field of script generation, relevant research is still in its infancy. Existing studies mainly focus on short text generation or dialogue generation, and there are still many challenges in generating complex plot developments and multi-character dialogue scripts. Furthermore, existing models often encounter issues such as repetition and logical inconsistencies when generating long scripts, limiting their effectiveness in practical applications [5].

The development of the industry and practical needs have propelled the continuous exploration of intelligent script generation algorithms. The current research focus is on enhancing the creativity and linguistic naturalness of generated text while ensuring its coherence [6]. Additionally, achieving high-quality multilingual generation

[1] College of Arts and Media, Jinzhong College of Information，Taigu ShanXi 030800, China

Corresponding Author: Guo Nili

Email：gnl110357778@163.com

Password：Poi741852@

for scripts in different languages is also an urgent issue [7]. To address these challenges, this study proposes an efficient script generation algorithm combining LSTM and attention mechanisms. By introducing Particle Swarm Optimization (PSO) to optimize model hyperparameters, we aim to improve the generation effect and model performance, providing robust support for intelligent script creation and promoting the development of the film and animation industry.

1. Script Generation Algorithm Based on LSTM and Attention Mechanism

The script generation algorithm based on Long Short-Term Memory (LSTM) networks and attention mechanisms leverages the advantages of LSTM in capturing temporal sequence dependencies and the superiority of attention mechanisms in handling long sequences to generate more coherent and creative script content [8]. Prior to model training, text data requires preprocessing. Firstly, the original text data is converted into sequential data to enable LSTM to learn the contextual relationships within the text. Assuming the original text is a string $text$ with a length of N, sequences of length T can be generated through a sliding window approach. Subsequently, an appropriate sequence length T is selected. The text is then segmented into multiple subsequences of length T. Each subsequence is encoded into numerical form for input into the model. The encoding of the subsequence can be expressed as Equation (1).

$$x_i = [vocab(text[i])], [vocab(text[i+1])], \cdots, [vocab(text[i+T-1])] \quad (1)$$

In Equation (1), $text[i:i+T]$ represents the corresponding subsequence, and $vocab$ is denotes the vocabulary. After preprocessing the text data, the LSTM model is constructed, which comprises an input layer, an LSTM layer, an attention layer, and an output layer. The input layer receives the preprocessed sequence data, the LSTM layer captures the temporal dependencies in the text, the attention layer enhances the model's focus on critical information, and the output layer generates the predicted word for the next time step. The unit structure of LSTM is illustrated in Figure 1.

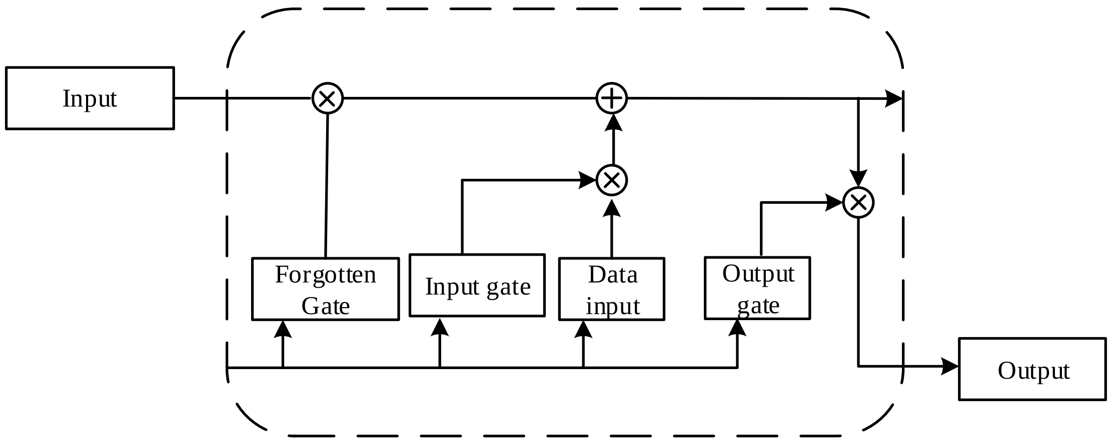

Figure 1: LSTM Unit Structure

As shown in Figure 1, at a given time step, the input gate of the LSTM is calculated as in Equation (2).

$$i_t = \sigma\left(W_i \left[x_t, h_{t-1}\right]^T + b_i\right) \quad (2)$$

In Equation (2), i_t represents the output of the input gate; σ is the activation function; W is the weight matrix for the input gate; x_t is the input sequence at time t; h is the output of the hidden layer; T is the transpose matrix; and b_i is the bias matrix for the input gate. The computation of the output value of the forget gate is shown in Equation (3).

$$f_t = \sigma\left(W_f\left[x_t, h_{t-1}\right]^T + b_f\right) \quad (3)$$

In Equation (3), f_t represents the output value of the forget gate; W_f is the weight matrix for the forget gate; and b_f is the bias matrix for the forget gate. To optimize the hyperparameters of the LSTM algorithm for script generation, a particle swarm optimization (PSO) algorithm is employed to enhance the algorithm's accuracy [9]. In the process of hyperparameter optimization, the formula for calculating the individual best position of each particle is given by Equation (4).

$$H_i^k = \left(h_{i1}, h_{i2}, ..., h_{iD}\right) \; i = 1, 2, ..., N \quad (4)$$

In Equation (4), h_{iD} represents the individual best position parameter. The global best position of the particles is calculated using Equation (5).

$$G_i^k = \left(g_1, g_2, ..., g_D\right) \quad (5)$$

In Equation (5), g_D represents the global best position parameter. Therefore, the velocity update of the particles is calculated as shown in Equation (6).

$$V_i^{k+1} = \omega V_i^k + c_1 r_1\left(H_i^k - X_i^k\right) + c_2 r_2\left(G^k - X_i^k\right) \quad (6)$$

In Equation (6), ω is the inertia factor; V_i^k is the particle's velocity before the update; c is the acceleration factor; r is a constant between 0 and 1; and k is the current iteration number. The position update of the particles is then calculated using Equation (7).

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (7)$$

In Equation (7), X_i^k represents the particle's position before the update. Therefore, in the text generation algorithm that integrates PSO and LSTM, the time series is first input. The maximum iteration count for LSTM, the size of the particle swarm, and the thresholds for particle velocity and position are then initialized. Additionally, the maximum iteration count for PSO, the number of neurons in the hidden layer, and the learning rate are also initialized. The script is generated progressively through the model's predicted outputs. To this end, the attention mechanism is integrated into LSTM as an attention layer, focusing on the parts of the input sequence that contribute most to predicting the current time step by calculating attention weights. The attention mechanism dynamically adjusts the model's focus on different positions in the input sequence at each time step by calculating the context vector c_t. Specifically, the attention scores are first calculated as shown in Equation (8).

$$e_{t,i} = v^T \tanh(W_h h_t + W_\varepsilon E_i) \quad (8)$$

In Equation (8), t represents the time step, h_t is the current hidden state, E_i is the input vector, and v, W_h, W_ε are training parameters. Subsequently, the attention weights are calculated by converting the attention scores into weights using the softmax function, as shown in Equation (9).

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{T} \exp(e_{t,j})} \quad (9)$$

After that, the context vector is computed by weighted summing the input vectors using the attention weights. The calculation is given in Equation (10).

$$c_t = \sum_{i=1}^{T} \alpha_{t,i} E_i \quad (10)$$

In Equation (10), c_t represents the context vector. The output layer calculation is then obtained by combining the context vector and the current hidden state. Once the creative model is determined, given an initial seed text, the model predicts the next word, adds it to the seed text, and continues predicting subsequent words until a complete script is generated. Therefore, after initializing the seed text, it is converted into a numerical representation and fed into the model for forward propagation to calculate the probability distribution of the next word. The next word is sampled based on the probability distribution and added to the seed text [10]. Assuming the seed text is $seed\_text$ and the script of length L is to be generated, the specific formula is given in Equation (11).

$$x_{t+1} = \arg\max(soft\max(W_c[c_t; h_t] + b)) \quad (11)$$

In Equation (11), h_t is the hidden state at time step t, W_c is the weight of the fully connected layer, b is the bias value, and c_t is the context vector calculated by the attention layer.

2. EXPERIMENTAL ANALYSIS OF THE SCRIPT GENERATION ALGORITHM BASED ON LSTM AND ATTENTION MECHANISM

2.1 Experimental Setup

The datasets used in this study originate from various publicly available script datasets and online script resources, including the Cornell Movie-Dialogs Corpus and OpenSubtitles. These datasets contain scripts of various types, including movies, TV series, animations, etc., covering multiple genres and styles. Each script in the dataset has been manually reviewed to ensure the integrity and coherence of its linguistic expression. Additionally, the datasets encompass scripts in different languages, enabling the trained model to generate scripts in multiple languages. The experimental hardware and software environment are presented in Table 1. Specifically, the experimental setup employs an Intel Xeon E5-2680 v4 2.40 GHz processor with 256 GB DDR4 memory. The GPU is a GEFORCE RTX 2080 Ti. The operating system is Ubuntu 18.04 LTS, and the deep learning framework is TensorFlow 2.4.0 with Python version 3.8.

Table 1: Software and Hardware Environment Table

Hardware and Software Environment	Configuration Instructions
Processor	Intel Xeon E5-2680 v4 2.40GHz
Memory	256GB DDR4
GPU	GEFORCE RTX 2080 Ti
Operating System	Ubuntu 18.04 LTS
Deep Learning Framework	TensorFlow 2.4.0
Python Version	Python 3.8
Other Dependent Libraries	NumPy, pandas, transformers.etc.

2.2 Performance Analysis of the Script Generation Algorithm Based on LSTM and Attention Mechanism

For the script generation algorithm (LSTM+Attention) under investigation, a comparative analysis is conducted based on BLEU scores and ROUGE scores. The results are presented in Figure 2.

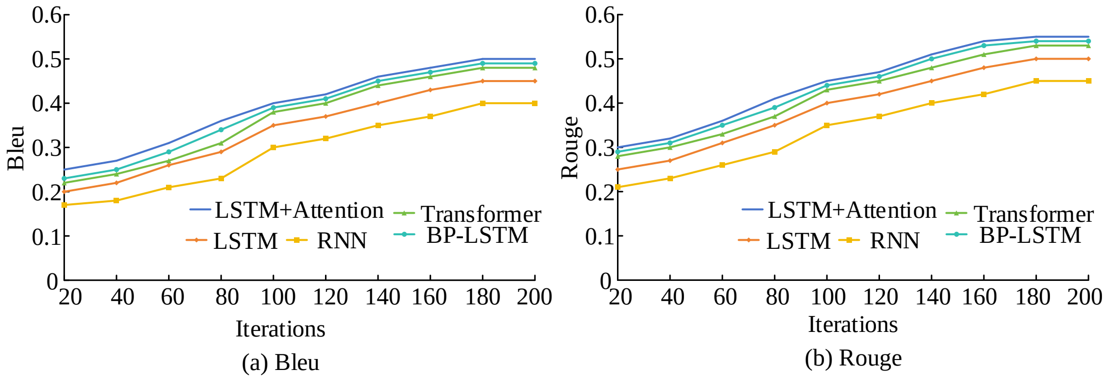

Figure 2: Comparison of BLEU and ROUGE Scores between Different Algorithms

As shown in Figure 2, during the initial 20 iterations, the BLEU score of the proposed algorithm is 0.25, which is higher than other algorithms. As the number of iterations increases, the BLEU score rises to 0.27 at 40 iterations, 0.31 at 60 iterations, and 0.36 at 80 iterations. After 100 iterations, the score reaches 0.40. The scores of LSTM and BP-LSTM remain relatively close throughout all iterations but are consistently lower than the proposed algorithm. RNN exhibits the worst performance. After 200 iterations, the BLEU score of the proposed algorithm stabilizes at 0.50, maintaining its leading position. As depicted in Figure 2(b), the proposed algorithm performs exceptionally well in terms of ROUGE scores. As the number of iterations increases, the scores steadily improve, reaching 0.32 at 40 iterations. By 100 iterations, the score of the proposed algorithm increases to 0.45, demonstrating a significant lead. After 200 iterations, the ROUGE score of the proposed algorithm stabilizes at 0.55. BP-LSTM's score remains similar to LSTM after multiple iterations, indicating limited improvement. Figure 3 shows the change curve of accuracy between the proposed algorithm and BP-LSTM on the dataset.

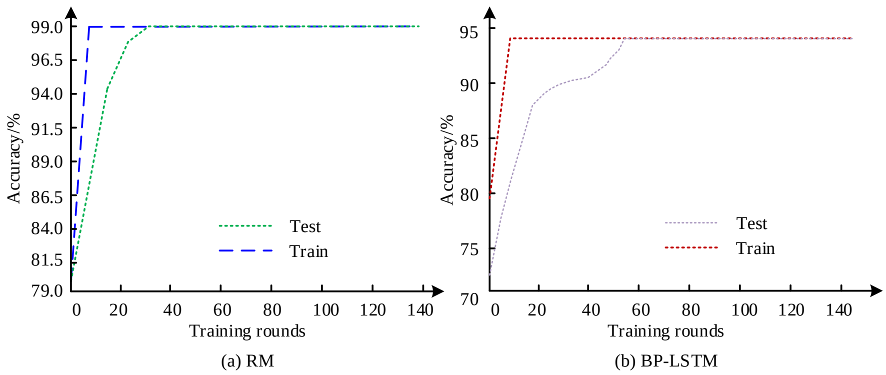

Figure 3: Accuracy Change Curves of Different Algorithms

As shown in Figure 3(a), the accuracy of the proposed model improves rapidly with a faster convergence speed, exceeding 95% when the number of iterations reaches 20. When the iteration count surpasses 40, the accuracy exceeds 98%. In contrast, Figure 3(b) reveals that while the accuracy of BP-LSTM improves significantly, reaching over 85% at 20 iterations, there are subsequent fluctuations in accuracy improvement. It stabilizes briefly at 40 iterations and reaches over 94% accuracy at 60 iterations. Upon comparing the two algorithms, it is evident that the proposed algorithm exhibits an over 40% increase in convergence speed and a more than 4% improvement in accuracy compared to BP-LSTM.

2.3 Empirical Analysis of the Script Generation Algorithm Based on LSTM and Attention Mechanism

To conduct an empirical analysis, this study examines the coherence, grammatical correctness, and content quality of the generated scripts from the perspectives of example generation and error analysis. It aims to identify the limitations and areas for improvement in the model's generation process. The generated examples are presented in Table 2. From a detailed examination, the dialogue and descriptions in the examples generated by the proposed model are more nuanced, capturing more emotional details and semantic information. For instance, the opening dialogue, "The sky is so beautiful today, perfect for a walk outside," provides a more vivid and contextualized sense compared to sentences generated by other algorithms. In terms of emotional expression, the proposed model is capable of generating deeper emotional expressions, such as "I've always loved you, no matter what happens." In contrast, the outputs of other algorithms tend to be more concise and emotionally monotonous. The sentences generated by the proposed model conform more closely to natural language expression habits, with complete sentence structures and correct grammar. For example, the description of a climax, "At that moment, he decided to stop running away and face it bravely," demonstrates complex sentence patterns and coherent logic. In terms of plot coherence, the examples generated by the proposed model exhibit stronger logical coherence between sentences, forming a complete narrative thread. For instance, the solution proposed after a conflict scene, "Let's sit down and talk to find a solution," naturally bridges the gap between the conflict and its resolution. It is apparent that the algorithm based on LSTM and attention mechanisms provides more detailed descriptions, more natural language expressions, and more coherent plot developments when generating scripts, demonstrating its significant advantages in script generation tasks.

Table 2: Examples Generated by Different Models

Scene	Examples				
	LSTM+Attention	LSTM	RNN	Transformer	BP-LSTM
Opening Dialogue	"The sky is so beautiful today, perfect for a walk outside."	"Today is beautiful."	"The sky is very blue."	"The weather is really nice today."	"The weather is nice."
Conflict Scene	"How could you do this to me? I've given so much for you!"	"Why are you like this?"	"Why are you doing this?"	"I'm upset with how you treated me."	"It's not good how you're treating me."
Resolution	"Let's sit down and talk this through, find a solution."	"Let's talk about it."	"Let's talk."	"We need to discuss a solution."	"Let's discuss how to resolve this."
Climax	"At that moment, he decided not to run away anymore and face it bravely."	"He decided not to escape anymore."	"He decided to face it."	"At that moment, he made a decision."	"He decided to bravely face it."
Ending	"Everything has finally settled, and the future is full of hope."	"It's all over."	"It's all done."	"Eventually, the issue was resolved."	"Everything has finally come to an end."
Emotional Expression	"I've always loved you, no matter what happens."	"I love you."	"I love you."	"I will love you no matter what happens."	"I have always loved you."

Table 3 presents a comparison of language error analysis among different models. The proposed model generates sentences with the fewest grammatical errors, whereas sentences generated by other algorithms often contain common grammatical mistakes. For instance, the LSTM-generated sentence "She walked to the door and took a look." contains unnecessary grammatical repetition. The proposed model generates sentences with more natural transitions in the plot, such as "He suddenly appeared in the room." In contrast, sentences generated by other algorithms exhibit jumps in plot coherence, appearing abrupt. The proposed model exhibits fewer logical errors, such as "He had breakfast in the morning and went to bed in the evening." Other algorithms often make errors in temporal and logical sequences, such as "Had breakfast in the morning and went to bed at night." The proposed model clarifies character references more precisely, for example, "She said to him, 'I like her very much.'" Other algorithms, such as RNN, are less clear in character references, generating sentences like "She said, 'I like her.'" The proposed model expresses emotions more subtly, such as "He felt an indescribable sadness." In contrast, the sentences generated by other algorithms tend to have more superficial emotional expressions, such as "He felt sad." The proposed model generates sentences with a lower repetition rate, such as "She walked to the door and took a look at the door." Other algorithms often exhibit repetition in sentence generation, such as BP-LSTM generating "She walked to the door, the door." It can be seen that the algorithm based on LSTM and attention mechanisms exhibits significant advantages in terms of grammatical correctness, plot coherence, logic, character references, emotional expression, and sentence diversity. This indicates that the algorithm can generate more natural and coherent text in script generation tasks, thereby improving the quality of the generated scripts.

Table 3: Comparative Analysis of Model Language Errors

Scenario	Language Error				
	LSTM+Attention	LSTM	RNN	Transformer	BP-LSTM
Grammatical Error	She walked to the door and had a look.	"She walked to the door and had a look at the door."	"She walked to the door and looked."	"She went to the door to have a look."	"She walked up to the door and looked."
Plot Jump	He suddenly appeared in the room.	"He was in the room."	"He was in the house."	"Suddenly, he was in the room."	"He appeared in the room."
Logical Error	"He had breakfast in the morning and went to bed in the evening."	"He had breakfast in the morning and went to bed in the evening."	"Had breakfast in the morning and went to bed in the evening."	"He ate in the morning and went to bed in the evening."	"He had breakfast in the morning and went to bed at night."
Character Role	"She said to him, 'I like her very much.'"	"She said to him, 'I like her.'"	"She said, 'I like her.'"	"She said, 'I like her.'"	"She said to him, 'I like her.'"
Emotional Expression	"He felt an indescribable sadness."	"He felt sad."	"He felt sadness."	"He felt a kind of sadness."	"He felt sad."
Repeated	"She walked to the door and had a	"She walked to the door and looked at	"She walked to the door and	"She walked to the door and	"She walked to the door

Sentence	look at the door."	the door."	looked at the entrance."	looked at it."

3.CONCLUSION

The present study aimed to investigate the script generation algorithm based on LSTM and attention mechanism to enhance the coherence, creativity, and quality of script generation. By incorporating the Long Short-Term Memory (LSTM) network and attention mechanism, the sliding window method was adopted in data preprocessing to convert textual data into sequential data. Furthermore, the Particle Swarm Optimization (PSO) algorithm was utilized to optimize the hyperparameters of the LSTM model, thus improving its accuracy and convergence speed. Experimental results demonstrated that the script generation algorithm based on LSTM and attention mechanism significantly outperforms traditional algorithms such as LSTM and RNN in terms of BLEU and ROUGE scores. After 100 iterations, the BLEU score of this algorithm reached 0.50, and the ROUGE score reached 0.55, both exhibiting a significant lead. In terms of accuracy, the research model surpassed 95% after 20 iterations and exceeded 98% after 40 iterations, demonstrating its efficiency and reliability in the task of script generation. Additionally, through empirical analysis, the algorithm exhibited significant advantages in terms of grammatical correctness, plot coherence, and other aspects, enabling it to generate more natural and coherent texts, thereby enhancing the quality of generation. However, the study also acknowledged limitations, noting that in the generation of long scripts, the model may encounter issues of repetition and redundancy, requiring further improvements in generation strategies and optimization mechanisms. In summary, the script generation algorithm based on LSTM and attention mechanism has demonstrated robust potential and superiority in script creation, providing an effective solution for empowering animation script creation with artificial intelligence. Future research will focus on optimizing the model's performance in multilingual processing and long-text generation to further enhance the quality of generation and expand its application scope.

Acknowledgements: Shanxi Provincial College Teaching Reform and innovation project "Composition Design" 'Curriculum-Practice-Competition' Trinity Curriculum Reform Action Research " (Project No.: J20221480)

REFERENCES

[1] Behera R K , Bala P K , Rana N P ,et al.Responsible natural language processing: A principlist framework for social benefits[J].Technological Forecasting and Social Change, 2023, 5(188):1-16.

[2] CM Valensise，I Grecco，D Pierangeli，C Claudio.Large-scale photonic natural language processing[J].Photonics Research, 2022(12):2846-2853.

[3] Atiwanwong T , Jantakun A , Sangsongfa A .Comparative Analysis of Optimization Value Between Artificial Neural Network and Long Short-Term Memory for Prediction Participate Matter (PM2.5) in Bangkok Thailand[J].Przeglad Elektrotechniczny, 2023, 99(7):89-95.

[4] Anh D T , Tanim A H , Kushwaha D P ,et al.Deep learning long short-term memory combined with discrete element method for porosity prediction in gravel-bed rivers[J].International Journal of Sediment Research, 2023, 38(1):128-140.

[5] R Chen，J Wang，LC Yu，X Zhang.Decoupled variational autoencoder with interactive attention for affective text generation[J].Engineering Applications of Artificial Intelligence: The International Journal of Intelligent Real-Time Automation, 2023,2(123):1-12.

[6] Galitsky B .Improving open domain content generation by text mining and alignment[J].Artificial Intelligence for Healthcare Applications and Management, 2022,2(1):489-521.

[7] Liu L , Sun Y , Liu Y ,et al.Research and Implementation of Text Generation Based on Text Augmentation and

Knowledge Understanding[J].Computational intelligence and neuroscience, 2022, 52(2022):2-3.

[8] Zhang Y , Wang S , Sun G ,et al.Aerodynamic surrogate model based on deep long short-term memory network: An application on high-lift device control:[J].Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 2022, 236(6):1081-1097.

[9] Kumari K , Singh J P , Dwivedi Y K ,et al.Multi-modal aggression identification using Convolutional Neural Network and Binary Particle Swarm Optimization[J].Future Generation Computer Systems, 2021, 118(9):187-197.

[10] Alzubi A .Learner performance prediction in the e-learning platform using the optimized deep long short-term memory classifier[J].International Journal of Wavelets, Multiresolution and Information Processing, 2022,2(20):51-80.