WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 4, Number 1, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v4n1.03
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Research on Deep Learning-Based Personalized Recommendation Systems

Hewei Cao *

Lanzhou University, Lanzhou, China
*Corresponding Author: 1448040482@qq.com

ABSTRACT

Traditional methods have struggled to meet the increasingly diverse needs of users and industry development, particularly on short video platforms. This paper investigates the application of deep learning-based recommendation systems in the domain of video recommendations and their impact on user experience. It explores how different deep learning algorithms can be utilized to understand user preferences, analyze video content, and subsequently provide personalized recommendations. The study evaluates the accuracy and efficiency of these systems and investigates potential optimization strategies. The aim of this research is to provide a comprehensive perspective on how deep learning-based recommendation systems function in the short video environment and to discuss their future development directions.

KEYWORDS

Short videos; Recommendation systems; Deep learning

1. INTRODUCTION

In today's rapidly advancing Internet technology landscape, short videos have become an engaging and captivating form of content, attracting significant attention from a wide range of users. The user base and influence of short video applications have been continuously expanding. According to data, as of December 2023, the user base of short video applications has reached 1.053 billion, accounting for a significant share of the internet application market. This trend is accelerating, with a growth rate of 4.1% from 2022 to 2023, indicating the flourishing development momentum of the short video industry. At the same time, areas such as online video and instant messaging are also developing, contributing to a diverse Internet application ecosystem.

Table 1. Short Video Users and Internet Utilization Rates, 2022-2023

Application	User Base (millions) Dec 2023	Internet User Penetration Dec 2023	User Base (millions) Dec 2022	Internet User Penetration Dec 2022	Growth Rate
Short Video	105,330	96.4%	101,185	94.8%	4.1%

Despite the flourishing development of the short-form video industry, traditional recommendation methods face numerous challenges. Real-time demands necessitate quick responses to changing user interests. The long-tail effect and novelty require discovering and recommending less popular but interesting content. The filter bubble phenomenon limits users to content that aligns with their existing views, hindering new perspectives. The cold start problem makes it difficult to recommend content to new users or new content. Additionally, the high demand for computational resources poses a

challenge for systems handling large data volumes. Traditional methods struggle to meet the diverse needs of users and industry development, necessitating new approaches to improve user experience and recommendation effectiveness.

Traditional personalized recommendation systems typically use three main approaches: content-based recommendation, user-based collaborative filtering, and item-based collaborative filtering. These methods focus on extracting features of the target short video and matching them with user preferences by computing cosine similarity.

$$\textbf{simlilarity} = \cos(\theta) = \frac{A \cdot B}{||A||\,||B|} = \frac{\sum_{i=1}^{n} A_i * B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} * \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (1)$$

In the formula, A_i and B_i represent the values of vectors A and B in the i-th dimension, with n denoting the number of dimensions of the vectors. The numerator signifies the sum of the products of the corresponding values of vectors A and B in each dimension, which is the inner product of these two vectors. The denominator represents the product of the norms of vectors A and B, that is, the product of their lengths, employed for normalization.

The final result of the calculation is the inner product divided by the product of the lengths, which gives the cosine similarity. This formula is used to measure the similarity between items or users in personalized recommendation systems by extracting features of the target short video to make recommendations. After calculating similarity, the recommendation process moves into the information flow system. A short video is subjected to a series of stages, including recall, ranking, and reranking, from the moment it enters the database until it is distributed. These stages ultimately determine the most suitable display position for the video in question prior to final distribution.

To illustrate this process, consider the following example, which uses the blibli app's video recommendation:

User Request: The user refreshes the app, triggering a request.

Request Analysis: The system analyzes the request information and context.

Content Recall: The system determines the content to be recalled based on predefined rules by searching the repository.

reranking. The content is reranked according to specific strategies, resulting in the final filtered outcome.

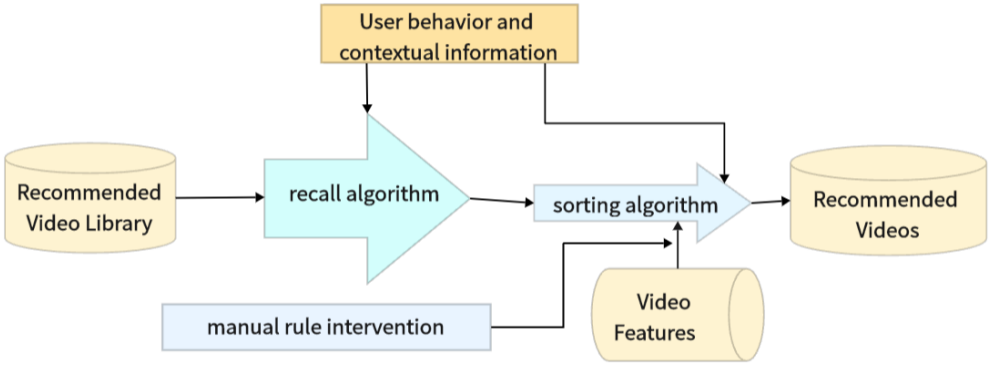

Figure 1. Two or more references

The proliferation of brief video content has rendered traditional recommendation algorithms inadequate in meeting the demands for real-time and personalized recommendations. This paper will examine emerging deep learning models, including xDeepFM and BERT, with a view to elucidating

their specific applications and performance in short video recommendation systems, with a view to addressing the limitations of traditional methods.

2. MODEL DISCUSSION

2.1. xDeepFM Model

2.1.1. The introduction of xDeepFM Model

The xDeepFM model represents an optimization of traditional recommendation models. It enhances the memory and exploration capabilities of features through the use of efficient feature crossing techniques, thereby significantly improving the accuracy and diversity of recommendations.

2.1.2. Wide & Deep Model

Prior to an examination of the xDeepFM model, it is beneficial to provide a concise overview of the fundamental principles of the Wide & Deep model. This model integrates two distinct strategies. The "wide" component emphasizes enhancing the relevance of recommendations by remembering the historical behavior of users. The "deep" component improves the model's adaptability to new situations by exploring new features. In particular, the following aspects are worthy of note:

The Wide Part This component is responsible for learning the interactions between features, thereby optimizing the model's ability to remember historical data.

The Deep Part: The deep learning architecture of this component enables the discovery of rare features, thereby enhancing the model's generalization capability.

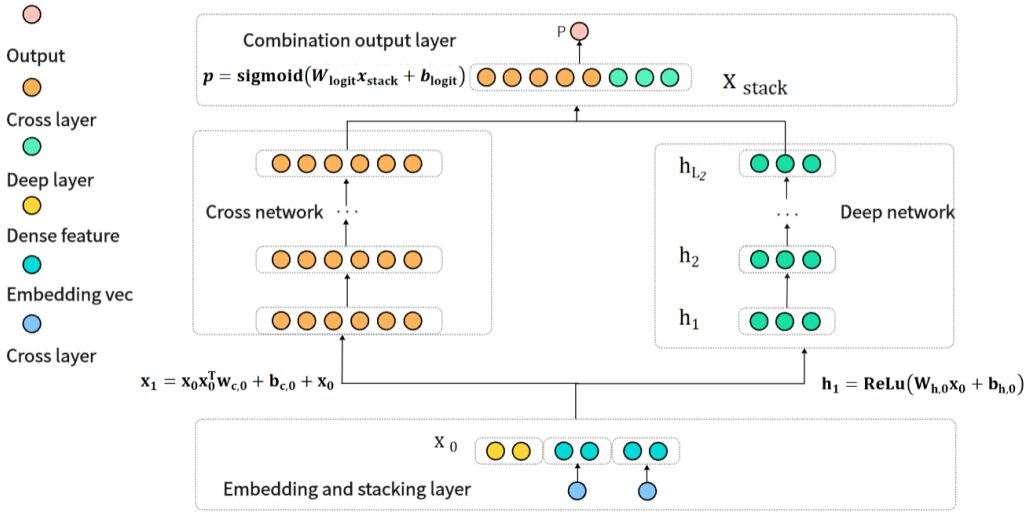

Figure 2. Wide & Deep implementation

2.1.3. The XDeepFM Model

The XDeepFM model has been further developed on the basis of the preceding discussion. The model combines deep neural networks and complex feature intersection techniques, which enables it to not only excel in feature memory but also to explore new user preference patterns through efficient feature combinations.

The XDeepFM model comprises three main parts: a linear part, which is used to process sparse features; a deep neural network (DNN) for processing the embedded dense vectors; and a Complex Interaction Network (CIN), which is designed to capture the complex interactions between different features.

2.1.4. Application of xDeepFM Model in Short Video Recommendation Systems

In the extraction of video features, the following model is typically employed. The input to this model generally comprises basic information, such as video ID and user ID, which are initially converted into one-hot encoded vectors to facilitate model processing. Furthermore, the utilization of comprehensive content data, such as the extraction of video keyframe features through algorithms like SIFT, is employed for training purposes, thereby enhancing the model's resilience in addressing data scarcity (e.g., in the context of new users or new videos).

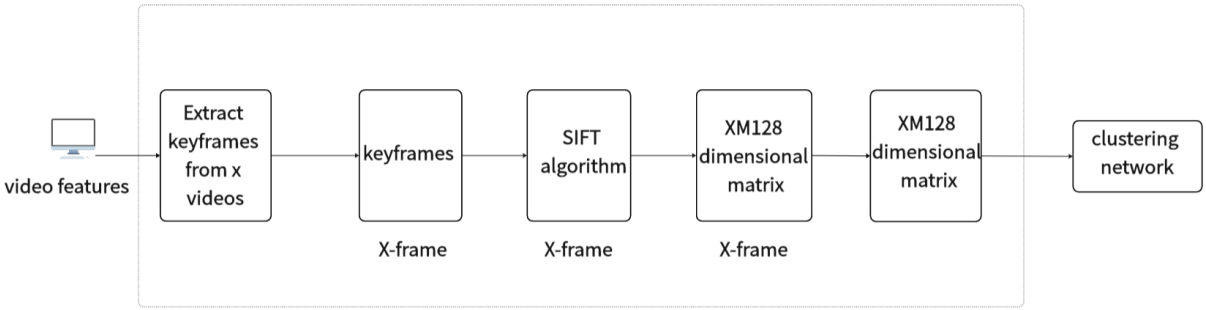

Figure 3. Video Feature Extraction

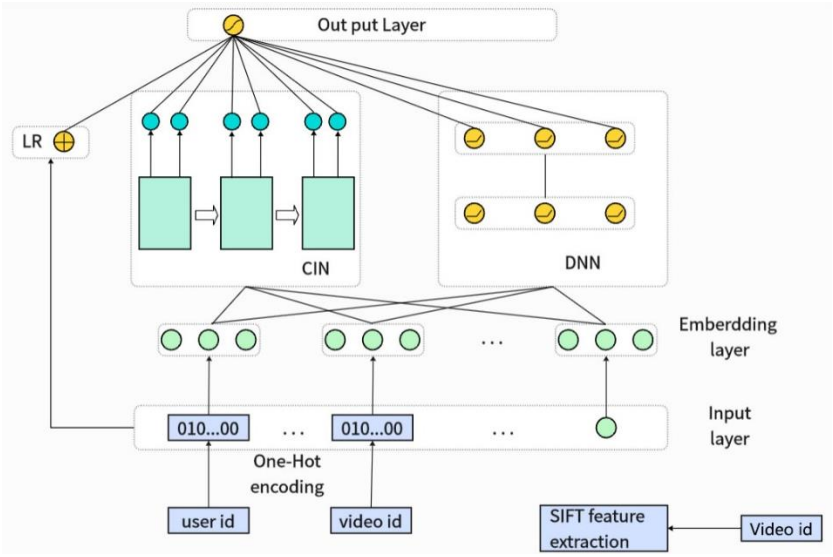

Figure 4. recommendation model

At the model input stage, fundamental data such as video ID, user ID, and video duration are transformed into one-hot vectors for machine learning training. Video features are extracted using the SIFT algorithm, forming a 128-dimensional vector that can be directly input into the embedding layer.

The Wide part recalls common patterns in user behavior and learns weights among them, requiring valuable features like user information and video features in the input layer. The CIN part automatically cross-multiplies all input features, creating combinations with shared parameters and eliminating the need for manual feature multiplication. The Deep part enhances the model's capability through deep learning, constrained by the Wide part to prevent overgeneralization. The model's final output, integrating the LR, DNN, and CIN parts, is the probability of an event occurring.

$$\hat{\mathbf{y}} = \boldsymbol{\sigma}\left(\mathbf{w}_{\mathbf{linear}}^{\mathbf{T}}\mathbf{a} + \mathbf{w}_{\mathbf{dnn}}^{\mathbf{T}}\mathbf{x}_{\mathbf{dnn}}^{\mathbf{k}} + \mathbf{w}_{\mathbf{cin}}^{\mathbf{T}}\mathbf{p}^{+} + \mathbf{b}\right) \quad (2)$$

The output result is a sigmoid function, where σ represents an input feature. The output range is between 0 and 1, indicating the probability of the event occurring.

In summary, the xDeepFM model effectively combines the directness of linear models with the deep learning capabilities of neural networks. Its innovative CIN component enables automatic feature crossing and the discovery of complex patterns, which are often challenging to achieve in traditional recommendation systems. The application of this model in recommendation systems has demonstrated superior performance, particularly in handling high-dimensional sparse data, significantly enhancing the accuracy and diversity of recommendations.

2.2. BERT Algorithm

In recommendation systems, accurately capturing users' interests is crucial. Users' interests change over time, and traditional models use historical behavior sequences to predict these interests. However, unidirectional structures have limitations, as similar item sequences with different click orders may lead to the same recommendation results. As user behavior sequences bear resemblance to text sequences and the BERT model is a robust bidirectional model, utilizing the Transformer's Encoder for pre-training allows for the full exploitation of bidirectional information.

In sequential recommendation, the following variables are defined:

$$\begin{aligned} \text{User set: } & \boldsymbol{\mathcal{U}} = \{\mathbf{u_1}, \mathbf{u_2}, \dots, \mathbf{u}_{|\mathcal{U}|}\} \\ \text{Video set: } & \boldsymbol{\mathcal{V}} = \{\mathbf{v_1}, \mathbf{v_2}, \dots, \mathbf{v}_{|\mathcal{V}|}\} \\ \text{User history sequence: } & \mathbf{S_u} = \left[\mathbf{v_1^{(u)}}, \dots, \mathbf{v_t^{(u)}}, \dots, \mathbf{v_{n_u}^{(u)}}\right] \end{aligned} \tag{3}$$

The model predicts the probability of a user clicking on video v in the subsequent time step, based on their historical sequenceS_u. $S_u$represents the user's historical sequence, while v denotes a video from the video set being predicted.

2.2.1. Embedding Layer

The model's input is primarily based on the user's historical sequence. Each video in the sequence comprises two parts in its embedding: the embedding of the video, denoted as v_i (the entire embedding matrix is denoted as E, applied to the output and input layers), and the embedding of positional information, denoted as p_i. The latter is learnable.

Furthermore, the input does not utilize all historical user behavioral data, but rather the most recent N behaviors. Given the considerable variability in the length of user behavioral sequences and the inherent dynamism of user interests, controlling the input size N can effectively reduce the model size while enhancing performance.

2.2.2. Transformer layer

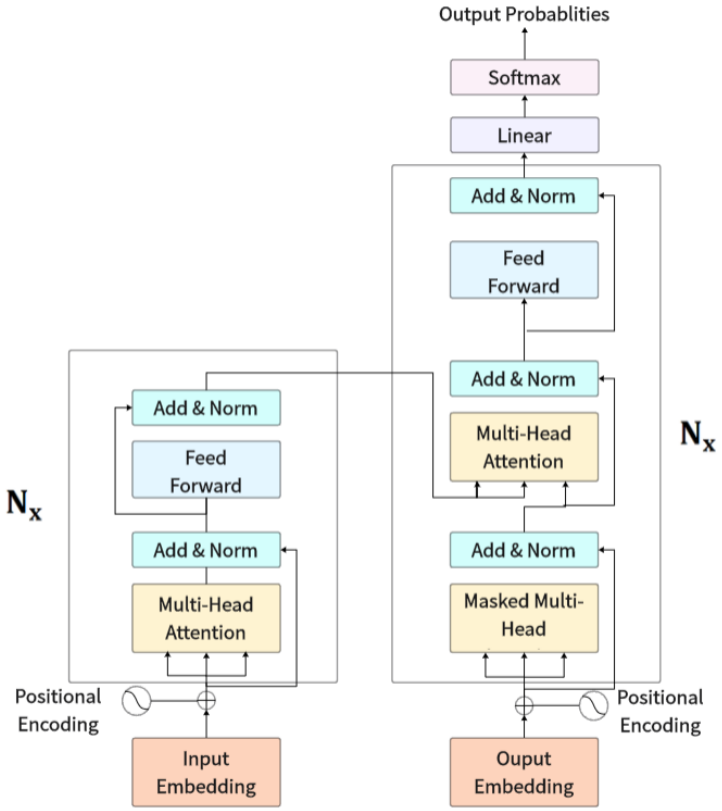

Figure 5. Transformer layer

The diagram illustrates that Multi-Head Self-Attention and Position-wise Feed-Forward constitute the two principal components of the network Transformer.

(1) Multi-Head Self-Attention:

In the context of the i-th layer of the Transformer within the model framework, the process commences with Multi-Head Self-Attention. This is followed by the Dropout, Add & Norm processes. Here, the "Add" operation represents the "Skip Connection," which aims to prevent the "vanishing gradient" problem during "backpropagation." The "Norm" operation refers to "Layer Normalization."

(2) Position-wise Feed-Forward Network:

Due to the linear mapping, in order to give the model non-linear properties, a "Position-wise Feed-Forward Network" is employed. "Position-wise" means that the vector at each position is separately input into the feed-forward neural network. The calculation method is as follows:

$$\text{A ttention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathbf{softmax}\left(\frac{\mathbf{QK}^{\top}}{\sqrt{\mathbf{d/h}}}\right)\mathbf{V} \tag{4}$$

Subsequently, the Dropout, Add, and Norm processes are executed. The Add operation represents the skip connection, which can mitigate and prevent the vanishing gradient issue that arises during backpropagation. The Norm operation pertains to layer normalization.

$$\mathbf{LN}(\mathbf{x}) = \boldsymbol{\gamma}\left(\frac{\mathbf{x}-\mathbf{u}}{\sqrt{\boldsymbol{\sigma}^2+\boldsymbol{\varepsilon}}} + \boldsymbol{\beta}\right) \tag{5}$$

First, the position-wise Feed-Forward Network Layer

The Position-wise Feed-Forward Network method is employed to input vectors at each position separately, thereby achieving non-linear properties for the model.

$$\mathbf{GELU(X) = x\Phi(x)} \tag{6}$$

And the activation function utilized is the Gaussian Error Linear Unit (GELU), as detailed in "Gaussian error linear units (gelus)." This method builds upon the ReLU function by incorporating statistical characteristics to enhance the model's expressive power.

Next, the embedding Layer.

The design of the embedding layer is of paramount importance in this model. The Transformer lacks any RNN or CNN modules, rendering it unable to directly perceive the order of the input sequence. To utilise the sequence order information, positional embeddings are introduced in the Transformer's embedding layer. Unlike common sinusoidal position encodings, learned position vectors are employed. The position vector matrix provides vector representations for any position; however, the maximum sequence length must be specified, and the input sequence is truncated accordingly.

Last, the output Layer.

In the output layer, a two-layer feedforward network with GELU activation functions is employed to generate the final output. For this model, the input and output layers share the item embedding matrix design, with the objective of mitigating overfitting and reducing model complexity.

3. METHOD ACCURACY COMPARISON

3.1. Dataset

The experiment uses the MovieLens dataset from the MovieLens community website, which contains rating data from thousands of users on thousands of movies. Each user has rated more than twenty movies, making it an excellent public dataset for this purpose.

3.2. Evaluation Metrics

To evaluate the effectiveness of the algorithms, the two methods proposed in this paper are compared with traditional knowledge graph-based and collaborative filtering algorithms. The commonly used evaluation metric is the confusion matrix, as shown below:

Table 2. Confusion Matrix for Recommendation Systems

	User Interested	User Not Interested
Correct Recommendation	Recommended and liked (TP)	Recommended but not liked (FP)
Incorrect Recommendation	Liked but not recommended (FN)	Not recommended and not liked (TN)

Precision (P):

$$\mathbf{P} = \frac{\mathbf{TP}}{\mathbf{TP+FP}} \tag{7}$$

3.3. Experimental Design

The MovieLens dataset containing 100,000 user movie ratings was filtered and randomly divided into 80% training data and 20% test data. Experiments were conducted using the xDeepFM and BERT

algorithms, with each experiment repeated 10 times. A movie recommendation was considered accurate (successful prediction) if the predicted rating was 4 or higher. The average of these 10 runs was taken as the final result.

To evaluate the overall effectiveness of deep learning-based intelligent recommendation algorithms, tests need to be conducted. Reference [1] introduces a method called the knowledge graph method, which integrates knowledge graphs into traditional collaborative filtering algorithms. It uses path ranking techniques to discover relationships between entities and embeds these relationships in a low-dimensional semantic space for semantic similarity computation. This method combines semantic similarity and user behavior similarity to perform video recommendations.

On the other hand, reference [2] proposed the tag weighting method, which constructs user, item, and user-to-user association matrices by optimizing the tag weighting algorithm to predict user needs. This method combines bipartite graphs and diffusion algorithms, and uses user tag similarity to perform video recommendations.

We applied the xDeepFM algorithm and the BERT model from this paper, along with the aforementioned methods, to the dataset and compared their accuracy.

3.4. Comparison of Accuracy

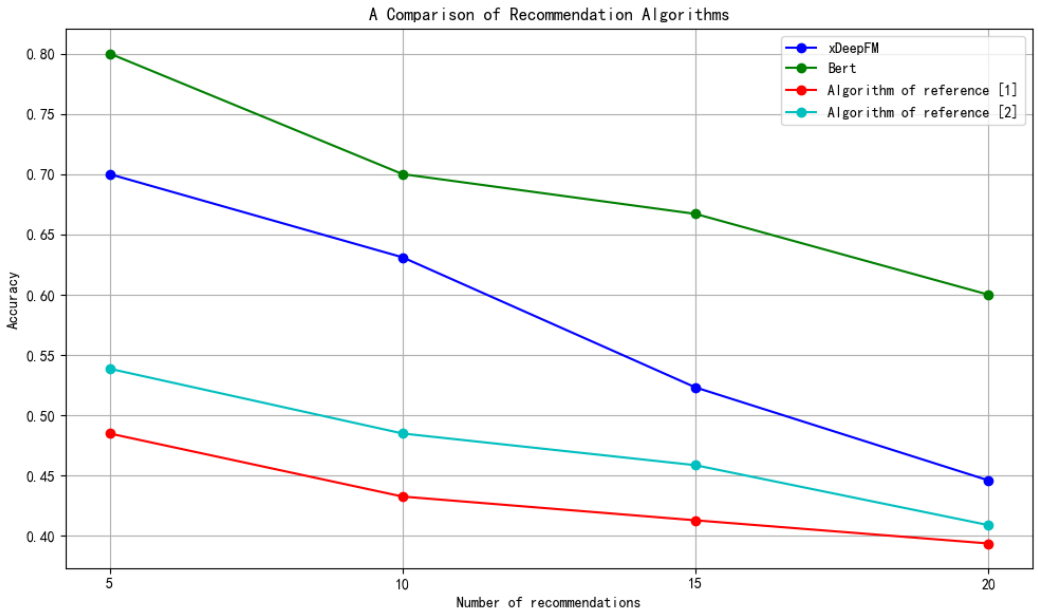

Figure 6. Comparison of the accuracy of different algorithmic models

It can be observed that the accuracy of all four methods decreases as the number of recommended videos increases. However, the accuracy of the deep learning-based xDeepFM and BERT algorithms is significantly higher than that of the traditional knowledge graph-based algorithm and the tag-based collaborative filtering algorithm.

Moreover, we found that when the dataset size is relatively small (less than 800,000 entries), the xDeepFM model performs exceptionally well, with an overall accuracy of more than 70%.

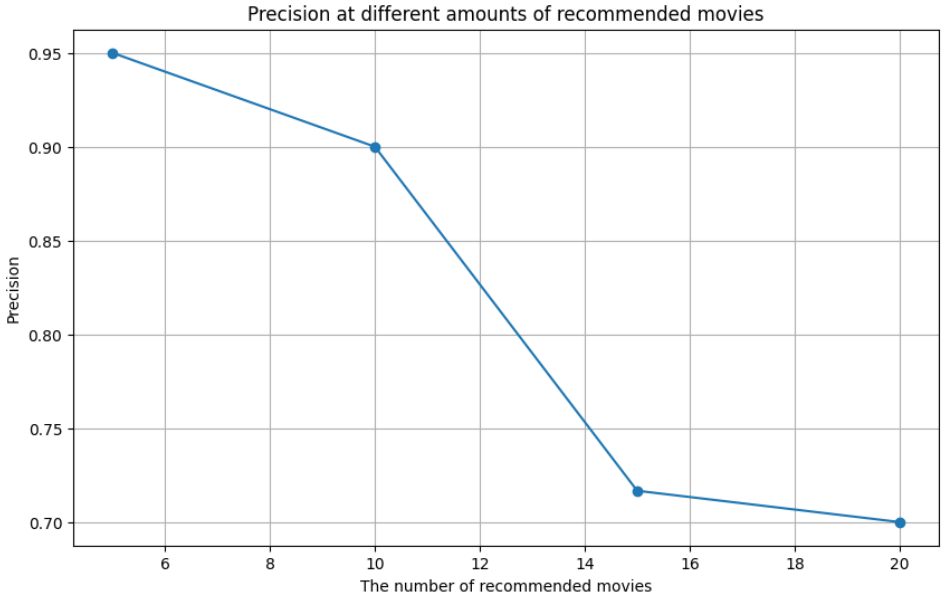

Figure 7. Accuracy of XDeepFM with a small dataset

4. CONCLUSION

This paper has analyzed the shortcomings of traditional recommendation systems in the context of the current popularity of recommendation systems. It provides a detailed introduction to deep learning-based methods such as xDeepFM and BERT, which optimize traditional recommendation systems to address issues such as cold start, low accuracy, long-tail effects, poor diversity, and information cocooning.

However, these algorithms are relatively new, and their theories and models are still being optimized. In future research, we will continue to learn and master more efficient and comprehensive deep learning algorithms and apply them to recommendation systems.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge Prof. Guo for his theoretical support, my parents for their financial and spiritual support, and Miss. Liu and Mr. Yang for their intellectual contributions and assistance.

REFERENCES

[1] Xu Zhihong, Zhao Xing, Dong Yongfeng, et al. Video Recommendation Algorithm Based on Knowledge Graph Knowledge Reasoning [J]. Computer Engineering and Design, 2020, 41(3): 710-715.

[2] Lei Man, Gong Qin, Wang Jichao, et al. Collaborative Filtering Algorithm Based on Tag Weight [J]. Computer Applications, 2019, 39(3): 634-638.

[3] Dan Hendrycks, Kevin Gimpel. Gaussian Error Linear Units (GELUs) [J] 2023.

[4] Bao Lin, Zhu Zhiyu, Sun Xiaoyan, Xu Biao, et al. A Review of Personalized Search and Recommendation Algorithms for Multi-Source Heterogeneous Data [J] 2024, 41(02): 189-209.

[5] Zhou Yongbo. Research on User Behavior Prediction in Short Videos Based on Deep Neural Networks [D] Nanjing University of Posts and Telecommunications, 2023.

[6] Meng Lu. Research on Click-Through Rate Prediction of Recommendation System Based on xDeepFM [D] Liaoning University of Science and Technology, 2022.

[7] Wen Yaoyao. Research on Click-Through Rate Prediction Method Based on Deep Learning with Attention Mechanism [D]. Beijing Jiaotong University, 2019.

[8] Sunny Sharma, Vijay Rana, Vivek Kumar. Deep Learning Based Semantic Personalized Recommendation System [J] 2021.

[9] Fawad Naseer, Muhammad Nasir Khan, Muhammad Tahir, Abdullah Addas, S.M. Haider Aejaz. Integrating Deep Learning Techniques for Personalized Learning Pathways in Higher Education [J] 2024.

[10] Li Shengyu, Wang Lei, Xu Wenchang, He Yuwei, Li Xinde. Gout Medical Record Classification Method Based on BERT and Improved Adversarial Training [J] Computer Engineering and Design 2024, 45(06), 1668-1673.

[11] Shen Xiaopeng, Zhao Ming, Liu Shanzhi, et al. Research on Short Video Recommendation Methods and Models [J]. Computer Knowledge and Technology 2023, 19(34): 116-118.