# Research on Interpretable Recommendation Algorithms Based on Deep Learning

Q. F. Wei, K. Yang

**Abstract—This paper proposes an explainable recommendation algorithm based on deep learning for developing a transparent and explainable recommendation system. The proposed algorithm combines a multi-feature fusion model for text sentiment analysis and the innovative DeepxDeepFM recommendation model to provide accurate and interpretable recommendations. First, Bi-LSTM and MCNN are employed to extract multi-dimensional features from comment data. Then, the DeepxDeepFM recommendation model is used explicitly for comment data with numerous feature vectors and significant linear changes. Finally, experimental results demonstrate that our proposed explainable recommendation algorithm increases the accuracy by 1.57% and decreases the root mean square error by 2.69%, contributing to higher model performance. Compared to other models, the improved interpretable recommendation model is smaller in size and more accurate, so it can maximize the click-through rate of e-commerce recommendation systems, which is crucial for achieving precise recommendations in the field of e-commerce.**

***Index Terms*—Deep learning, Recommendation System, Sentiment Analysis, Multi-feature Fusion, Interpretability**

## I. Introduction

The emergence of the intelligent information age has made personalized recommendations crucial for understanding user needs. Academic research on personalized recommendation [1] mainly focuses on accurately recommending items to users but often neglects the importance of explainability in recommendation systems, which leads to low feasibility and transparency of the system. In contrast, the explainable recommendation system [2] provides users with a better understanding and acceptance of recommendations. Meanwhile, the system provides relevant reasons and results to improve the user experience. Previous studies on recommendation algorithms have emphasized the development of high-precision recommendation algorithms, but there is still an ongoing effort to investigate the explainability of recommendation systems. For instance, Yu Si et al. [3] established a context-enhanced deep neural collaborative filtering model for item recommendation. This model incorporates interactive mechanisms to learn user representations and employs deep learning networks to obtain distributed representations. Ultimately, these two components are combined to make scoring predictions. Yu Shuzhen [4] et al. constructed the NAIRS model that assigns attention weights to users and provides high-quality personalized recommendations. Based on collaborative topic regression, Ye Xiao [5] et al. developed an interpretable sequential three-way recommendation system that focuses on the interpretability of the recommendations. This approach obtains multi-level explainable recommendations through three-way interactions. Ye Lu et al. [6] presented an interpretable recommendation model based on the XGBoost tree to promote the interaction between invisible items, ensure recommendation quality, and improve interpretability. Liu Hai et al. [7] developed an influence-based explainable recommendation approach, which analyzes the weight of historical influence to select users and items to explain each rating. Yu Yi et al. [8] proposed to obtain solvable recommendations in unfamiliar areas for users from a perspective of user experience. Though the accuracy of this method may not be as high as that of deep learning frameworks, it has better real-time capability and interpretability.

This paper investigates the relationship between recommendation generation reasons and score prediction tasks. To achieve accurate recommendations, the paper proposes an interpretable recommendation model based on deep learning. This model improves the explainability and personalization of the recommendation system by considering two new perspectives.

To address the issue of generating recommendation reasons, this paper proposes a multi-scale text feature fusion model that combines the Bi-LSTM model [9] and the MCNN [10] model. The proposed model aims to generate high-quality text that is both informative and personalized. It incorporates external knowledge and item recommendation reasons based on aspect emotion. Meanwhile, the user's item aspect emotion, item title, and external knowledge are fused through a two-way extended short-term memory network. Additionally, to perform aspect-level sentiment analysis and capture more detailed feature information, a fusion time series is proposed for comment text by adopting sentiment polarity analysis, thereby obtaining a high-quality text output. Secondly, considering the abundance of user and item comment text features, different from the traditional single-mode approach, in this paper, two parallel deep neural networks are used for feature vector learning in text analysis

Manuscript received June 3, 2023; revised January 9, 2024. This work was supported by the Basic Research Project of Higher Education Institutions in Liaoning Province, the Education Department of Liaoning Province (LJKZ0285), and Research on the Construction and Application of Machine Learning Improved Prediction Model.

Q.F.Wei is a postgraduate student at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan114051, China(e-mail: 942155588@qq.com).

K.Yang is a Professor at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China (corresponding author, phone: 86-15124122757; e-mail: asyangkai@126.com).

research. Also, the DeepxDeepFM model is adopted as the final recommendation model to perform rating prediction. By incorporating text comment information and deeply mining the latent features of comment texts, the model improves the accuracy of recommendations by understanding user interests and preferences. Finally, experimental results indicate that the proposed explainable recommendation model achieves a higher rating prediction accuracy and has better interpretability than similar recommendation models. By analyzing product comment information, users can understand the recommended results more intuitively, thus enhancing their acceptance of the system's explainability.

## II. Algorithm Principle

### A. Network principles related to the emotional analysis

*MCNN model*

In the traditional Convolutional Neural Network (CNN) model, as shown in Figure 1, linear filters are typically used in the convolutional layer. When extracting feature information based on the CNN convolutional layer, it is generally assumed that the features are linearly separable. However, this assumption does not hold in practical applications. Meanwhile, the computation only involves a single-layer neural network, with the local receptive field of the traditional convolutional layer. To address these issues, this paper introduces the Multi-Layer Perceptron Convolutional Network (MCNN) model, where the traditional convolutional layer is replaced with a multi-layer perceptron. The improved convolutional layer, referred to as the Mlpconv layer, is illustrated in Figure 2. This improvement enables the neurons in the local receptive field to perform relatively complex data operations and obtain more dimensional semantic features.

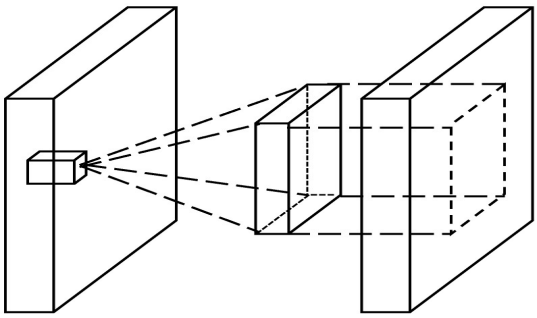


Fig. 1. CNN structure

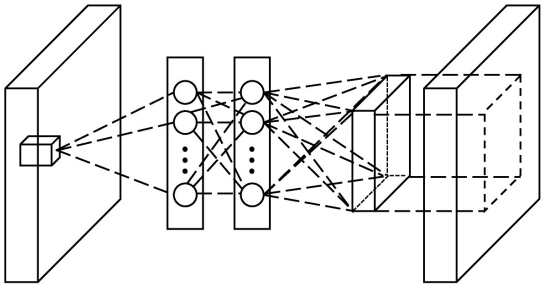


Fig. 2. Mlpconv structure

The calculation of each feature map in the Mlpconv layer is as follows (Equation 1):

$$f_{i,j,k_i}^{i} = \max\left(w_{k_i}^{i} x_{i,j} + b_{k_i}, 0\right) \quad (i = 1, 2, \cdots, n) \quad (1)$$

where, $i$ and $j$ represent the index of the word. The value of all word vectors of the text sentence, represented by the output text vector of the Bert model pre-training module in this paper, is represented as $x$. Besides, the convolution parameter of the $n$th layer is represented as $w$, and the bias value is represented as $b$. The three CNN channels have convolution kernels of different sizes, so they can capture different types of feature information in text vectors. After the convolution operation, a Mlpconv feature map F can be obtained for each text vector. $\mathrm{F} = \mathrm{f}_{1,j,k_1}^{1}, \mathrm{f}_{2,j,k_2}^{2}, \ldots, \mathrm{f}_{n,j,k_n}^{n}$, where n denotes the word sequence length of the text.

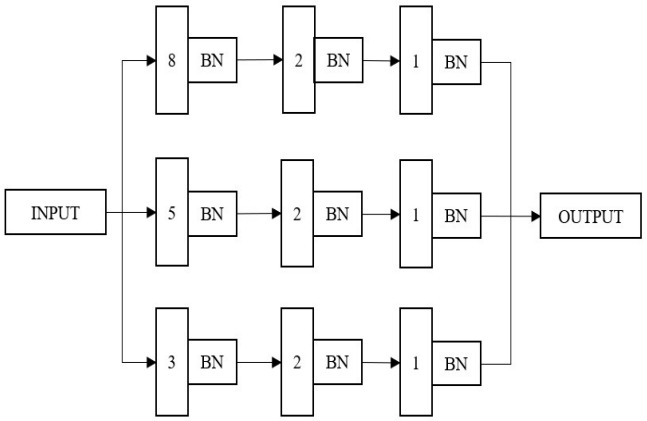



Fig. 3. MCNN architecture

As demonstrated in Figure 3, when the MCNN model takes a multivariate text sequence as input, it extracts spatial features using three different types of convolutional networks. Specifically, the first convolutional network consists of convolutional layers with kernel sizes of 8, 2, and 1, combined with batch normalization (BN) layers. The second convolutional network consists of convolutional layers with kernel sizes of 5, 2, and 1, combined with BN layers. The third convolutional network consists of convolutional layers with kernel sizes of 3, 2, and 1, combined with BN layers. Compared to one-dimensional convolutional networks, these convolutional networks have different receptive fields, so they can extract features at different scales of the temporal sequence and obtain more detailed feature information. This design enables better information extraction and ensures the integrity of the information.

*Bi LSTM model*

As a type of bidirectional recurrent neural network, the Bi-directional long-short-term memory network (Bi-LSTM) is built upon the original LSTM (Long Short-Term Memory) and consists of two LSTM layers: a forward LSTM and a backward LSTM. The forward LSTM reads the input sequence from the beginning to the end, while the backward LSTM reads the input sequence from the end to the beginning. In the last layer, the outputs of both LSTMs are concatenated. The structure of a single-layer Bi-LSTM neural network is shown in Figure 4.

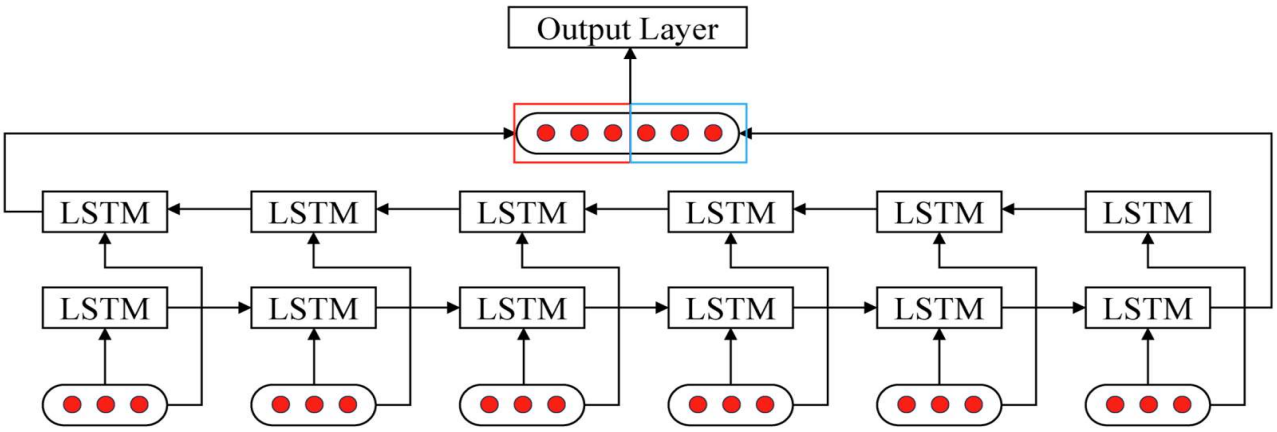



Fig. 4. The structure of the Bi-LSTM neural network

LSTM provides a solution for addressing the short-term memory problem. The model incorporates a gate mechanism that learns to retain important information in the sequence while discarding irrelevant data. This allows for capturing dependencies with a longer distance and is particularly useful for modeling time series data, including text data. However, when using LSTM to model sentences, only the preceding information is considered, while the subsequent information and the information from back to front are ignored, resulting in sentence ambiguity and deviation in polarity classification. To solve this problem, Bi-LSTM can be employed to exploit both forward and backward information in the input sequence, thereby realizing better feature extraction and classification and capturing of two-way semantic dependencies.

### *B. Principles related to recommended models*

xDeepFM [11] (Extreme Deep Factorization Machine) is a deep learning model that integrates traditional factorization machines (FM) and deep neural networks (DNN) to perform recommendations. It first models the low-order feature interactions using FM; then, it models the high-order feature interactions using a cross-network; finally, it employs DNNs to capture high-level features of the input data. The key concept is to combine the outputs of FM, cross-network, and DNNs and feed them into a fully connected layer for classification or regression. By combining the strengths of factorization machines, cross-networks, and deep neural networks, xDeepFM is a robust model for recommendation tasks, and it can effectively model both low-order and high-order feature interactions and captures high-level features of the input data.

## III. Improved Strategy

Usually, recommendation systems can be optimized by enhancing the quality of data, optimizing the algorithm, improving novelty, engaging in cross-platform collaborative recommendations, and improving real-time performance. Recommendation models often perform feature cross or feature assembly repeatedly, either explicitly [12] or implicitly. Two optimizations have been achieved in the implementation process and complexity: the multi-scale fusion model for text features and the enhanced DeepxDeepFM model.

The multi-scale text feature fusion model enhances data quality through sentiment analysis. It analyzes the sentiment tendencies expressed by users in articles, sentences, or specific product attributes and strengthens the feature extraction of comment data. This is achieved through optimizations based on specific aspect words, feature word extraction, and model fusion. On the basis of the xDeepFM model, the improved DeepxDeepFM model is developed based on the consideration that the dataset contains numerous text features that are more suitable for cross-domain feature interactions rather than memory-based expressions. It aims to ensure feature extraction accuracy while avoiding the loss of memory-based expressions. Finally, the multi-scale text feature fusion model and the improved DeepxDeepFM model are combined to establish an interpretable recommendation model, which provides not only accurate recommendations to users but also corresponding explanations.

### *A. Multi-scale text feature fusion model*

To address issues such as inaccurate classification, incomplete text feature extraction, and insufficient recognition of contextual semantics in complex text reviews [13], this paper develops a multi-scale text feature model. The model consists of two parallel modules: the Meaning Extraction Module (MCNN) and the Positional Word Tracking Module (Bi-LSTM). As depicted in Figure 5, the developed model incorporates attention-based feature fusion of word vectors, POS tags, positions, and syntactic dependency features extracted by Bi-LSTM and MCNN, to extract a thematic word library [14].

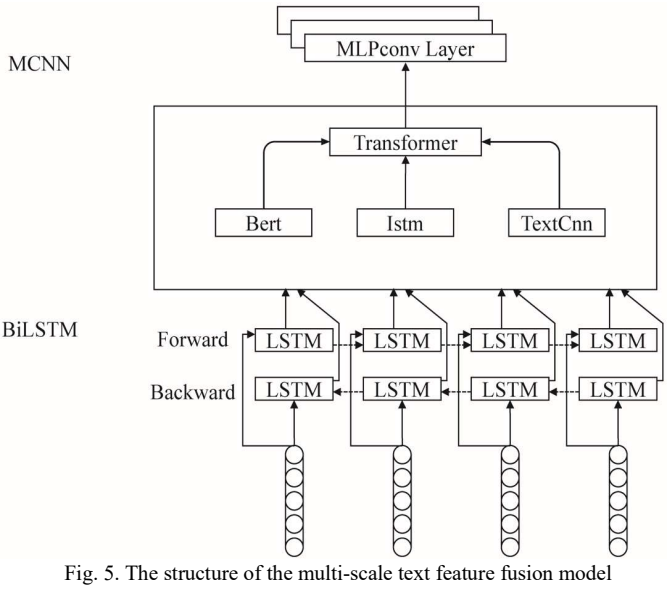



Fig. 5. The structure of the multi-scale text feature fusion model

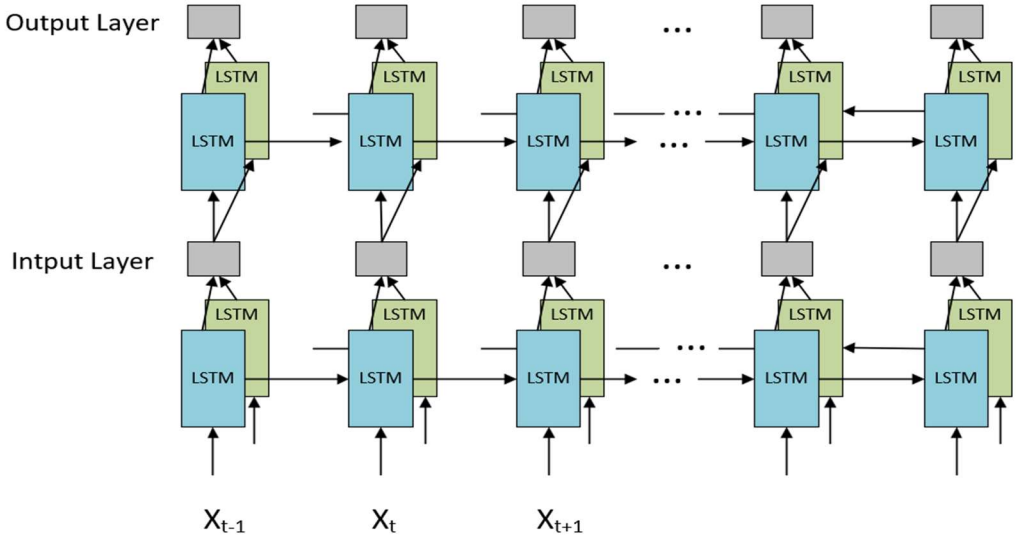



Fig. 6. The principle of the Bi-LSTM algorithm

Instead of segmenting the entire user review sentence into individual semantics that could change the sentence's meaning, the Bi-LSTM module represents it as a cohesive unit by capturing bidirectional semantic dependencies. This is illustrated in Figure 6. Conversely, the MCNN module combines the embeddings from Bert, LSTM, and TextCNN to generate textual vectors for each word in the comments. The embeddings obtained from these three models encompass various multi-dimensional features, including semantic ambiguity, repetition, and completeness.

This paper employs the Bert model to process the text and convert the original three-class sentiment text into a more detailed five-class sentiment text [15]. The three-class sentiment text consists of positive, neutral, and negative sentiments, while the five-class sentiment text consists of highly positive, optimistic, neutral, negative, and extremely negative sentiments. In the five-class text, each sentiment is assigned a corresponding score: highly positive (-5), optimistic (-4), neutral (-3), negative (-2), and highly negative (-1). In the data preprocessing stage, this fine-grained text classification helps in the subsequent sentiment analysis phase and lays the foundation for feature extraction in the MCNN module.

The multi-scale text feature fusion model aims to simultaneously capture features from both the Bi-LSTM module and the MCNN module. It extracts semantic information in the network context and combines the representations generated by these two modules using matrix operations, thereby allowing for multi-dimensional sentiment analysis of user comments. To preserve the essential features collected by the Bi-LSTM and MCNN modules, the weights of the matrices are concatenated to fuse the output of the two modules. The calculation can be expressed as follows:

$$O=(1-\beta)*Z+\beta*H \quad (2)$$

The final feature output is a feature vector O. The Bi-LSTM network assigns a weight β to each feature to indicate its importance, $\beta \in (0,1)$, Meanwhile, the SoftMax function is utilized to calculate the classification probabilities based on the feature output, as shown in equation (3) below:

$$P(y_t \mid o_t) = SoftMax(W_0 O_t + b_0) \quad (3)$$

where $P(y_t \mid o_t)$ represents the probability that the output is $y_t$ given the input feature vector $o_t$, and $o_t$ denotes the t-th vector of O.

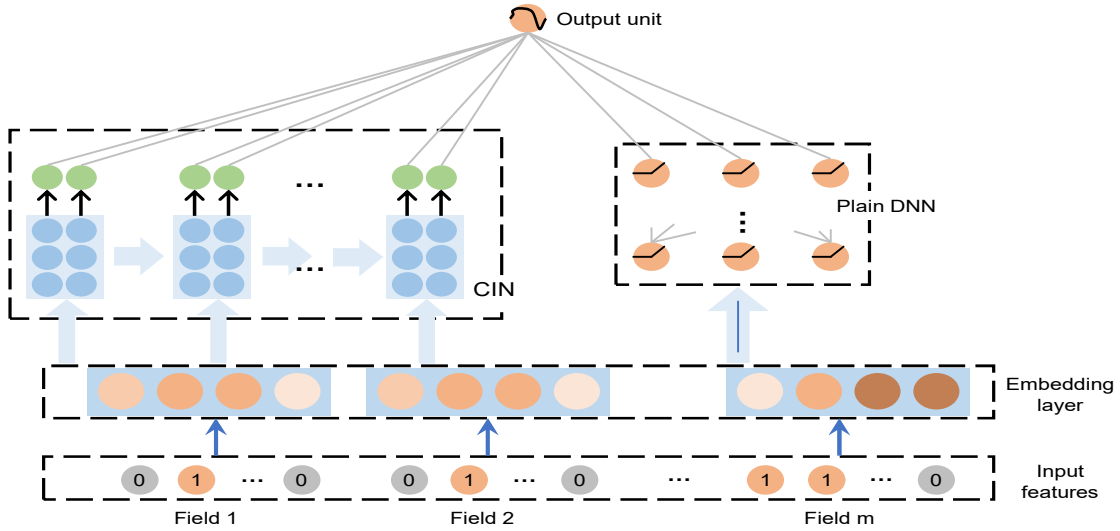



Fig. 7. The structure of the DeepxDeepFM model

### B. DeepxDeepFM model

Inspired by models such as Wide & Deep [16], DeepFM [17], and xDeepFM [18], the integration of multiple components with different structures can enhance a model's expressive power and precision. By combining the advantages of FM, cross-networks, and DNNs, the xDeepFM model can capture high-level features of input data. Based on this concept, this study presents an improved model tailored explicitly for datasets with abundant text features and relatively sparse comment data.

The original xDeepFM network has a deep structure and large model size. However, this study aims to establish a lightweight recommendation model that maintains the advantages of the xDeepFM model while achieving accurate feature extraction. The dataset used in this study contains multiple text features that are suitable for cross-interactions at the domain level instead of relying on memorization-based expressions. The improved model is constructed based on xDeepFM, and the changes are illustrated in Figure 7. The main improvements include removing the linear Logistic Regression (LR) component to prevent text features from interfering with the memory-based expressions of the model while preserving the core compressed sensing layer and deep neural network layer. Meanwhile, the DNN layer of the xDeepFM model is deepened. These improvements ensure the capture of both explicit and implicit features while preserving data integrity. The explicit features demonstrate controlled feature interactions at the vector level instead of the element level.

## IV. Analysis of Experiments And Results

### A. Dataset Selection

A dataset consisting of real reviews from Taobao is used in the experiment. The dataset comprises 30,000 pieces of scoring information, covering 24 categories of product information and related purchase details. In the dataset, there are at least 5 purchases and five comment records for each user and product, with scores ranging from 1 to 5. For this experiment, the dataset was transformed into the VOC format, and four categories of data were selected for analysis: books, songs, sports, and digital products. The selected data encompasses interactive scoring information, product reviews, and user and merchant attribute information. The sorted comment data were divided into a training set, test set, and verification set at a ratio of 8:1:1 for the experiment.

### B. Experimental Evaluation Criteria

Accuracy (ACC) is widely used in experimental recommender systems to measure the proportion of correctly predicted samples among all samples, and its calculation is shown in equation (4):

$$\mathrm{ACC}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{4}$$

Meanwhile, recall (R) and precision (P) are calculated in equations (5) and (6), respectively:

$$\mathrm{R}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{5}$$

$$\mathrm{P}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{6}$$

F1-score represents the harmonic mean of precision and recall, and its calculation is shown in equation (7):

$$F_1=\frac{2PR}{P+R} \tag{7}$$

To evaluate the model's performance, the Root Mean Square Error (RMSE) is calculated as the average squared difference between predicted and actual values of corresponding samples. RMSE measures the deviation between predicted and actual values, and a smaller RMSE indicates higher performance. In this study, the RMSE is calculated by taking the square root of the average of the sum of squared differences between the predicted ratings and the actual ratings provided by users. The calculation is shown in equation (8):

$$RMSE=\sqrt{\left(\frac{1}{n}*\Sigma(yi-\hat{y}i)^2\right)} \tag{8}$$

### C. Experimental Platform

The model was trained on a computer running Windows 10 operating system, and TensorFlow 2 was used as the deep learning framework. The computer was equipped with an Intel Core i7-11800H processor and an NVIDIA GeForce RTX 3070 graphics card with 8GB memory. The programming language was Python 3.6, and the development tool was PyCharm. To prevent overfitting, the Dropout layer was introduced to randomly disable neurons, and the Batch Normalization layer was incorporated to prevent gradient disappearance and speed up training convergence. Additionally, the optimizer used was Adam. Evaluation indicators such as loss value, accuracy, recall, F1-score, and RMSE value were compared and analyzed.

The loss function of the model is:

$$\mathrm{Loss}=\frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i-y_i\right| \tag{9}$$

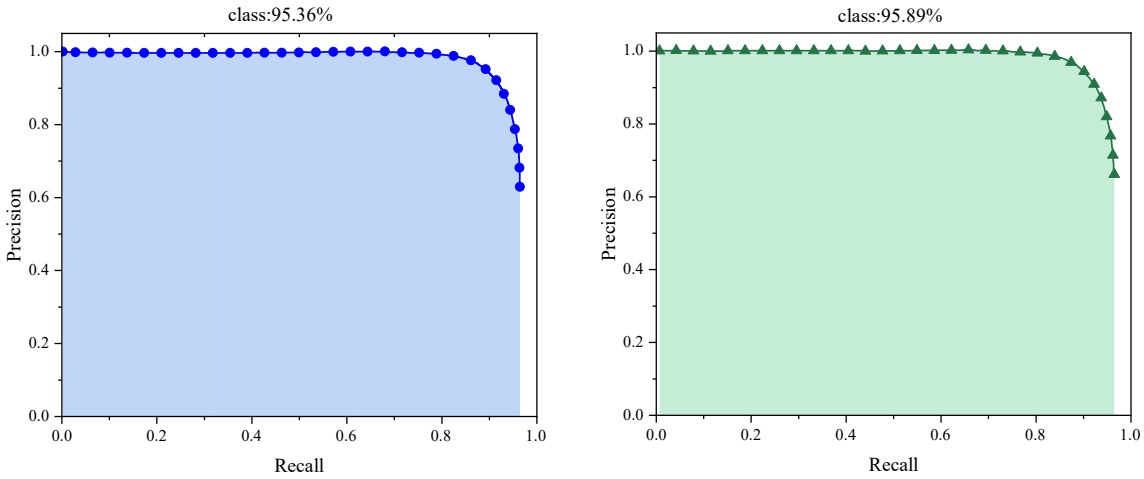
### D. Model Comparison

*Experiment on the multi-scale text feature fusion model*

In this experiment, the content and features of the data set are preprocessed, including the handling of particular words and sensitive words. Feature processing involves extending and reducing features. Then, the proposed multi-scale text feature fusion model is compared with various combination models. Two comparison experiments are carried out [19] to compare the multi-scale text feature fusion model without/with the BN layer with the original model. The specific experimental results are presented in Table I.

TABLE I
COMPARING THE EXPERIMENTAL RESULTS OF MULTIPLE COMBINATION MODELS

| Model | Precision | Recall | F1-score |
|---|---|---|---|
| CNN-LSTM Model | 80.12% | 79.15% | 79.63% |
| CNN-BiLSTM Model | 94.45% | 93.21% | 93.83% |
| MCNN-LSTM Model | 95.36% | 95.41% | 95.38% |
| Multi-scale text feature fusion model | **96.34%** | **96.31%** | **96.32%** |




a. Multi-scale text feature fusion model b. Multi-scale text feature fusion model +BN layer

Fig. 8. Comparison of P-R curves of the model

From the data in Table I, it can be seen that when our proposed model is used explicitly for information extraction, it yields superior results in both the accuracy and recall. Meanwhile, by combining multiple models, it becomes possible to extract both the positional information and the semantic meaning of words within the entire sentence, thereby enhancing the accuracy of extraction. Moreover, the integration of the MCNN-BiLSTM model ensures that the positional information and semantic meaning of the extracted words remain consistent throughout the sentence, thereby improving the accuracy of extraction and providing valuable support for subsequent explainable recommendation tasks.

For the comparative experiment regarding the inclusion of the BN layer, the results are listed in Table II. It can be seen from Table II that adding the BN layer to the algorithm increases the precision of word extraction from 96.12% to 96.34% and the recall from 96.20% to 96.31%. Figure 8 presents the P-R curve, which plots the recall on the x-axis and the precision on the y-axis, with the area under the curve representing the AP (Average Precision). A larger AP value indicates higher accuracy in comment text extraction. The addition of the BN layer leads to higher accuracy and recall in word extraction. Thus, adding the BN layer can accelerate model training, address the issues of gradient disappearance and explosion, and enable the extraction of multi-dimensional semantic information through the model.

TABLE II
IS THERE A BN LAYER COMPARISON EXPERIMENT RESULT

| Model | Precision | Recall | F1-score |
|---|---|---|---|
| Multi-scale text feature fusion model | 96.12% | 96.20% | 96.16% |
| Multi-scale text feature fusion model+BN layer | **96.34%** | **96.31%** | **96.32%** |

*DeepxDeepFM Model Experiment*

The performance of the DeepxDeepFM model is evaluated by comparing it with benchmark models such as DNN [20], AutoInt [21], Wide&Deep, DeepFM, and xDeepFM. The experimental results are presented in Table III. It is noteworthy that the data set used in this experiment is the same as that used in the previous experiment.

TABLE III
COMPARATIVE EXPERIMENTAL RESULTS OF DIFFERENT MODELS

| Model | Acc | Auc | Loss |
|---|---|---|---|
| AutoInt | 72.42% | 75.65% | 117.54% |
| Wide&Deep | 73.99% | 76.64% | 112.74% |
| DeepFM | 75.48% | 77.57% | 112.22% |
| DNN | 76.24% | 78.24% | 90.80% |
| xDeepFM | 80.64% | 84.95% | 73.07% |
| DeepxDeepFM | **82.21%** | **88.88%** | **68.11%** |

The comparison of the experimental results in Table III indicates that our optimized DeepxDeepFM model achieves significant improvements in both accuracy and recall. Meanwhile, our model obtains the lowest loss function values among various mainstream models. Besides, our improved model achieves 1.57% higher accuracy than the baseline model.

TABLE IV
COMPARATIVE EXPERIMENTAL RESULTS OF DIFFERENT MODELS

| Model | RMSE | Loss |
|---|---|---|
| AutoInt | 101.87% | 117.54% |
| Wide&Deep | 100.28% | 112.74% |
| DeepFM | 98.2% | 112.22% |
| DNN | 114.56% | 90.80% |
| xDeepFM | 96.4% | 73.07% |
| DeepxDeepFM | **93.71%** | **68.11%** |

Combining Table IV and Figure 9, the RMSE value and the loss value are compared separately. The model based on deep learning shows better prediction performance than the traditional shallow linear model. DeepFM and xDeepFM are models with both memory and generalization capabilities, so they can extract high-order feature expressions. As an extension of the Wide&Deep model, DeepFM replaces the linear Wide part with FM to strengthen the feature crossing ability of the Wide part, and thus it achieves higher ACC and AUC values. AutoInt improves the factorization machine by incorporating an attention mechanism and a multi-head self-attention mechanism, leading to higher prediction

accuracy in click-through rate. However, the multi-head attention mechanism incurs significant network losses. After the improvement, the DeepxDeepFM model reduces the RMSE value by 96.4% (from 93.71% to 68.11%) and the loss value by 73.07% compared to the baseline model xDeepFM.

TABLE V
PERFORMANCE COMPARISON EXPERIMENT OF EACH MODEL

| Model | CPU utilization | Memory usage |
|---|---|---|
| AutoInt | 74.32% | 78.29% |
| Wide&Deep | 73.21% | 77.14% |
| DeepFM | 70.12% | 71.25% |
| DNN | 71.19% | 74.37% |
| xDeepFM | 60.55% | 70% |
| DeepxDeepFM | **50.21%** | **62.41%** |

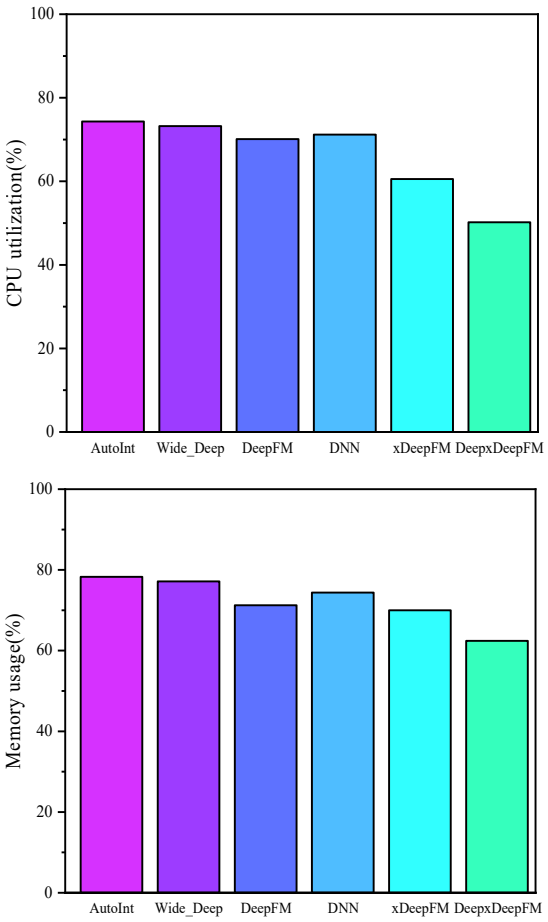



Fig. 9. Columnar comparison of CPU utilization and memory usage

Compared to traditional and similar deep learning models, our improved model performs significantly better. This paper carefully considers the ACC, AUC, and loss values and additional indicators such as CPU utilization and memory usage. Table V and Figure 9 show that our improved DeepxDeepFM model has the lowest CPU utilization and memory usage, outperforming other mainstream models. This further highlights our improved model's performance advantages, which are achieved by analyzing the correlation between features from multiple perspectives and leveraging mutual information from information theory to extract meaningful information from the original features. Meanwhile, by enhancing the bilinear feature interaction and introducing a global attention mechanism to learn second-order combined information from the features, our model has strong memory capabilities.

TABLE VI
COMPARISON OF ABLATION RESULTS

| Model | Acc | Auc | Loss |
|---|---|---|---|
| DeepxDeepFM | **82.21%** | **88.88%** | **68.77%** |
| CIN | 78.53% | 83.57% | 72.89% |
| Emotional Layer | 80.97% | 84.43% | 70.12% |
| Self-attention | 81.13% | 85.63% | 70.63% |
| Plain DNN | 80.86% | 84.24% | 70.80% |
| Linear Layer | 81.75% | 85.95% | 70.07% |

As listed in Table VI, the ablation experiment shows that after the emotion prediction mechanism is removed, the performance of the model decreases. This confirms the importance of multi-scale fusion coding based on MCNN and Bi-LSTM in capturing the features of text data. Fine-grained emotion prediction not only improves the understanding of users' emotional tendencies but also helps the model accurately capture users' attitudes towards similar products by generalizing users' emotional perception of goods, thereby optimizing the recommendation results. Secondly, when the compressed interaction network (CIN) is removed, the performance degradation is the most significant, indicating that CIN plays a core role in building user commodity interaction. By explicitly simulating the interaction between user and product feature vectors, CIN can comprehensively capture the relationship between features, thus significantly improving the model's prediction ability. Moreover, the performance of the model also decreases after the self-attention mechanism is removed. This reveals the key role of the self-attention mechanism in adaptive feature fusion, as it can infer hidden potential features based on existing information, thereby effectively reducing the interference of repeated or redundant features and enhancing attention to key features. Finally, the removal of the plain DNN and linear layer also leads to performance degradation, indicating that these components are essential for providing complementary features and enhancing the predictive ability of the recommendation system.

Overall, as illustrated in Figure 11, the AUC value gradually increases and reaches the maximum as the loss value decreases. This observation, combined with Figure 10, suggests that the DeepxDeepFM model achieves the best results in terms of accuracy and performance. Meanwhile, the model utilizes the attention mechanism [22] to filter out invalid text comment information and assign a larger weight to valuable comments [23]. During the recommendation process, users can provide explanations, and by learning the hidden semantic relationship between sentences, the model achieves higher interpretation performance. The DeepxDeepFM model achieves excellent performance by integrating a variety of efficient mechanisms, such as the emotion layer, CIN, self-attention mechanism, plain DNN,

and linear layer. This indicates that our optimized model, after performing sentiment polarity discrimination and extracting multi-dimensional aspect categories, can better comprehend the meaning of each sentence. Finally, the improved DeepxDeepFM model prevents the disruption of textual features, thereby achieving higher accuracy. Furthermore, indicated by the lowest RMSE value compared to other models, our selected model demonstrates great performance enhancements, as shown in Figure 11. This design improves the model's generalization ability and its expressive power, contributing to excellent model performance.

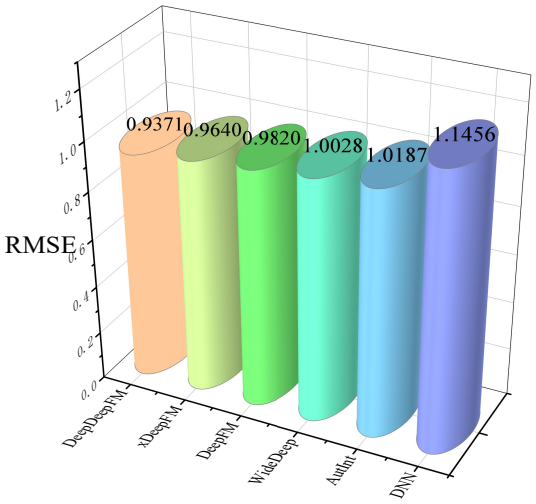



Fig. 10. Comparison of RMSE values between different models

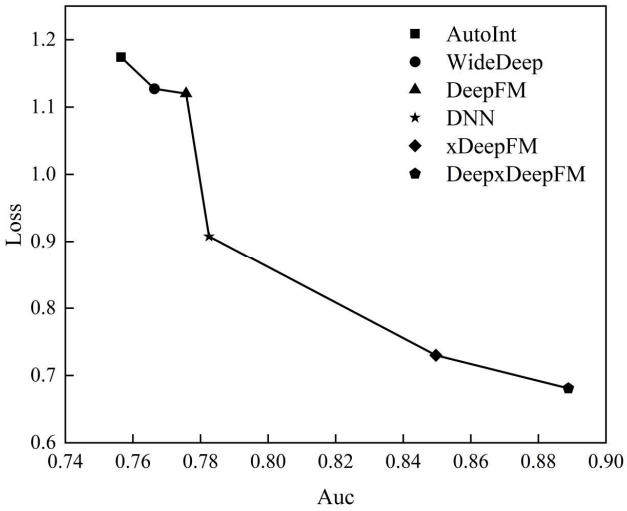



Fig. 11. Loss-Auc line comparison chart

## V. Conclusion

This paper proposes a deep learning-based interpretable recommendation algorithm, which consists of two concatenated modules: the multi-scale text feature fusion model and the DeepxDeepFM model. The main idea of the proposed algorithm is to extract features from user reviews of product information and combine multi-dimensional product and user features from auxiliary information to obtain high-order cross features, which serve as explanatory reasons. Based on this, the entire prediction process is interpretable and transparent. Meanwhile, the attention mechanism is introduced to effectively aggregate informative features and filter out irrelevant review information. Among different models, our proposed model achieves the highest accuracy, fully showing its feasibility. However, in complex semantic environments, the model may be biased in extracting semantic information as the influence of interfering factors increases. Existing research mainly improves model performance by adding neural network layers and increasing model depth. Future studies will investigate deeper neural network structures to optimize the model and improve its performance.

## References

[1] D. Yang, D. Zhang, Z. Yu, and Z. Wang, "A sentiment-enhanced personalized location recommendation system," *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, Paris, France, 2013, pp. 119-128.

[2] Z. Zeng, Y. Shi, L. F. Pieptea, and J. Ding, "Using latent features for building an interpretable recommendation system," *The Electronic Library*, vol. 39, no. 2, pp. 281-295, 2021.

[3] S. Yu, M. Yang, Q. Qu, and Y. J. E. S. W. A. Shen, "Contextual-boosted deep neural collaborative filtering model for interpretable recommendation," *Expert Systems with Applications*, vol. 136, pp. 365-375, 2019.

[4] S. Yu, Y. Wang, M. Yang, B. Li, Q. Qu, and J. J. A. Shen, "NAIRS: A Neural Attentive Interpretable Recommendation System," *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019,* Melbourne, VIC, Australia, 2019, pp. 790-793.

[5] X. Ye, and D. J. E. S. W. A. Liu, "An interpretable sequential three-way recommendation based on collaborative topic regression," *Expert Systems with Applications*, vol. 168, no. 2, pp. 114454, 2021.

[6] Y. E. Lv, Y. Yang, J. X. J. J. O. I. Zeng, and F. Systems, "An interpretable mechanism for personalized recommendation based on cross feature," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 2, pp. 1-12, 2021.

[7] H. Liu, J. Wen, L. Jing, J. Yu, and M. Zhang, "In2Rec: Influence-based Interpretable Recommendation," *Proceedings of the 28th ACM International Conference.*, Beijing, China, 2019, pp. 1803-1812.

[8] Y. Yu, R. Wei, K. Hu, Y. Bu, and X. Zhang, "Research on an Interpretable Real-Time Information Recommendation Model based on BAS-ICF algrithm," *2020 Management Science Informatization and Economic Innovation Development Conference (MSIEID).*, Guangzhou, China, 2020, pp. 304-308.

[9] V. R. Kota, and S. D. Munisamy, "High accuracy offering attention mechanisms based deep learning approach using CNN/bi-LSTM for sentiment analysis," *International Journal of Intelligent Computing and Cybernetics*, vol. 15, no. 001, pp. 61-74, 2022.

[10] Cui Z, Chen W, and Chen Y, "Multi-scale convolutional neural networks for time series classification," *arXiv preprint arXiv*:1603.06995, 2016.

[11] L. S. Li, J. Hong, S. Min, and Y. Xue, "A Novel CTR Prediction Model Based On DeepFM For Taobao Data," *2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID).*, Guangzhou, China, 2021, pp. 184-187.

[12] P. Bhuvaneshwari, A. N. Rao, and Y. H. J. W. P. C. a. I. J. Robinson, "Top-N Recommendation System Using Explicit Feedback and Outer Product Based Residual CNN," *Wireless Personal Communications*, vol. 128, no. 02, pp. 967-983, 2023.

[13] T. Jayasankarp, J. Jayalakshmip, and P. K. Rajasekaranp, "A Deep and Wide Analysis for Speech-Emotion Recognition Using Multilayer Perceptron," *International Journal of Innovative Science, Engineering & Technology*, vol. 1 no. 9, 2014.

[14] M. Al-Ghalibi, and K. Lawonn, "Topic Aspects-Based Generative Mixture Model for Movie Recommendation System using Deep Convolutional Network," *International Conference on Machine Vision(ICMV).*, Amsterdam, The Netherlands, 2019, pp. 114333K.

[15] C. Hu, and J. J. I. C. J. O. E. Cheng, "Named Entity Recognition Based on Character-level Language Models and Attention Mechanism," *International Core Journal of Engineering*, vol. 6, no. 1, pp. 196-201, 2020.

[16] Y. Ma, J. Jiang, S. Dong, C. Li, and X. Yan, "Book Recommendation Model Based on Wide and Deep Model," *IEEE International Conference on Artificial Intelligence and Industrial Design.*, Guangzhou, China, 2021, pp. 247-254.

[17] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A Factorization-Machine based Neural Network for CTR Prediction," *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, (IJCAI)*, Melbourne, Australia, 2017, pp. 1725-1731.

[18] L. Tian, C. Daofang, and X. Zhenyu, "Research on Apparel Retail Sales Forecasting Based on xDeepFM-LSTM Combined Forecasting Model," *Information*, vol. 13, no. 10, pp. 497, 2022.

[19] I. Merad, Y. Yu, E. Bacry, and S. Gaffas, "About contrastive unsupervised representation learning for classification and its convergence," *CoRR* , vol .abs/2012.01064, 2020.

[20] Narayanan D, Harlap A, Phanishayee A, et al. "PipeDream: Generalized pipeline parallelism for DNN training" *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, New York, NY, United States, 2019, pp. 1-15.

[21] Song W, Shi C, Xiao Z, et al. "Autoint: Automatic feature interaction learning via self-attentive neural networks", *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*., New York, NY, United States, 2019, pp. 1161-1170.

[22] K. Zhang, Z. Qin, Y. Liu, and X. Qin, "Multiple Attention Modules-based Knowledge Tracing," *IAENG International Journal of Computer Science*, vol. 50, no. 2, pp. 633-640, 2023.

[23] Z. Hu, W. Wang, K. Zhu, H. Zhou, and J. Chen, "Loop Closure Detection Algorithm Based on Attention Mechanism," *IAENG International Journal of Computer Science*, vol. 50, no. 2, pp. 592-598, 2023.