Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2025 SCPE.
Volume 26, Issues 5, pp. 1955–1963, DOI 10.12694/scpe.v26i5.4752

AFM AUTOINT INTELLIGENT RECOMMENDATION SYSTEM BASED ON ATTENTION MECHANISM AND AUTOMATIC INTERACTION MODELING

QIANG HAN * AND LIANG DONG †

Abstract. With the rapid growth of Internet data, intelligent recommendation systems are crucial for enhancing user experience and platform efficiency. Traditional algorithms struggle with high-dimensional sparse data and complex feature interactions. To address this, we propose the AFM-AutoInt model, integrating deep learning, attention mechanisms, and automatic feature interaction modeling. It utilizes embedding layers for dimensionality reduction, attention mechanisms for adaptive learning, and multi-layer self-attention for capturing high-order interactions. Experimental results show that AFM-AutoInt outperforms traditional methods in accuracy and robustness, making it a promising solution for next-generation recommendation systems.

Key words: Intelligent recommendation system; Deep learning; Attention mechanism; AFM AutoInt model; Data sparsity

1. Introduction. In recent years, the rapid development of the Internet and the surge in user behavior data have made intelligent recommendation systems a crucial technology for online platforms [1, 2]. These systems not only enable users to quickly discover relevant content but also enhance user satisfaction and platform revenue [3, 4]. However, as data volume grows and user preferences diversify, traditional recommendation algorithms face increasing difficulties in processing complex data structures and capturing high-order feature interactions. To overcome challenges such as data sparsity, high-dimensional feature interactions, and adaptive learning of feature weights, researchers have been actively exploring more efficient recommendation algorithms.

The existing recommendation algorithms can be mainly divided into three categories: content-based recommendation, collaborative filtering, and hybrid recommendation [5, 6]. Content based recommendation algorithms focus on analyzing the characteristics of items or users, while collaborative filtering methods rely on the similarity of user behavior [7, 8]. However, these two methods often struggle to achieve ideal results when faced with data sparsity and feature interaction complexity. Especially when dealing with high-dimensional sparse data, the interaction between features cannot be fully captured, resulting in poor recommendation performance.

With the rapid advancement of deep learning technology, new possibilities have emerged for enhancing recommendation systems. Deep learning models can effectively extract complex feature interaction patterns from vast amounts of data, thereby improving recommendation accuracy [9, 10]. However, these models also face challenges such as overfitting, limited interpretability, and prolonged training times. In particular, when handling high-dimensional sparse data, designing models that can effectively capture and leverage feature interactions has become a critical research focus in both academia and industry. To address these challenges, this paper proposes an intelligent recommendation system that integrates deep learning, attention networks, and clustering injection algorithms. The primary contribution lies in the fusion of attention mechanisms with automatic interaction modeling. The suggested method successfully gets over the drawbacks of conventional recommendation systems in learning high-order feature interactions by implementing an adaptive learning mechanism. The two main components of the AFM AutoInt model are AutoInt (Automatic Feature Interaction) and AFM (Attention Factorization Machine). By using an attention mechanism to provide weights to second-order feature interactions, the AFM module improves recommendation accuracy by allowing the model to automatically learn the importance of various feature combinations. Meanwhile, the AutoInt module further refines high-order feature interactions through a multi-layer self-attention mechanism, significantly improving the model's representation capability and overall performance [11].

*Qiongtai Normal University, Haikou, Hainan 570100, China.
†Qiongtai Normal University, Haikou, Hainan 570100, China (15143943839@163.com).

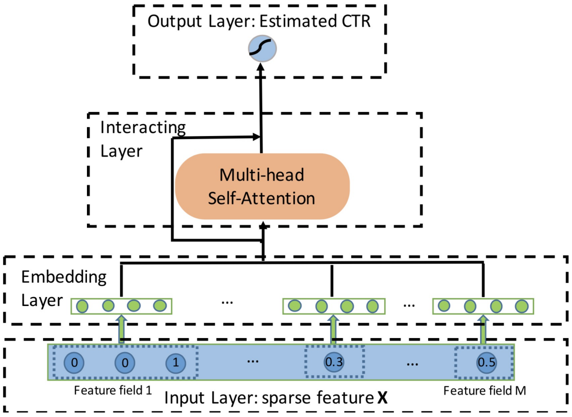

Fig. 2.1: AFM AutoInt model structure diagram.

Compared to existing recommendation algorithms, the AFM AutoInt model has significant advantages in handling data sparsity, high-dimensional feature interaction, and feature weight adaptive learning [12]. Firstly, the model proposed in this article reduces data dimensionality and improves the convergence speed and prediction accuracy of the model by embedding sparse data.Secondly, the AFM module utilizes attention mechanism to weight the second-order feature interactions, enabling the model to effectively distinguish the importance of different feature combinations for recommendation results and reduce the noise caused by invalid feature interactions [13]. Finally, the AutoInt module enhances the predictive ability of the model by introducing a multi-layer self attention mechanism to further capture high-order feature interaction information.

Although the AFM AutoInt model proposed in this article has demonstrated superior performance in experiments, there are still some challenges. For example, the complexity of the model is high, the training time is long, and further optimization may be required when dealing with large-scale data. In addition, the interpretability of the model is weak, and future research can consider combining interpretability techniques to make the recommendation results more transparent and understandable [14].

In summary, this article proposes a new recommendation algorithm by introducing deep learning and attention mechanisms, and verifies its superiority on different datasets through experiments. This study not only provides new ideas for solving the problems of data sparsity and feature interaction in recommendation systems, but also lays the foundation for further improving the performance of recommendation systems. Future research can further optimize the efficiency and interpretability of the model based on existing work, providing more effective solutions for practical applications.

2. AFM AutoInt recommendation algorithm. The AFM AutoInt model proposed in this article is shown in Fig.2.1. The overall structure of the AFM AutoInt model consists of AFM module and AutoInt module, with sparse feature data input layer, embedding layer, model layer, and output layer from bottom to top.

1. AFM (Attention-based Factorization Machine). AFM is mainly used to learn the interactions between features, while introducing an attention mechanism to give different importance to different feature interactions. Its core structure is as follows: Input layer: the input is a feature vector $x \in R^d$, where ddd is the feature dimension, and each feature is mapped to a low-dimensional space through the embedding layer. Second-order feature interaction layer: the AFM uses a factorization machine (FM) to model second-order feature interactions, calculated as follows:

$$FM(x) = \sum_{i=1}^{d} \sum_{j=i+1}^{d} (v_i \cdot v_j) x_i x_j \tag{2.1}$$

where v_i, v_j are the embedding vectors of features i and j, and, x_i, x_j are the corresponding feature values.

Attention mechanism: weights are assigned to different feature interactions, and weighted summation is computed by attention weights:

$$h = \sum_{i=1}^{d} \sum_{j=i+1}^{d} a_{ij} \left(v_i \cdot v_j\right) x_i x_j \tag{2.2}$$

where the attention weights a_{ij} are computed by an MLP network and normalized by softmax.

Output Layer: outputs the final prediction after fully connected layer and sigmoid activation function.

2. AutoInt (Automatic Feature Interaction). AutoInt mainly learns higher-order feature interactions automatically through the Self-Attention mechanism, avoiding the tedious process of manually designing feature interactions. Its network structure is as follows:

Input layer: as in the AFM, the input is the feature embedding vector matrix $X \in R^{d\times k}$, where d is the number of features and k is the embedding dimension. Multi-Head Self-Attention (MHSA): the multi-head self-attention mechanism in the Transformer structure is used to compute the importance of each feature with respect to other features:

$$Attention(\mathbf{Q},\mathbf{K},\mathbf{V}) = soft\max\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{2.3}$$

where are obtained from the input feature embedding mapping and d_k is the scaling factor.

Feature Interaction Layer: multi-layer self-attention captures higher-order interaction information and ultimately optimizes the information flow through Residual Connection and Layer Normalization.

Output layer: a fully connected layer is performed on the learned feature interaction results to compute the final predicted values.

3. Model implementation and optimization. *Loss function*: Binary Cross-Entropy (BCE) is usually used for optimization:

$$L = -\sum_{i=1}^{N} \left(y_i \log\left(\hat{y}_i\right) + \left(1 - y_i\right) \log\left(1 - \hat{y}_i\right)\right) \tag{2.4}$$

Optimization algorithms: Use the Adam optimizer with a learning rate decay strategy (e.g. Warmup + Cosine Decay) to improve convergence speed.

Regularization: Dropout and L2 regularization are used to prevent overfitting.

2.1. Sparse Feature Data Input Layer. Data sparsity is a challenge faced by recommendation systems. In this paper, the user attribute information and movie attribute information in the dataset moveielens-1M are merged based on user IDs and movie IDs to obtain N samples S, where $S = \{s_1, s_2, \cdots s_i, \cdots s_N\}$, where N is a positive integer and s_i represents the i-th user viewing record. In each user viewing record, there is only one active user, for example, s_i is the viewing record of Dirty Dancing movie by user ID 2896. s_i has m features, including both user and movie features, and one hot encoding is used to process each feature [15].

2.2. Embedding Layer. After one-hot encoding, each user and movie is represented as a unique high-dimensional sparse vector. However, such representations pose challenges, including slow convergence, excessive computational complexity, and an overwhelming number of model parameters, which makes direct input into the model impractical. To address these issues, an embedding layer is introduced to map high-dimensional sparse vectors into lower-dimensional dense representations [16]. This transformation not only reduces the model's complexity and computational burden but also preserves meaningful semantic relationships between features, thereby enhancing the model's learning ability and improving recommendation performance.

Before inputting the data into the model for training, these encoded features are divided into m different feature domains (m is a positive integer), and then matrix mapping is used to transform the features into dense and appropriately long vectors on each feature domain, thereby alleviating the problem of data sparsity. The embedding layer, in simple terms, is an initialized matrix, and its mapping process is actually a matrix multiplication. The output of the embedding layer of the jth feature domain is shown in Eq. 2.5:

$$e_j = x_j v_j \tag{2.5}$$

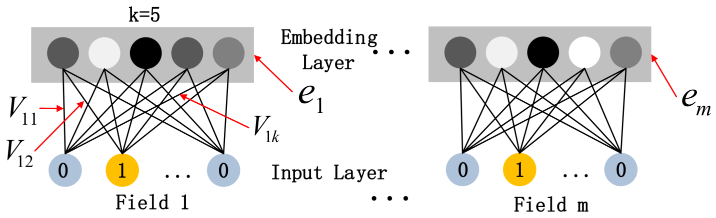

Fig. 2.2: Structure of Embedded Layer.

Among them, x_j is the vector in the jth feature field, and $j \in \{1, 2, \cdots, m\}$, v_j is the embedding matrix corresponding to feature field j. Simply put, as shown in Eq. 2.6 and Eq. 2.7, the left part of the equal sign is the one hot encoded sparse vector multiplied by the initialized embedding layer matrix. The initialized matrix is updated after each training, resulting in continuously updated results.

$$[01] \times \begin{bmatrix} 3174 \\ 2843 \end{bmatrix} = [2843] \tag{2.6}$$

$$[0100000] \times \begin{bmatrix} 7529 \\ 1748 \\ 3913 \\ 1576 \\ 2395 \\ 6473 \\ 8912 \end{bmatrix} = [1748] \tag{2.7}$$

As illustrated in Fig. 2.2, the sub-network structure from the sparse feature input layer to the embedding layer is clearly depicted. At this stage, it can be observed that while the vector representations of different input feature domains vary, the neurons in their respective embedding layers maintain a consistent dimensionality of k . This consistency arises from the fact that the embedding matrix's dimensions are determined by the product of the feature domain's dimensionality and the embedding size k . Notably, for the same feature domain, a shared embedding matrix is utilized, ensuring parameter efficiency and facilitating effective learning of feature representations. By transforming sparse high-dimensional data into dense lower-dimensional vectors, the embedding layer not only reduces computational complexity but also preserves semantic relationships, thereby enhancing the model's ability to capture intricate feature interactions.

However, features may have multiple values, so they need to be calculated separately. Multiple values are encoded separately using one hot encoding, and then input into the embedding layer. The average value of the corresponding feature embedding vector element is taken. Taking movie viewing prediction as an example, the type of a movie may have multiple values. The movie Titanic is of drama, romance, and adventure genre, and three values need to be encoded separately using one hot encoding, and then embedded. The corresponding elements of the three values are added and divided by 3, as shown in Eq. 2.8:

$$e_i = \frac{1}{q}\left(x_i^{(1)} v_i + x_i^{(2)} v_i + \cdots x_i^{(q)} v_i\right) \tag{2.8}$$

Among them, q is the number of median values in the multi valued feature domain, $q \in \{1, 2, \cdots\}$.

2.3. Model Layer.

2.3.1. AFM Model. The FM model improves the performance of recommendation models by combining individual features and introducing cross term features through pairwise combinations of features. This article improves its formula as shown in Eq. 2.9:

$$\hat{y}_{FM} = w_0 + \sum_{i=1}^{m} w_i^T e_i + p^T \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} e_i \odot e_j \tag{2.9}$$

Among them, w_0 is a global offset, w_i^T is the weight vector of different feature domains i, m represents the number of feature domains, $\odot$ is the Hadamard product, representing the multiplication of vector corresponding positions, p is the parameter vector, used to represent second-order combined features as a scalar.

The left part of Figure 2.1 is the AFM model, which is an improvement based on Eq. 2.9. Compared to other methods, FM can more effectively capture second-order feature interactions on sparse datasets. However, when the FM model outputs, the weights of second-order combined features are all 1, making it difficult to adaptively learn weights. Some second-order combined features are meaningless, which can bring unnecessary noise to the model and affect user behavior judgment.For example, girls prefer to watch romantic movies, while boys prefer sports movies. Therefore, combination features such as ”girls” and ”romance”, ”boys” and ”sports” have a positive impact on movie recommendation, while combination features such as ”girls” and ”sports”, ”boys” and ”romance” do not have a significant positive impact on movie recommendation and may also bring noise to the data. Therefore, this article introduces an attention mechanism at the output of FM second-order combination features to form a new model AFM, which enables the model to adaptively learn weights and improve its performance.

The attention mechanism model has two inputs, one is the target movie vector, and the other is the second-order combination feature output by the FM model. The purpose of using attention mechanism here is to discover which of all second-order combination features is more helpful for predicting the target movie rating.This article uses a scaled inner product attention mechanism model to solve the problem. Firstly, the correlation between the two is calculated through inner product operation. To prevent excessively long input vectors from causing the inner product to be too large, it is scaled and then normalized to obtain the weight of the second-order combined feature, which is a_{ij} in Eq. 2.10.

$$a_{ij} = softmax\left(\frac{e_i \odot e_j \cdot e_M}{\sqrt{d_k}}\right) \tag{2.10}$$

By using a soft attention mechanism to calculate the weighted distribution of second-order combined features, where each second-order combined feature is weighted according to its own weight,C_{ij} is obtained as shown in Eq. 2.11.

$$C_{ij} = a_{ij} \cdot e_i \odot e_j \tag{2.11}$$

The AFM model is shown in Eq. 2.12:

$$\hat{y}_{AFM} = w_0 + \sum_{i=1}^{m} w_i^T e_i + p^T \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} C_{ij} \tag{2.12}$$

2.4. Output Layer. The **AFM-AutoInt** model builds upon enhancements to both the **traditional AFM** and **AutoInt** models, making it highly adaptable for prediction tasks across diverse application scenarios. In this study, we evaluate its performance using the **Movielens-1M** and **Douban movie** datasets, focusing on predicting **users’ movie ratings**, a classic **regression task** in recommendation systems. To ensure accurate predictions of continuous rating values, we employ a **linear activation function** in the output layer. The model’s effectiveness is assessed through a **comparative analysis** against benchmark models, measuring key performance metrics such as **Root Mean Square Error (RMSE)** and **Mean Absolute Error (MAE)**. Additionally, we explore the impact of different hyperparameter settings, training strategies, and feature interaction mechanisms to **optimize predictive accuracy and model robustness**. Future work will extend the model’s applicability to **multi-modal data** (e.g., incorporating text and image features), explore **real-time recommendation scenarios**, and enhance model interpretability for better user trust and system transparency.

Table 3.1: Experimental Data Field Information.

Field Name	Field information
User ID	User ID, numerical code 1-6060
Gender	Gender, M/F (M is male, F is female)
Age	User age, 7 stages (1, 18, 25, 35, 45, 50, 55)
Occupation	20 professions, numbered 0-20
Movie ID	Movie ID, numerical code 0-201-3590
Genres	Movie genres, 18 types (Action, Adventure,...)
Rating	User ratings for movies, ranging from 1-5

The output of the AFM AutoInt model is shown in Eq. 2.13.

$$\hat{y} = \mu_1 \hat{y}_{AFM} + \mu_2 \hat{y}_{AutoInt} \tag{2.13}$$

Among them, μ_1 and μ_2 are the weight coefficients of AFM and AutoInt, respectively. The values of $\mu_1 + \mu_2 = 1$ and μ_2 can be used to determine the importance of low order combined features and high-order combined features.

3. Experimental results.

3.1. Experimental data. The datasets used in this experiment are Movielens-1M dataset and Douban movie dataset. The Movielens-1M dataset not only contains user attribute information and movie data information, but also over one million rating information from 6060 users for 3888 movies. The Douban movie dataset contains 29030 valid movie data, including movie data information and rating information.

3.2. Data Preprocessing. This article takes the Movielens-1M dataset as an example, merges the user attribute information and movie attribute information of the Movielens-1M dataset based on user ID and item ID. The merged data fields are shown in Table 3.1. Then, using the five fold cross validation method, the dataset is randomly divided into two categories: training set and testing set. The training set selects 80% of the data, and the testing set selects the remaining 20% of the data. Then, the features of the training set and the testing set are encoded one hot separately.

3.3. Evaluation indicators. The experiment in this article selects four evaluation indicators, namely mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), and R^2 (R-Square) coefficient of determination, to evaluate the effectiveness of predictive scoring. MSE is used to measure the degree of dispersion of a set of data itself, while RMSE is the arithmetic square root of MSE. Compared with MSE, RMSE is more sensitive to dimensionality. MAE is used to measure the degree of deviation between predicted scores and actual scores. The smaller the values of MSE, RMSE, and MAE, the smaller the model prediction error and the better the prediction result. The larger the R^2 (R-Square) coefficient of determination, the better the model prediction result.

3.4. Experimental Comparison Method. In order to demonstrate the value of this study and make the experimental results more informative and persuasive, the following five comparative algorithms were adopted in the experiment:

(1) NFM: Adding a deep neural network to the second-order feature interaction layer and introducing a feature cross pooling layer to "add up" the embedding vectors of different feature domains. High order feature interactions can be captured through the nonlinearity of the neural network, combining the modeling ability of FM for low-level feature interactions with the learning ability of DNN for high-order feature interactions.

(2) AFM: An extension of FM that uses attention mechanism to assign different weight indices to each interaction feature vector to distinguish the different importance of second-order combined features.

(3) DeepFM: combines traditional second-order factorization machines with feedforward neural networks, sharing the same sparse data input and embedding layers to extract low - and high-order features.

Table 3.2: Results of MSE and MAE under Different λ Values.

λ	MSE	MAE
0.00001	0.8020	0.7290
0.0001	0.7706	0.6966
0.001	0.7661	0.6880
0.01	0.7660	0.6863
0.1	0.7711	0.6931

Table 3.3: Comparison of evaluation indicators for each model on the test set.

Model	MSE	RMSE	MAE	R^2
NFM	0.7920	0.8902	0.7007	0.3677
AFM	0.8168	0.9035	0.7138	0.3482
Deep FM	0.7655	0.8746	0.6911	0.3888
xDeep FM	0.7672	0.8757	0.6838	0.3878
Auto Int	0.7811	0.8838	0.6977	0.3760
AFM-AutoInt	0.7603	0.8719	0.6746	0.3933

(4) xDeepFM: A new method for explicitly crossing high-order features based on vector wise pattern is proposed, which can construct finite order crossing features.

(5) AutoInt: Maps the original sparse high-dimensional feature vectors to a low dimensional space while modeling high-order feature interactions.

3.5. Results. In order to verify the effectiveness of the algorithm proposed in this article, several main parameters in the algorithm model, including regularization parameter λ, embedding layer dimension k, weight coefficientsμ_1 and $\mu_2$of AFM and AutoInt, were carefully analyzed. Multiple comparative experiments were conducted to comprehensively analyze and compare the performance of each algorithm based on four evaluation indicators: MSE, RMSE, MAE, and R^2 (R-Square) determination coefficient.

3.5.1. Regularization parameter λ. Different regularization parameters λ can also have a certain impact on the overall performance of our model. Table 3.2 lists the MSE and MAE results obtained by the AFM AutoInt model at different λ values, with an embedding layer dimension of 4.

From Table 3.2, it can be seen that as the value of λ continues to increase, the results of MSE and MAE show a continuous downward trend between λ 0.00001-0.01. At λ=0.1, the results of MSE and MAE increase again, and both MSE and MAE reach their lowest values at λ=0.01. Therefore, the model performs best when λ=0.01 is chosen.

If the regularization parameter λ value is too small, it will cause overfitting during the training process of the model. When the λ value gradually increases beyond a threshold, underfitting will occur, resulting in the loss of too many features. Therefore, when λ=0.01, it can prevent overfitting and make the model more robust during training.

3.5.2. Iteration times. After selecting the regularization parameter λ and embedding layer dimension, Fig.3.1 shows the changes in MSE and MAE of the algorithm test set with increasing iteration times.

From Fig.3.1, it can be seen that the MSE and MAE of the algorithm converge to their minimum values after 15 iterations. At this point, the model predicts that the user's rating error for the movie is minimized.

3.5.3. Comparative experimental results. Table 3.3 shows the comparative results of various algorithms.

According to Table 3.3, the AFM-AutoInt model outperforms both the AFM and AutoInt models individually, reducing prediction errors by 5.65% and 2.12%, respectively. This demonstrates that a hybrid approach combining both low- and high-order feature interactions is more effective than models that rely solely on either.

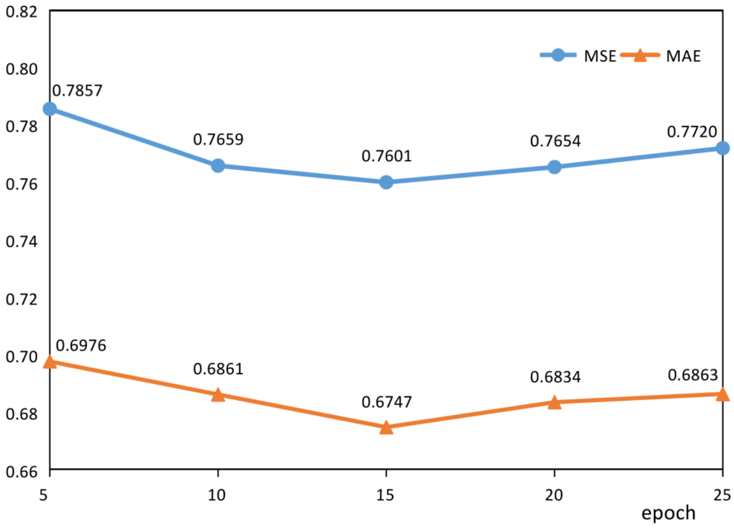

Fig. 3.1: Changes in MSE and MAE with increasing iteration times.

Table 3.4: Comparison of evaluation indicators of various models on the Douban movie dataset test set.

Model	MSE	RMSE	MAE	R^2
NFM	0.7947	0.8918	0.6945	0.3656
AFM	0.8323	0.9120	0.7086	0.3353
Deep FM	0.7908	0.8890	0.6967	0.3689
xDeep FM	0.8647	0.9300	0.7241	0.3096
Auto Int	0.7867	0.8873	0.6877	0.3715
AFM-AutoInt	0.7832	0.8847	0.6875	0.3746

However, AFM-AutoInt requires a longer training time due to its comprehensive consideration of both feature types. Extensive parameter tuning plays a crucial role in minimizing loss, which is a key factor in the model's superior performance. Additionally, AFM-AutoInt surpasses the DeepFM model by 0.52%, benefiting from the integration of attention mechanisms that refine the weighting of second-order feature combinations. Experimental results confirm that the proposed model effectively captures movie data characteristics and enhances the accuracy of user rating predictions.To fully verify the advantages of the proposed model in various evaluation indicators, this paper processed and trained the Douban movie dataset, and finally obtained a comparison of the evaluation indicators of each model on the Douban movie dataset test set, as shown in Table 3.4.

According to Table 3.4, compared to other models, the AFM AutoInt model also achieved the best prediction results on the Douban movie dataset.

4. Conclusion. This article presents a novel intelligent recommendation system model, **AFM-AutoInt**, which integrates deep learning, attention mechanisms, and automatic feature interaction modeling to address key challenges in recommendation systems, such as **data sparsity** and **complex feature interactions**. The **AutoInt module** enhances the model by leveraging a multi-layer self-attention mechanism to mine **high-order feature interactions**, significantly improving predictive accuracy. Experimental results on real-world datasets, including **Movielens-1M** and **Douban movie**, validate the superiority of AFM-AutoInt over traditional recommendation algorithms, demonstrating **lower prediction errors** and achieving state-of-the-art performance across multiple evaluation metrics. This study highlights the crucial role of incorporating both **low- and high-order feature interactions** in recommendation systems, offering a promising advancement for personalized content recommendations. Future work can explore several directions: (1) **Optimizing model complexity** to improve computational efficiency without compromising accuracy, (2) **Enhancing generalization** by adapting AFM-AutoInt to diverse datasets and real-world applications, such as e-commerce and personalized healthcare, (3) **Integrating additional contextual information**, such as temporal dynamics and user behavior patterns, to refine recommendations, and (4) **Developing explainable recommendation strategies** to enhance user trust and system transparency. By addressing these areas, AFM-AutoInt can further advance the development of next-generation intelligent recommendation systems.

Data Availability. The experimental data used to support the findings of this study are available from the corresponding author upon request.

Funding Statement.

1. 2024 Hainan Provincial Higher Education Teaching Reform Research Project, "Construction and Practical Research on the Teaching Model of Pair Programming Based on Shared Regulation" (Hnjg2024-154)
2. 2023 Hainan Provincial Higher Education Teaching Reform Research Project, "Practical Research on the Teaching of the Public Course 'Modern Educational Technology' Based on the DELC Model in a Blended Learning Environment" (Hnjg2023-139)
3. 2024 Hainan Provincial Social Science Planning Project, "Research on Strategies for Enhancing Digital Literacy of Primary and Secondary School Teachers in Rural Hainan under the Background of Educational Digital Transformation" (HNSK(ZC)24-173)

REFERENCES

[1] Deng, L. P., Guo, B., & Zheng, W. *A Service Recommendation Algorithm Based on Self-Attention Mechanism and DeepFM.* International Journal of Web Services Research (IJWSR), (2023),20(1), 1-18.

[2] Yin, P., Ji, D., Yan, H., Gan, H., & Zhang, J. *Multimodal deep collaborative filtering recommendation based on dual attention.* Neural Computing and Applications, (2023), 35(12), 8693-8706.

[3] Gao, C., Zheng, Y., Wang, W., Feng, F., He, X., & Li, Y. Causal inference in recommender systems: A survey and future directions. ACM Transactions on Information Systems, (2024), 42(4), 1-32.

[4] Dong, H., & Wang, X. Hoint: Learning explicit and implicit high-order feature interactions for click-through rate prediction. Neural Processing Letters, (2023), 55(1), 401-421.

[5] Bi, Z., Sun, S., Zhang, W., & Shan, M. *Click-through rate prediction model based on graph networks and feature squeeze-and-excitation mechanism.* International Journal of Web Information Systems, (2024), 20(4), 341-357.

[6] Gan, M., Li, D., & Zhang, X. *A disaggregated interest-extraction network for click-through rate prediction.* Multimedia Tools and Applications, (2023), 82(18), 27771-27793

[7] Jiang, Z., Li, L., & Wang, D. *MCGM: A multi-channel CTR model with hierarchical gated mechanism for precision marketing.* World Wide Web, (2023), 26(4), 2115-2141.

[8] Zheng, J., Chen, S., Cao, F., Peng, F., & Huang, M. *Explainable recommendation based on fusion representation of multi-type feature embedding.* The Journal of Supercomputing, (2024), 80(8), 10370-10393.

[9] Zhang, L., Liu, F. A., Wu, H., Zhuang, X., & Yan, Y. *CFF: combining interactive features and user interest features for click-through rate prediction.* The Journal of Supercomputing, (2024), 80(3), 3282-3309.

[10] Liliang, Z. H. O. U., Shili, Y. U. A. N., Zijian, F. E. N. G., Guilan, D. A. I., & Guofu, Z. H. O. U. *A Lambda Layer-Based Convolutional Sequence Embedding Model for Click-Through Rate Prediction.* Wuhan University Journal of Natural Sciences, (2024), 29(3), 198-208.

[11] Zhou, H., Zhang, S., Qiu, L., Wang, Z., & Hu, K. *A factorisation-based recommendation model for customised products configuration design.* International Journal of Production Research, (2023), 61(19), 6381-6402.

[12] Zheng, Q., Peng, Z., Dang, Z., Zhu, L., Liu, Z., Zhang, Z., & Zhou, J.*Deep tabular data modeling with dual-route structure-adaptive graph networks.* IEEE Transactions on Knowledge and Data Engineering, (2023), 35(9), 9715-9727.

[13] Yang, L., Zheng, W., & Xiao, Y. *Exploring different interaction among features for CTR prediction.* Soft Computing, (2022), 26(13), 6233-6243.

[14] Wu, Y. *Exploration of the Integration and Application of the Modern New Chinese Style Interior Design.* International Journal for Housing Science and Its Applications, (2024), 45(2), 28-36.

[15] Chen, P. *Research on Business English Approaches from the Perspective of Cross-Cultural Communication Competence.* International Journal for Housing Science and Its Applications, (2024), 45(2), 13-22.

[16] Wang, W.*Esg Performance on the Financing Cost of A-Share Listed Companies and an Empirical Study.* International Journal for Housing Science and Its Applications, (2024), 45(2), 1-7

Edited by: Ashish Bagwari
Special issue on: Adaptive AI-ML Technique for 6G/ Emerging Wireless Networks
Received: Aug 28, 2024
Accepted: May 27, 2025