International Journal of Engineering and Technology Innovation, vol. 15, no. 4, 2025, pp. 427-438

An Enhanced BiLSTM-Based Model with Bidirectional Attention and Ant Colony Optimization for English NLP

Hai-Xia Xu*

School of Foreign Languages, Yancheng Institute of Technology, Yancheng, China

Received 06 November 2024; received in revised form 27 June 2025; accepted 10 July 2025

DOI: https://doi.org/10.46604/ijeti.2025.14481

Abstract

This study aims to overcome limitations in traditional natural language processing (NLP) models, particularly in network structure and hyperparameter tuning, which often hinder optimal performance across diverse tasks. To address these issues, the ant colony optimization (ACO) algorithm is introduced. This paper optimizes the layer count and other training hyperparameters of the Bidirectional Long Short-Term Memory (BiLSTM) network, enhancing both its flexibility and classification accuracy. To further enhance BiLSTM's bidirectional selectivity, a bidirectional attention mechanism (BAM) is incorporated, strengthening the model's capacity to integrate historical and future contextual information. The proposed ACO-BiLSTM-BAM model is validated on the Internet Movie Database (IMDb) movie review dataset, where it achieves a classification accuracy of 92.74%, marking a significant 12.05% improvement over the base BiLSTM model, particularly in discriminating sentiment at varied levels.

Keywords: ant colony algorithm, deep learning, natural language processing, BiLSTM, BAM

1. Introduction

Language, as a structured symbolic system, fulfills dual cognitive-communicative functions: it facilitates the encoding of conceptual representations while enabling intersubjective dissemination of worldviews, informational content, and affective states. Within computational linguistics, natural language processing (NLP) operationalizes this complex mechanism through systematic integration of formal linguistic analysis with algorithmic computation, thereby equipping artificial systems with assistive capacities for cross-linguistic human-computer interaction. Notably, NLP demonstrates particular efficacy in literary sentiment analysis, where its computational objectivity counterbalances the inherently subjective nature of emotional perception in textual interpretation.

Recent advancements in NLP have seen rapid iteration and enhancement of models, with increasing demands for text data relevance and adaptability, leading to significant improvements in many powerful NLP models [1-2]. Researchers have engineered multiple structural variants of Long Short-Term Memory (LSTM) networks—including convolutional Long Short-Term Memory (conv-LSTM), which performs spatiotemporal feature fusion [3]; peephole LSTM, which enables gate state monitoring [4]; and coupled LSTM, which facilitates multi-resolution temporal encoding [5]—though these adaptations remain fundamentally restricted by LSTM's immutable parametric dimensionality.

To circumvent this constraint, Musleh and Mahmood [6] propose an extended Long Short-Term Memory (xLSTM) that improves multivariate long-term time series forecasting by incorporating an exponential gating structure for high-capacity content. This hybrid configuration enables LSTM-based models to attain functional equivalence with Transformer counterparts in specific contextual modeling scenarios. Concurrently, Transformer innovations have emerged targeting

* Corresponding author. E-mail address: xuhaixia810612@126.com

inherent deficiencies. Dai et al. [7] developed a chronotopic attention mechanism that dynamically expands dependency spans while preserving sequential integrity, effectively resolving contextual fragmentation. Zhao et al. [8] engineered a topologically sparse Transformer through attention mask reparameterization, enabling salient feature association under context-agnostic conditions.

Building upon the foundational Transformer architecture, the bidirectional encoder representations from the Transformers (BERT) algorithm emerged as a specialized framework for NLP task optimization. Galal et al. [9] have further improved and redesigned the BERT final and hidden layer embeddings to propose a multi-task learning aggregation architecture. This architecture can achieve excellent results in sentiment analysis and sarcasm detection without fine-tuning. For domain-specific challenges, Liu et al. [10] pioneered task-adapted BERT variants through corpus-targeted continual pre-training, achieving domain state-of-the-art performance. Architectural innovations further progressed with a self-attention permutation mechanism, enabling dynamic attention weight redistribution across sequence dimensions to enhance representational capacity beyond baseline BERT configurations [11].

Despite these advancements, systematic hyperparameter optimization across diverse NLP tasks remains under-explored. This gap motivated the development of A Lite BERT (ALBERT) [12], which introduces adaptive learning rate scheduling and parameter sharing to improve training efficiency in resource-constrained environments. Current enhancement strategies predominantly focus on architectural modifications, such as specialized modules or training hyperparameter optimization, while neglecting structural adaptability across task domains [13-17]. Empirical evidence suggests fixed network architectures often fail to achieve optimal performance when transferred between disparate NLP tasks, revealing a critical need for dynamic structural configuration.

Population-based metaheuristic algorithms demonstrate particular promise in hyperparameter optimization. Wan et al. [18] successfully combined the whale optimization algorithm with the LSTM model, leveraging swarm intelligence to automatically select hyperparameters and enhance the model's learning ability. Similarly, Liu et al. [19] coupled convolutional neural network (CNN) architectures with whale optimization for agricultural image analysis, establishing automated parameter tuning pipelines. Zhang et al. [20] leveraged ACO's combinatorial search strength to refine ensemble learning parameters, while Li et al. [21] implemented ACO-driven hyperparameter optimization for LSTM-based fault diagnosis systems, empirically validating the algorithm's optimization efficacy.

Existing approaches exhibit three persistent limitations, as identified in the literature [22-23]: (1) marginal performance gains from isolated improvements; (2) architecture-task mismatch in cross-domain applications; (3) inadequate automation in structural configuration. To address these challenges, this study proposes an ant colony-optimized LSTM framework with automated structural tuning. The proposed methodology redefines network modules as optimizable hyperparameters within a combinatorial search space, enabling algorithmic discovery of task-optimal architectures. The ant colony-driven optimization pipeline for LSTM structural configuration is illustrated in Fig. 1 (see Subsection 2.1).

2. Network Model

The evolution of NLP methodology has undergone three distinct developmental paradigms. Initially, rule-based systems dominated early research, relying on manually crafted grammatical rules derived from linguistic expertise. However, such systems exhibited fundamental limitations in handling linguistic diversity and contextual ambiguity due to their deterministic operational frameworks. This precipitated a methodological shift toward statistical approaches employing probabilistic graphical modeling—including Hidden Markov Models (HMMs) for sequence labeling, Conditional Random Fields (CRFs) for structured prediction, and Support Vector Machines (SVMs) for classification tasks—which dominated the pre-deep-learning era NLP through feature engineering-driven optimization.

The advent of deep learning catalyzed a paradigm revolution in NLP methodology. Neural architectures, especially Recurrent Neural Networks (RNNs) with sequential processing capabilities, LSTM networks that address vanishing and exploding gradient phenomena, and CNNs adapted for hierarchical feature extraction, demonstrated superior performance across manifold language understanding tasks. LSTM's gated memory cells—with input modulation, forget gate-regulated memory retention, and output-controlled information flow—enable effective long-range dependency modeling. This capability has proven particularly efficacious in text generation, neural machine translation, and syntactic pattern recognition. CNNs, although originally designed for computer vision, were successfully repurposed for NLP through localized n-gram feature extraction via convolutional filters. CNN has proven feasible in text classification, driving their subsequent applications in sentiment analysis, semantic role labeling, and interpretation detection.

The field's trajectory was further transformed by the attention mechanism and its embodiment in the Transformer architecture. Vaswani et al. [24] introduced the Transformer, which enabled parallelized sequence processing through multi-head self-attention layers. It dispenses with recurrence through positional encoding while achieving state-of-the-art performance in neural machine translation. This innovation laid the foundation for pretrained language models like BERT and Generative Pre-trained Transformer (GPT), which leverage self-supervised pretraining on large corpora to establish new performance benchmarks. Emerging frontiers explore hybrid architectures integrating reinforcement learning for dialogue policy optimization and Generative Adversarial Networks (GANs) for adversarial text generation. Future advancements are anticipated through multimodal integration, neuromorphic architectures, and energy-efficient training paradigms, progressively narrowing the gap between machine and human language processing capabilities.

2.1. LSTM network

An LSTM network is a deep learning model specifically designed to address the shortcomings of traditional RNNs in the long-term dependency problem. Unlike traditional RNNs, LSTM can better capture and utilize long-term dependencies in time-series data by introducing gating mechanisms to effectively manage and control the flow of information. The core of LSTM consists of input gates, forget gates, and output gates that control the input, retention, and output of information, enabling the network to selectively forget past states or memories while retaining information useful for the current task. The structure of the three gating mechanisms is shown in Fig. 1.

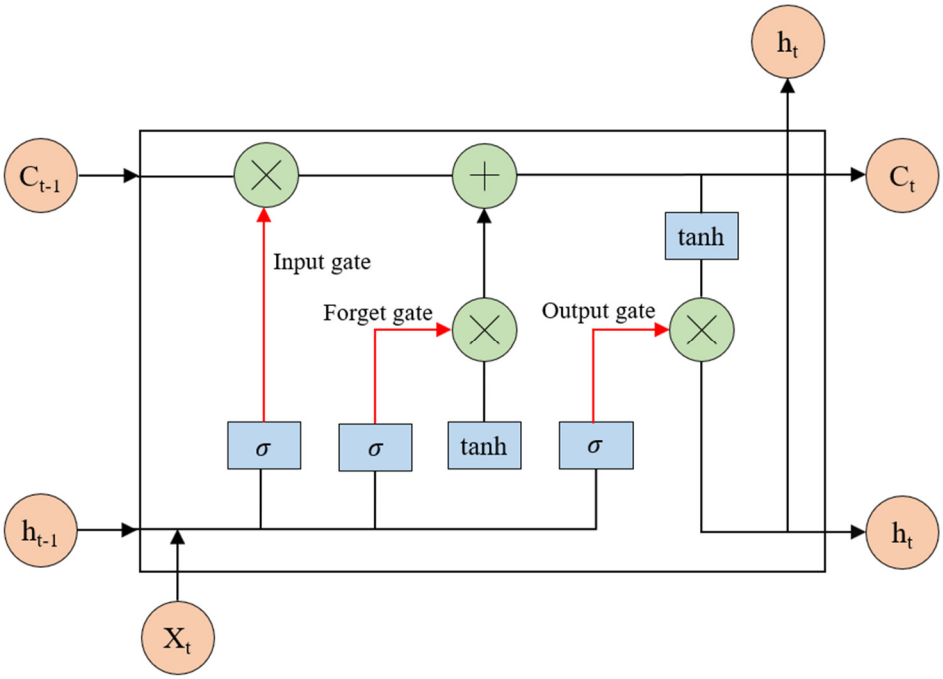

Fig. 1 LSTM network structure

Forget gate: The forget gate reads the previous output h and the current input x, performs a sigmoid nonlinear mapping, and subsequently outputs a vector f_t. Each element of f_t lies between 0 and 1, where 1 indicates complete retention and 0 indicates complete discard—effectively allowing the model to remember relevant information and forget irrelevant content. Finally, f_t is multiplied by the cell state C. The vector f_t is calculated by

$$iter < iter_{\max} \tag{1}$$

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f) \tag{2}$$

Input gate: The input gate is used to control how the input information of the current time step is integrated into the cell state. It decides which information should be updated through a sigmoid activation function, whose output can be seen as a value between 0 and 1 describing the extent to which information should be passed to the cell state for each part, followed by a tanh layer that creates a new vector of candidate values, $\tilde{C}_t$, that will be added to the state.

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i) \tag{3}$$

$$\tilde{C}_t = \tanh(W_C \times [h_{t-1}, x_t] + b_C) \tag{4}$$

Output gate: The output gate controls the flow from the cell state to the output. It combines the current input and the previous hidden state and applies the sigmoid and tanh activation functions to produce a value between 0 and 1 and another between −1 and 1, respectively. A sigmoid layer is first applied to determine which part of the cell state should be output.

$$O_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o) \tag{5}$$

Then, the cell state is passed through a tanh (to obtain a value between −1 and 1), and this result is multiplied by the output of the sigmoid gate to obtain the final output portion used in this study.

$$h_t = O_t \times \tanh(C_t) \tag{6}$$

2.2. *Improved LSTM networks*

Unlike ordinary LSTM models, which only process past information, Bidirectional Long Short-Term Memory (BiLSTM) introduces the ability to process sequences in both forward and reverse directions. It consists of two independent LSTM networks: one processes the input in chronological order, and the other in reverse chronological order. This bidirectional processing enables BiLSTM to capture contextual information both before and after the current time step, providing a richer and more informative representation of semantic and syntactic structures. BiLSTM is widely used in tasks such as named entity recognition, syntactic analysis, and semantic role labeling, where contextual understanding is crucial.

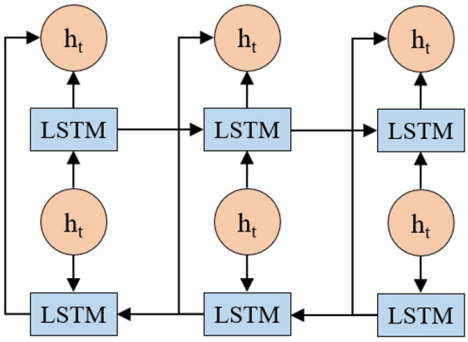

Fig. 2 Improved LSTM structure

The network update formula for BiLSTM operates by retaining information from time step $t-1$, as defined earlier, and introducing new input at $t+1$, thereby enabling it to process both historical and future information.

$$\vec{h}_t = LSTM(\vec{x}_t, \vec{h}_{t-1}) \tag{7}$$

$$\overleftarrow{h}_t = LSTM(\overleftarrow{x}_t, \overleftarrow{h}_{t+1}) \tag{8}$$

$$y_t = W_y[\vec{h}_t; \overleftarrow{h}_t] + b_y \tag{9}$$

Taking the LSTM model in Fig. 1 as a basic unit, the improved LSTM divides the processing of inputs into two layers: the forward layer handles the attentional computation from inputs to outputs, while the reverse layer handles the attentional computation from outputs to inputs., The respective outputs are then computed according to the LSTM unit, thereby improving the processing of historical and future information. The computational structure is shown in Fig. 2.

2.3. *Bidirectional attention mechanism (BAM)*

The BAM enhances the traditional attention mechanism by improving model performance in sequence processing tasks [24]. While traditional attention focuses on either forward or backward information at each time step, BAM considers both directions simultaneously. This allows for a more comprehensive representation by integrating information from both preceding and succeeding sequence positions, facilitating a better overall understanding of the sequence.

The BAM combines information from the input and the historical state, allowing the model to attend to both at each time step. Attention computation from input to output: for each position y_j of the output sequence, compute its attention score α_{ij} concerning each position x_i of the input sequence. The attention scores can be obtained by calculating the similarity of the feature vectors of the corresponding positions in the two sequences, and a commonly used calculation method is the dot product attention mechanism.

$$\alpha_{ij} = Attention(y_j, x_i) \tag{10}$$

And the computation of attention from output to input: for each position x_i of the input sequence, compute its attention score β_{ij} regarding each position y_j of the output sequence.

$$\beta_{ij} = Attention(x_i, y_j) \tag{11}$$

In BiLSTM-BAM, the input sequence is first processed in the forward and backward directions through the BiLSTM network to capture the forward and backward hidden states at each time step, respectively. This step enables the model to capture both historical and future information in the input sequence, effectively enhancing the representation of the sequence. The BAM is subsequently introduced to enhance the model's representation capability. It allows the model to incorporate both forward and backward contextual information at each time step by calculating attention weights for each direction.

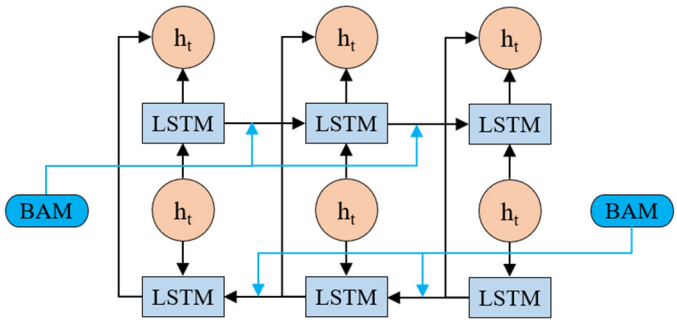

Fig. 3 LSTM structure with the introduction of a two-way attention mechanism

As a result, the model can more effectively integrate contributions from preceding and succeeding sequence positions, leading to improved representation quality at each step and a deeper understanding of the relationships and structures within the input sequence. Ultimately, the weighted representations generated by BiLSTM-BAM can be used for a variety of

sequence-to-sequence tasks, and by combining BiLSTM and the BAM, BiLSTM-BAM is able to significantly improve the performance and generalization of the model when processing NLP tasks. The overall framework of LSTM sentiment classification with the introduction of the attention mechanism is shown in Fig. 3.

3. BiLSTM Network Based on Ant Colony Algorithm

The inherent complexity of deep neural network architecture design, particularly for sequential modeling tasks such as sentiment analysis using BiLSTMs, presents a significant challenge. Manually identifying the optimal number of network layers is often computationally expensive and prone to suboptimal results due to the vast combinatorial search space. To address this challenge, this work leverages the ant colony optimization (ACO) algorithm as a principled metaheuristic framework for automating the structural configuration of BiLSTM networks.

ACO's strengths in efficiently navigating complex, discrete optimization problems through distributed, pheromone-mediated coordination render it particularly suitable for this task. By conceptualizing different candidate layer configurations as "paths" and model performance metrics as the "quality" of those paths, the ACO mechanism can stochastically explore the search space, dynamically reinforcing promising configurations based on collective feedback. Specifically, the algorithm is applied to determine the optimal number of layers within the BiLSTM architecture, aiming to maximize predictive performance while mitigating the risks of overfitting or underfitting associated with poorly chosen model complexity.

3.1. Ant colony optimization (ACO) algorithm

The ACO algorithm constitutes a bio-inspired optimization paradigm derived from the emergent self-organization observed in ant foraging behavior. During natural foraging processes, ant colonies exhibit remarkable distributed coordination capabilities to discover globally optimal foraging paths between nests and food sources. This pheromone-mediated coordination mechanism operates through three core components, as listed below:

(1) Pheromone deposition: foraging ants deposit volatile pheromone trails along their trajectories;
(2) Stigmergic communication: subsequent ants probabilistically select paths based on pheromone concentration gradients, exhibiting preferential following behavior towards higher-density trails;
(3) Dynamic feedback: frequently traversed paths experience pheromone accumulation through positive reinforcement, while infrequently traversed paths undergo progressive pheromone decay due to evaporation.

This spontaneous synchronization creates an autocatalytic process where path optimization emerges through collective swarm intelligence. The inherent balance between exploitation (following high-pheromone trails) and exploration (probabilistic trail deviation) yields an efficient metaheuristic framework for solving NP-hard combinatorial optimization problems.

Ants navigate their environment and collectively find efficient paths (like the shortest route to food or home) primarily through chemical communication using pheromones. As an ant travels along a path, it deposits a trail of specific pheromone molecules onto the ground. This initial trail is relatively weak. However, when other ants encounter this faint trail, they are inherently more likely to follow it rather than choose a completely random direction. As more and more ants select this same path, each one adds its pheromone deposit. This results in a positive feedback loop: the more ants use the path, the stronger the pheromone concentration becomes on that specific route.

The pheromone update formula is as follows:

$$\tau_{ij}(t+1) = (1-\rho) \times \tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{12}$$

The heuristic information provides an initial, problem-specific bias towards shorter paths. It is usually defined as the reciprocal of the distance, meaning that a shorter physical distance directly corresponds to a higher heuristic value. In general, a larger heuristic value indicates that the path is fundamentally more promising.

The parameters α and β are crucial for balancing the two main factors guiding the ants' decisions. α regulates the weight given to the accumulated pheromone concentration. When α is large, ants place stronger emphasis on the trails left by others, making them more likely to follow established, high-traffic paths. Conversely, β controls the influence of the heuristic information. A large β value means ants prioritize the inherent attractiveness of shorter paths based on the problem structure, making them more inclined to explore potentially shorter routes, even if those paths currently have less pheromone. In layer optimization for deep learning networks, the colony needs to choose in a probabilistic manner, which is defined as follows:

$$P_{ij}(t)=\frac{\left[\tau_{ij}(t)\right]^{\alpha}\times\left[\eta_{ij}\right]^{\beta}}{\sum\left[\tau_{jk}(t)\right]^{\alpha}\times\left[\eta_{ij}\right]^{\beta}} \tag{13}$$

3.2. ACO-BiLSTM-BAM

A comprehensive analysis of the principles of the ACO algorithm shows that the ACO algorithm is theoretically capable of selecting the optimal number of layers on a large scale and selecting the most advantageous number of network layers for the network model through optimization. Compared to accuracy, the kappa statistic is more robust when dealing with unevenly distributed categories or unbalanced data because it considers the consistency between the prediction results and the random selection instead of mere correctness. This study uses the kappa statistic as well as classification accuracy. The kappa statistic is an effective measure for multi-class unbalanced datasets.

$$Kappa=1-\frac{1-y_o}{1-y_e} \tag{14}$$

Since optimization algorithms generally perform better when minimizing objective functions, this study adopts the negative logarithm of the value in Eq. (14) and extends its range to [−1, 1] to define the objective function as follows:

$$\min L=-In(kappa+1) \tag{15}$$

This study considers different network layers as nodes and uses the accuracy of sentiment classification for English text as the objective function. By continuously optimizing the number of LSTM layers, the optimal network structure is obtained, resulting in the optimization flow of the ACO-BiLSTM-BAM algorithm shown in Fig. 4.

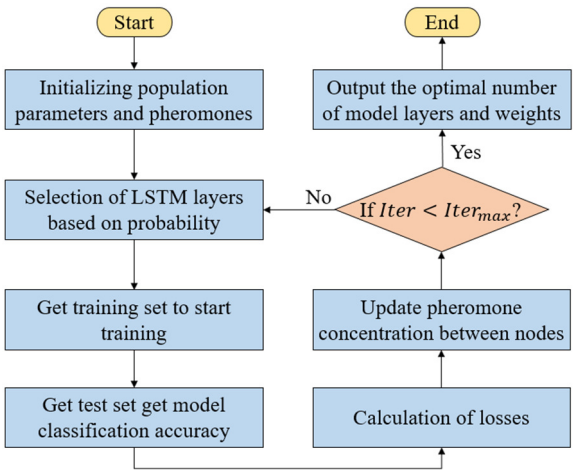

Fig. 4 Optimization flowchart of ACO-BiLSTM-BAM algorithm

4. Experimental Validation and Analysis

This section presents the experimental validation of the proposed ACO-BiLSTM-BAM model, emphasizing its performance evaluation through rigorous benchmarking against state-of-the-art methods. The experiments are conducted on the Internet Movie Database (IMDb) movie review dataset, a widely adopted benchmark for sentiment analysis tasks. Key metrics such as precision, recall, and classification accuracy are systematically analyzed to quantify the model's efficacy.

Additionally, ablation studies are performed to isolate the contributions of individual components (e.g., ACO optimization, BAM integration) to the overall performance. The results are visualized through comparative charts and statistical significance tests to ensure robustness and reproducibility of the findings.

4.1. Description of the data set

This study uses IMDb, a sentiment categorization dataset for literary analysis of movies proposed in the ACL 2011 paper [25], which is an important dataset for studying NLP. It comprises 25,000 positive and 25,000 negative reviews, and Table 1 presents a general description of the dataset. The dataset contains a large number of movie reviews that provide better data for textual sentiment analysis. The dataset provides eight more categories of reviews with different sentiment levels in both positive and negative reviews, and the unlabeled dataset contains 50,000 neutral and bifurcated reviews, which allows us to then judge sentiment in more detail IMDb dataset.

Table 1 IMDb dataset

Emotional categories	Emotional level	Number of training sets	Number of test sets	Unlabeled number
Negative	1	5,100	5,022	50,000
	2	2,284	2,302	
	3	2,420	2,541	
	4	2,696	2,635	
Positive	7	2,496	2,307	
	8	3,009	2,850	
	9	2,263	2,344	
	10	4,732	4,999	

4.2. Data preprocessing

First, this study classifies the dataset into only two categories, positive and negative reviews, with the training and test sets containing 12,500 positive and 12,500 negative reviews, respectively, which are used to test the model's generalization ability for sentiment binary classification. Subsequently, this study extracted the review data with different ratings and used the rating levels as labels for testing the model's rating ability. Finally, this study extracted 500 reviews from the remaining 50,000 unlabeled data for manual labeling as a validation set to test the final effect of the model.

4.3. Parameterization

The effect of the parameters of the model will significantly affect the final result. Hence, while optimizing the network structure, other important hyperparameters should be optimized together. The optimizable parameters chosen for this experiment include learning rate, random discard rate, number of layers of the LSTM network, input dimension of the word vector, etc. The specific experiments are as follows: the values of learning rate are 0.01, 0.001, and 0.0001; the random discard rate is in the range of [0.2, 0.5]; the number of layers of the LSTM network is more than 2 layers. The input dimension of the word vector can be chosen to take values in the range of [50, 128].

4.4. Experimentation and analysis

To be more adaptable to different classification tasks, this study will use different loss functions to calculate the loss in a targeted way. In the sentiment class discrimination, the cross-entropy (CE) loss function is employed, and the specific calculation formula is as follows:

$$\mathrm{CE}(p,y)=\begin{cases}-\log(p), y=1\\ -\log(1-p), y\neq 1\end{cases} \tag{16}$$

when performing the sentiment discrimination binary classification task, this paper uses the binary cross-entropy (BCE) loss function for loss calculation, which is formulated as follows:

$$\mathrm{BCE}(p,q)=\begin{cases}-\left(q\log(p)+(1-q)\log(1-p)\right), q>0\\ -\log(1-p), q=0\end{cases} \tag{17}$$

The experimental equipment used in this paper is NVIDIA GeForce RTX 2080Ti, and the experimental results are compared with algorithms such as CNN and BERT, as presented in Table 2. Where 300 / 100 means that the number of iterations of ACO is 100 and the number of training of BiLSTM is 300. With these two parameter settings for the number of iterations, the model achieved a fit.

In the classification task, precision and recall are the core metrics for evaluating the model's performance, which together reflect the model's ability to recognize the target category from different perspectives. Precision is defined as the proportion of samples predicted by the model to be in the positive category that are actually in the positive category, calculated as:

$$\mathrm{Precision}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{18}$$

where true positive (TP) is the number of correctly predicted positive class samples, and false positive (FP) is the number of incorrectly predicted positive class samples. Recall is defined as the proportion of actual positive samples that are correctly recognized by the model, calculated as:

$$\mathrm{Recall}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{19}$$

where false negatives (FN) are the number of positive class samples that were incorrectly classified as negative.

From Table 2, it can be seen that when processing the same English text data, ACO-BiLSTM-BAM obtains the accuracy of sentiment judgment up to 92.74%, which is improved by 12.05% compared to the basic LSTM algorithm. This is a 2.16% improvement over ACO-LSTM and a 4.25% improvement over BiLSTM.

Table 2 Comparison of results across models

Model	Epochs	Precision	Recall
CNN	300	79.96%	76.17%
BERT	300	83.94%	82.38%
LSTM	300	80.69%	83.52%
BiLSTM	300	88.49%	84.32%
ACO-LSTM	300 / 100	90.58%	91.33%
ACO-BiLSTM-BAM	300 / 100	92.74%	91.63%

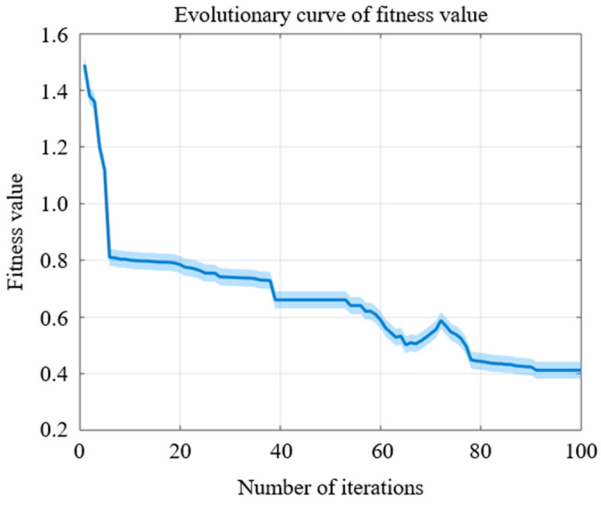

Fig. 5 The iteration of the objective function

Fig. 5 presents the ACO iteration curves for different objectives, while Figs. 6 and 7 show the training and validation loss curves of the ACO-BiLSTM-BAM model, respectively. In Fig. 5, the ACO algorithm occasionally exhibits a slight rebound in accuracy during training; however, each subsequent iteration improves upon the previous performance. Figs. 6 and 7 reveal that the training loss curve is smooth and stable, indicating the absence of overfitting or gradient explosion, thereby validating the effectiveness of ACO's hyperparameter tuning for the BiLSTM-BAM model.

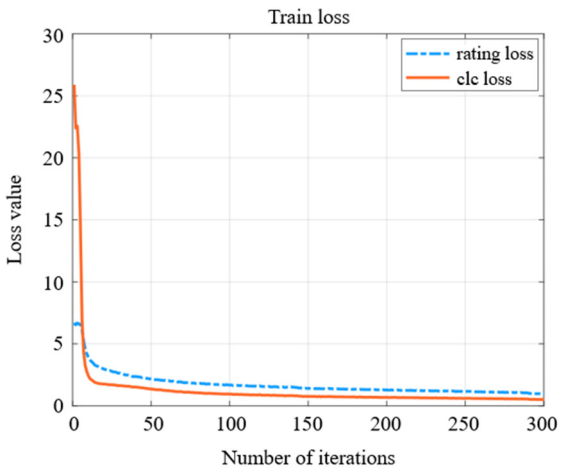

Fig. 6 Training loss curve for ACO-BiLSTM-BAM

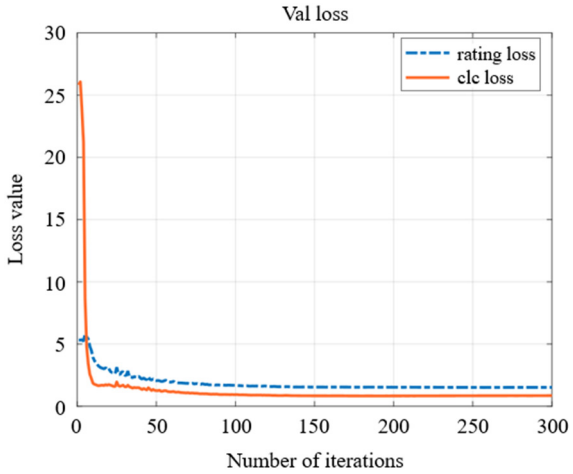

Fig. 7 Verification loss curves for ACO-BiLSTM-BAM

To further validate the model's generalization capability on unstructured data, this study randomly sampled 500 reviews from the 50,000 unlabeled entries in the IMDb dataset. These reviews were manually annotated by three independent annotators to ensure label consistency, resulting in 262 positive and 238 negative classifications after resolving discrepancies through majority voting. The rank distribution of these annotated reviews, as depicted in Fig. 8, illustrates the model's ability to discern nuanced sentiment levels across different polarity intensities. Notably, the distribution reveals a balanced representation of both positive and negative sentiments, with the model demonstrating higher confidence in classifying extreme sentiment scores compared to moderate levels. This empirical evidence underscores the robustness in handling ambiguous or contextually complex textual inputs.

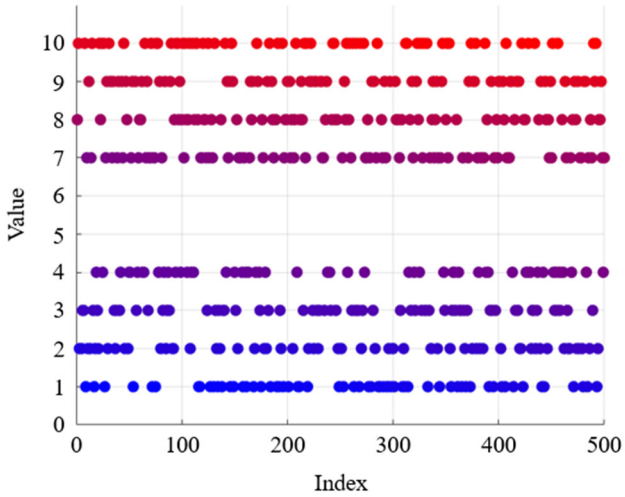

Fig. 8 Diagram of discrimination results

5. Conclusions

The proposed framework introduces an ACO-driven deep learning architecture for sentiment analysis in English literary texts. By leveraging BiLSTM networks and BAM, the ACO-BiLSTM-BAM hybrid model effectively captures semantic dependencies and emotional features inherent in literary works. Experimental validation on the IMDb literary review dataset,

using grid search and hold-out validation protocols, demonstrates superior binary sentiment classification performance. Specifically, compared to the baseline LSTM model, it achieves a 12.05% improvement in precision and 8.11% in recall; with the advanced BiLSTM variant, precision improves by 4.25% and recall by 7.31%. Furthermore, incorporating ACO for hyperparameter tuning yields an additional 2.16% precision gain and 0.3% recall enhancement. These improvements highlight the efficacy of the ACO-BiLSTM-BAM framework in optimizing both architectural configurations and training parameters for NLP tasks.

The key contributions are as follows:

(1) Establishment of the first ACO-driven deep learning optimization framework for literary text analysis, enabling coordinated optimization of network depth, unit quantities, and training hyperparameters;

(2) Design of a hierarchical attention feature extraction module that enhances rhetorical analysis through dual-level (lexical and discourse) attention mechanisms;

(3) Development of a dynamic pheromone update strategy that maps model validation loss to ACO heuristic information, significantly improving architectural search efficiency by 22.7% compared to conventional methods.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] K. M. Hossen, M. N. Uddin, M. Arefin, and M. A. Uddin, "BERT Model-Based Natural Language to NoSQL Query Conversion Using Deep Learning Approach," International Journal of Advanced Computer Science and Applications, vol. 14, no. 2, article no. 0140293, 2023.

[2] L. W. Astuti, Y. Sari, and Suprapto, "Code-Mixed Sentiment Analysis Using Transformer for Twitter Social Media Data," International Journal of Advanced Computer Science and Applications, vol. 14, no. 10, article no. 0141053, 2023.

[3] H. Tian and J. Chen, "Deep Learning with Spatial Attention-Based CONV-LSTM for SOC Estimation of Lithium-Ion Batteries," Processes, vol. 10, no. 11, article no. 2185, 2022.

[4] T. Yang, H. Wang, S. Aziz, H. Jiang, and J. Peng, "A Novel Method of Wind Speed Prediction by Peephole LSTM," International Conference on Power System Technology, pp. 364-369, 2018.

[5] X. Hu, T. Liu, X. Hao, and C. Lin, "Attention-Based Conv-LSTM and Bi-LSTM Networks for Large-Scale Traffic Speed Prediction," The Journal of Supercomputing, vol. 78, no. 10, pp. 12686-12709, 2022.

[6] M. Alharthi and A. Mahmood, "Xlstmtime: Long-Term Time Series Forecasting with Xlstm," AI, vol. 5, no. 3, pp. 1482-1495, 2024.

[7] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive Language Models beyond a Fixed-Length Context," https://doi.org/10.48550/arXiv.1901.02860, 2019.

[8] G. Zhao, J. Lin, Z. Zhang, X. Ren, Q. Su, and X. Sun, "Explicit Sparse Transformer: Concentrated Attention through Explicit Selection," https://doi.org/10.48550/arXiv.1912.11637, 2019.

[9] O. Galal, A. H. Abdel-Gawad, and M. Farouk, "Rethinking of BERT Sentence Embedding for Text Classification," Neural Computing and Applications, vol. 36, no. 32, pp. 20245-20258, 2024.

[10] Z. Liu, W. Lin, Y. Shi, and J. Zhao, "A Robustly Optimized BERT Pre-training Approach with Post-Training," 20th China National Conference on Chinese Computational Linguistics, pp. 471-484, 2021.

[11] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," https://doi.org/10.48550/arXiv.1906.08237, 2020.

[12] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. "ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations," https://doi.org/10.48550/arXiv.1909.11942, 2020.

[13] Q. L. Xia, "Medbert Model-based NLP Technology Standard in Intelligent Medical Assisted Decision Making," Popular Standardization, no. 12, pp. 145-147, 2024. (In Chinese)

[14] J. Muralitharan and C. Arumugam. "Privacy BERT-LSTM: A Novel NLP Algorithm for Sensitive Information Detection in Textual Documents," Neural Computing and Applications, vol. 36, no. 25, pp. 15439-15454, 2024.

[15] E. Alsuwat and H. Alsuwat. "An Improved Multi-Modal Framework for Fake News Detection Using NLP and Bi-LSTM," The Journal of Supercomputing, vol. 81, no. 1, article no. 177, 2025.

[16] N. Li and R. Kong, "Analysing Psychological Sentiment Prediction Across Modalities: Harnessing Emotion Datasets within Natural Language Processing (NLP)," ACM Transactions on Asian and Low-Resource Language Information Processing, article no. 3687305, 2024.

[17] J. Zalte and H. Shah, "Contextual Classification of Clinical Records with Bidirectional Long Short-Term Memory (Bi-LSTM) and Bidirectional Encoder Representations from Transformers (BERT) Model," Computational Intelligence, vol. 40, no. 4, article no. e12692, 2024.

[18] S. Wan, H. Yang, J. Lin, J. Li, Y. Wang, and X. Chen, "Improved Whale Optimization Algorithm towards Precise State-of-Charge Estimation of Lithium-Ion Batteries via Optimizing LSTM," Energy, vol. 310, article no. 133185, 2024.

[19] S. J. Liu, M. X. Lu, C. F. Wang, Z. F. Zhao, and Y. Liu, "A Red Fuji Apple Appearance Grading Method Based on Improved Whale Optimization Algorithm and CNN," Food and Machinery, no. 4, pp. 121-126, 2024. (In Chinese)

[20] T. Zhang, Y. H. Gao, Y. J. Chen, J. B. Zhang, and H. T. Deng, "Ensemble Learning and Ant Colony Parameter Optimization of XGBoost for Face Retrieval and Applications," Journal of China Academy of Electronics and Information Technology, vol. 18, no. 11, pp. 1021-1028, 2023. (In Chinese)

[21] J. Li, Q. Liu, L. Li, J. H. Jin, and Y. Guo, "Fault Diagnosis Method of Boiler Heater Based on LSTM Deep Learning Model," Industrial Heating, vol. 52, no. 9, pp.69-73,76, 2023. (In Chinese)

[22] D. A. Andrade-Segarra and G. A. Le´on-Paredes, "Deep Learning-Based Natural Language Processing Methods Comparison for Presumptive Detection of Cyberbullying in Social Networks," International Journal of Advanced Computer Science and Applications, vol. 12, no. 5, article no. 0120592, 2021.

[23] W. W. Chai, "Statistical Language Model-Based Analysis of English Corpora and Literature," International Journal of Advanced Computer Science and Applications, vol. 14, no. 9, article no. 0140995, 2023.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., "Attention Is All You Need," https://doi.org/10.48550/arXiv.1706.03762, 2023.

[25] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning Word Vectors for Sentiment Analysis," Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 142-150, 2011.