IFP-DeepFM: Integrating Improved FP-Growth and Attention-Based Deep Factorization Machines for User Purchase Behavior Modeling and E-commerce Recommendation

Ning Li*, Lihan Gu
Business School, Taishan University, Tai'an 271000, Shandong, China
E-mail: sdtsu2021@163.com; Lihan@163.com
*Corresponding author

Keywords: E-commerce platform, intelligent recommendation algorithm, marketing strategies, user purchase behavior

Received: October 30, 2025

This study examines how intelligent recommendation algorithms can be used to analyze user purchase behavior on e-commerce platforms and to support the development of more effective marketing strategies. First, a systematic review of existing recommendation algorithms is conducted, which shows that traditional methods have limitations when processing large-scale and complex user data. To address these issues, an Improved Frequent Pattern-growth (IFP-growth) algorithm is introduced. By incorporating time-window parameters, the algorithm mines frequent patterns from user behavior logs and extracts timely and relevant purchasing features. In parallel, an Attention-enhanced Deep Factorization Machine (DeepFM) is adopted, integrating the strengths of Factorization Machines and Deep Neural Networks. The attention mechanism further refines feature-interaction representations, enabling more accurate and personalized recommendations. To validate the effectiveness and robustness of the proposed model, experiments are conducted using real behavioral data from a major e-commerce platform over a three-month period, comprising approximately 1.2 million behavior logs, 80,000 active users, and 100,000 products. The IFP-DeepFM model is compared with traditional collaborative filtering, IFP-growth, and standard DeepFM algorithms. The results show that IFP-DeepFM outperforms DeepFM in precision, recall, F1-score, and the area under the precision–recall curve by 6.81%, 11.9%, 9.51%, and 11.5%, respectively. In practical metrics such as Click-Through Rate (CTR) and Conversion Rate (CVR), the model also achieves improvements of 10.4% and 8.9%. Based on these results, this study outlines several marketing strategy implications for e-commerce platforms, including personalized recommendations, targeted advertising, and user behavior prediction. These strategies enhance user satisfaction and platform performance while providing data-driven support for operational decision-making. The proposed IFP-DeepFM model demonstrates a practical approach to analyzing user purchase behavior and refining marketing strategies in large-scale e-commerce settings.

Povzetek: Študija predstavi model IFP-DeepFM, ki z rudarjenjem pogostih vzorcev in pozornostno okrepljenim DeepFM natančneje analizira nakupno vedenje ter izboljša personalizirana priporočila in marketinške strategije na e-trgovinskih platformah.

1 Introduction

In the context of today's digital and information age, e-commerce platforms have become an important channel for consumers' daily shopping. With the widespread adoption of the internet and mobile devices, the number of users on e-commerce platforms has rapidly increased, and the variety and quantity of goods on these platforms have shown explosive growth. Faced with such a large scale of data and complex user behaviors, accurately recommending products to users to enhance their shopping experience and boost platform sales has become a critical issue for e-commerce enterprises [1-3]. Recommendation systems, which can automatically suggest items that users might be interested in based on their historical behaviors and preferences, have garnered extensive attention and research. However, traditional recommendation algorithms often show limitations when dealing with massive data and complex user behaviors, making it challenging to meet the growing demands of e-commerce platforms.

Traditional recommendation algorithms, such as Collaborative Filtering (CF) and Content-based Recommendation, often face high computational complexity and poor real-time performance when processing large-scale data. These algorithms rely on explicit features of users and items or the similarity between users to make recommendations. However, in the rapidly changing e-commerce environment with vast amounts of data, traditional algorithms struggle to efficiently process complex and dynamic data. Additionally, traditional recommendation algorithms face challenges in addressing the cold start problem, where a lack of sufficient historical data for new users and new items results in less accurate and personalized recommendations [4,5]. Therefore, finding a recommendation algorithm that can efficiently handle large-scale complex data while maintaining strong real-time performance and accuracy has become a crucial direction for current recommendation system research.

To overcome the limitations of traditional recommendation algorithms, intelligent recommendation algorithms have gradually emerged in recent years and have achieved significant results in the field of recommendation systems. By incorporating machine learning and deep learning technologies, intelligent recommendation algorithms can better uncover potential patterns in user behavior data, improving the precision and personalization of recommendations. Techniques such as the Factorization Machine (FM) and Deep Neural Network (DNN) have been widely applied in recommendation systems, significantly enhancing the performance of recommendation algorithms [6,7]. However, despite these intelligent recommendation algorithms addressing some issues of traditional algorithms to a certain extent, they still fall short in handling the timeliness and relevance of user behavior data.

Existing research indicates that the application of recommendation systems on e-commerce platforms can not only enhance the shopping experience for users but also significantly increase platform sales and user retention rates. However, traditional recommendation algorithms exhibit high computational complexity, poor real-time performance, and cold start issues when dealing with large-scale data and complex user behaviors, making it difficult to meet the growing demands of e-commerce platforms. In recent years, with the advancement of machine learning and deep learning technologies, intelligent recommendation algorithms have become a hotspot in recommendation system research. The performance of recommendation algorithms has been effectively improved by introducing techniques such as FM and DNN. Nonetheless, despite these intelligent recommendation algorithms addressing some issues of traditional algorithms to a certain extent, they still fall short in handling the timeliness and relevance of user behavior data. Therefore, this work innovatively proposes the Improved Frequent Pattern - Attention Deep Factorization Machine (IFP-DeepFM) model. It intends to provide a new method and approach for analyzing user purchase behavior and developing marketing strategies for e-commerce platforms. The primary objective of this study is to address the limitations of existing recommendation systems in modeling data timeliness and behavioral relevance. The specific research questions include: (1) how to enhance the real-time extraction of user behavior features through an improved frequent pattern mining algorithm; (2) how to optimize the representation of feature interactions in the DeepFM model by incorporating an attention mechanism; and (3) how to achieve higher recommendation accuracy, click-through rate, and conversion rate based on the improved model.

2 Related work

In recent years, the rapid progress of e-commerce platforms has made the analysis of user purchase behavior and the research of intelligent recommendation algorithms a hot topic in academia and industry. Various scholars have explored the design and optimization of recommendation systems through different methods and models. Raja et al. proposed a recommendation algorithm based on matrix factorization, which introduced latent factor models to mitigate the issues caused by data sparsity. However, this method struggles with low computational efficiency when processing data with high real-time requirements and complexity [8]. Zhou et al. investigated recommendation systems based on deep learning, particularly the application of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) in recommendations. Their research indicated that deep learning models had significant advantages in capturing user behavior patterns and feature representation, but they required substantial computational resources for large-scale data training [9]. Ma et al. proposed the DeepFM model, which combines FM and DNN. This model improves recommendation accuracy and personalization by learning nonlinear feature interactions through deep networks. However, the DeepFM model has limitations in handling frequent pattern mining [10]. Zhou et al. explored a time-window-based frequent pattern mining algorithm and proposed an improved Apriori algorithm, achieving good results in user behavior analysis. Nonetheless, the high computational complexity of this method when handling large-scale data limits its real-time application [11]. Widiyaningtyas et al. studied the application of intelligent recommendation systems in e-commerce and proposed a personalized recommendation algorithm based on user profiling, significantly enhancing recommendation relevance and user satisfaction. However, this method performs inadequately in handling dynamically changing data [12]. Bhaskaran et al. proposed a hybrid recommendation algorithm that combines machine learning and data mining, improving the stability and accuracy of recommendation systems through the fusion of multiple models. However, the complexity of model training and parameter tuning increases the difficulty of system implementation [13]. Cui et al. studied recommendation systems based on attention mechanisms. By introducing attention mechanisms, they enhanced the model's ability to capture key features, improving the accuracy of recommendation results. However, this approach consumes significant computational resources when processing long sequence data [14]. The above methods are summarized in Table 1:

Table 1: Summary of methods in the literature

Algorithm	Techniques Used	Test Dataset	Key Performance Metrics	Limitations
Matrix Factorization	Latent factor decomposition	E-commerce rating data	Accuracy, recommendation relevance	Low computational efficiency under high real-time demands and complex data
CNN/RNN-Based Recommender Systems	Deep learning	User behavior sequences	Strong feature representation capability	Training on large-scale data requires substantial computational resources
DeepFM	FM + DNN	E-commerce behavior data	Accuracy, personalized recommendation	Limited capability in frequent pattern mining
Improved Apriori Algorithm	Frequent pattern mining	User behavior data	Effective user behavior analysis	High computational complexity for large-scale data; insufficient real-time performance
User Profiling Recommendation	Personalized recommendation	E-commerce user data	Recommendation relevance, user satisfaction	Limited ability to process dynamic data
Hybrid Recommendation Algorithms	Machine learning + data mining	Integrated datasets	Improved recommendation stability and accuracy	Complex model training; difficult parameter tuning
Attention-Based Recommendation	Attention mechanism + deep learning	User sequence data	Enhanced feature extraction capability	High computational cost when processing long sequence data

In summary, previous research has made significant progress in the design and optimization of intelligent recommendation algorithms, but some research gaps and deficiencies remain. Traditional CF and matrix factorization methods fall short in addressing data sparsity and the cold start problem. Additionally, while deep learning models excel in feature representation and pattern recognition, they demand high computational resources. Finally, existing frequent pattern mining algorithms face limitations in real-time performance and handling complex data. This work innovatively proposes the IFP-DeepFM model. By setting time window parameters for frequent pattern mining and combining the feature representation capabilities of DNN, this model overcomes the shortcomings of traditional algorithms in handling large-scale and complex data. The research results demonstrate that this method shows significant advantages in key metrics such as recommendation accuracy, recall rate, Click-Through Rate (CTR), and Conversion Rate (CVR). It provides new methods and insights for analyzing user purchase behavior and developing marketing strategies for e-commerce platforms.

3 IFP-Deepfm model construction

3.1 Advantages and disadvantages of traditional recommendation algorithms and their limitations in handling large-scale data

CF algorithms are among the most widely used techniques in recommendation systems, mainly divided into User-based CF and Item-based CF. User-based CF recommends items by identifying users with similar interests to the target user and suggesting items that similar users like. In contrast, Item-based CF recommends items to users by identifying items similar to a target item and suggesting these similar items to the users.

Content-based Filtering (CBF) algorithms recommend items by analyzing the content features of items that the user has liked in the past and suggesting items with similar content features. This algorithm typically uses natural language processing techniques to handle textual data and employs various similarity measurement methods (such as cosine similarity) for recommendations.

Hybrid Recommendation Systems combine the strengths of multiple recommendation algorithms to enhance recommendation effectiveness and system robustness. Common hybrid recommendation methods include linear weighted hybrid, cascade hybrid, and model fusion.
In the context of large-scale data processing, traditional recommendation algorithms generally suffer from high computational complexity, poor scalability, and data sparsity issues. CF algorithms require a large amount of user interaction data to ensure recommendation effectiveness, but their performance significantly decreases when data is sparse. CBF algorithms rely on the content features of items, demanding substantial computational resources to handle high-dimensional data. Although hybrid recommendation algorithms can optimize recommendation effectiveness, they also increase system complexity and computational resource requirements. These limitations become particularly pronounced when dealing with the massive and complex data on e-commerce platforms [15-17]. Therefore, improving the performance and efficiency of recommendation algorithms in large-scale data environments has become a crucial research direction in the field of recommendation systems. Table 2 summarizes the advantages and disadvantages of traditional recommendation algorithms.

Table 2: Advantages and disadvantages of traditional recommendation algorithms.

Recommendation Algorithm	Advantages	Disadvantages
CF Algorithm	Simple and easy to implement / Does not require content information	Cold start problem, data sparsity, poor scalability
CBF	Does not require user data / Strong interpretability	Overfitting issues, dependency on content features, high-dimensional data processing
Hybrid Recommendation Algorithm	Optimizes performance / Mitigates cold start and sparsity issues	High complexity, high computational resource requirements

3.2 Improved FP-growth (IFP-growth) algorithm

The IFP-growth algorithm is an enhanced version of the FP-growth algorithm, designed to improve the efficiency of frequent pattern mining in large-scale datasets. The FP-growth algorithm significantly enhances the efficiency of frequent pattern mining by compressing frequent itemsets into a frequent pattern tree (FP-Tree). Building on this foundation, the IFP-growth algorithm introduces improvement strategies to optimize performance in big data environments [18,19]. Figure 1 illustrates the e-commerce platform recommendation system based on the IFP-growth algorithm.

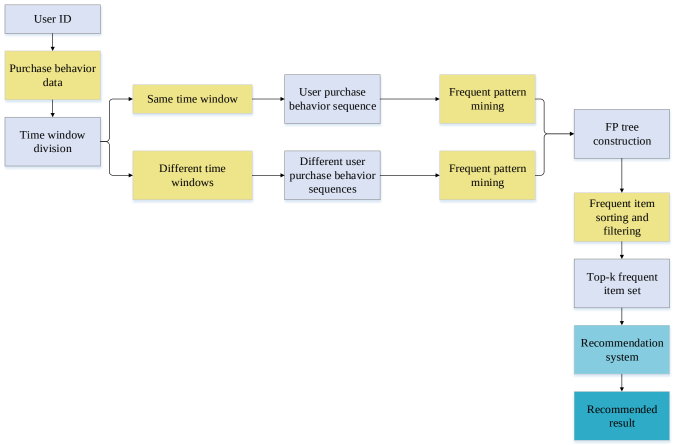

Figure 1: The E-commerce platform recommendation system based on the IFP-growth algorithm

When processing e-commerce platform data, the model's input includes user ID, item ID, purchase time, and purchase quantity. Based on this information, user behavior data must first be segmented into time windows. By setting a time threshold t (typically one month), user purchase behaviors within this time window are grouped into the same sequence; behaviors outside the time window are divided into different sequences. This method captures user purchasing preferences and behavior patterns, thereby improving recommendation accuracy.

The IFP-growth algorithm uses the FP-Tree structure to compress and store user purchase data. First, the dataset is scanned to construct the FP-Tree, and frequent items are sorted and filtered. The improvement of the IFP-growth algorithm lies in optimizing the construction of the FP-Tree and the frequent itemset mining process, handling large-scale datasets through efficient conditional FP-Tree mining. This algorithm requires only two database scans, significantly reducing computational overhead. Equation (1) displays the time complexity O of the FP-Tree construction process:

$$O = f(n) - f(n-1) + \sum_{i=1}^{t} g(i) \quad (1)$$

t represents the time window, and $g(i)$ denotes the complexity of processing the i-th time period within the time window. n is the total number of items in the database. $f(n)$ and $f(n-1)$ represent the time complexity required to process a database containing n or $n-1$ items, respectively, during the FP-Tree construction.

After frequent pattern mining, the IFP-growth algorithm sorts and filters the frequent itemsets according to their support. The recommendation results are usually based on the Top-k frequent itemsets or by selecting the items with the highest occurrence frequency for recommendation. This method ensures the timeliness and relevance of the recommendation results, enhancing the recommendation effectiveness for new users. Figure 2 shows the IFP-growth algorithm process.

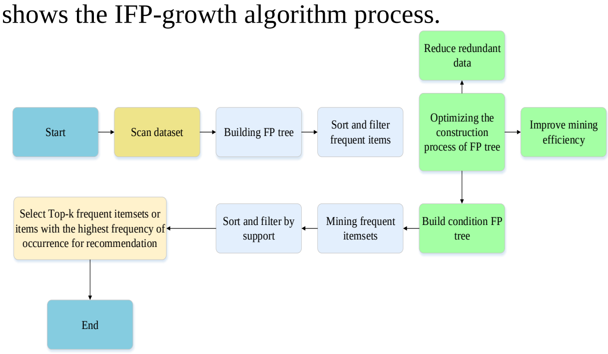

Figure 2: The IFP-growth algorithm process.

3.3 DeepFM algorithm

The Deep Factorization Machine (DeepFM) algorithm is a hybrid recommendation model that integrates FM with DNN. DeepFM retains the dual-model architecture of the Wide & Deep model, but replaces the Wide component with FM, thereby enhancing the feature interaction capability of the shallow network. This model leverages the parallel structure to merge the advantages of both FM and DNN. It effectively handles both first-order features and second-order features generated from the interactions of first-order features, as well as high-order features resulting from multiple interactions among first-order features. This structure supports end-to-end training, obviating the need for complex feature engineering, and shares the same input and Embedding vectors, thereby improving training efficiency [20]. Figure 3 illustrates the architecture of the DeepFM model.

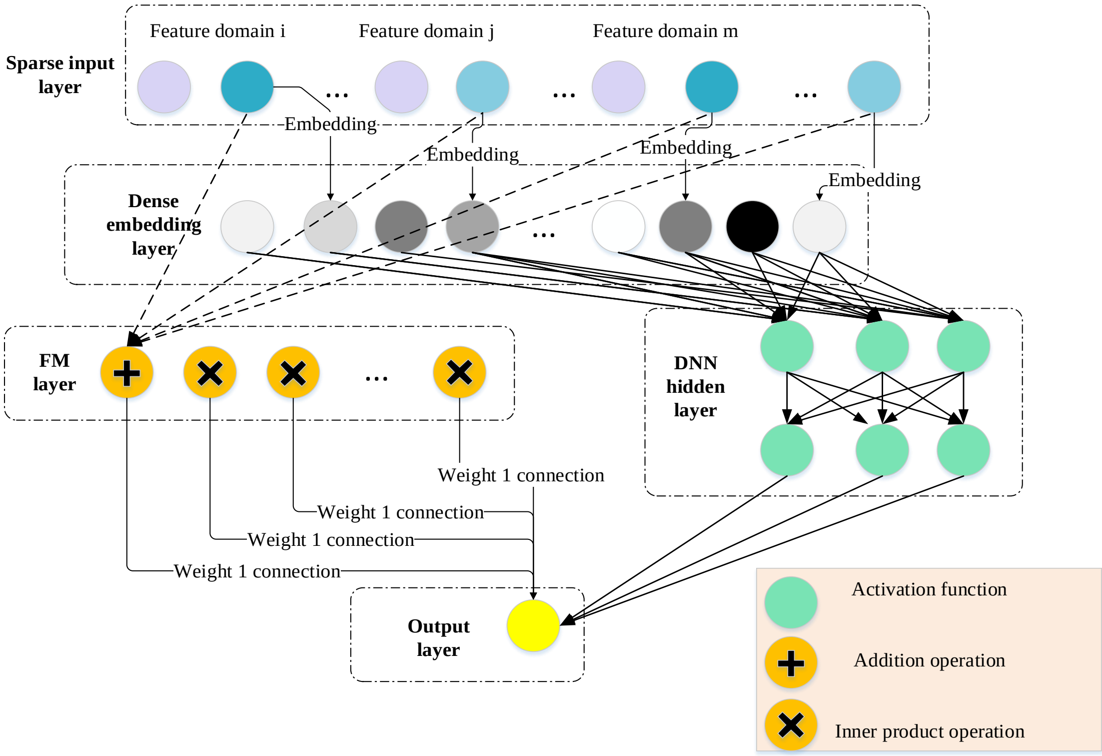

Figure 3: The architecture of the DeepFM model.

The DeepFM model consists of several key components: the Embedding layer, the FM component, the DNN component, and the Output layer. The input to DeepFM comprises multi-domain data, including user information, item information, and user historical behavior. There are a total of m user and item features. Input data may include both continuous variables and categorical variables encoded using One-Hot encoding. To address the high-dimensional sparse feature vectors of categorical variables, DeepFM incorporates an Embedding layer. This layer maps these high-dimensional sparse features into a lower-dimensional space where the vector elements are non-zero. The Embedding layer performs Embedding operations for each feature, and the resulting vectors are shared between the second-order interaction calculations of the FM component and the DNN component.

The FM component is divided into two parts: the linear part and the interaction part. The linear part assigns a weight to each feature and computes a weighted sum. The mathematical formulation of this process is as follows Equation (2):

$$Linear\ Part = \sum_{i=1}^{m} w_i x_i \qquad (2)$$

w_i represents the weight for the feature x_i. m is the total number of features. The interaction part involves pairwise multiplication of features, assigns weights to these interactions, and then performs a weighted sum. This mathematical formulation is expressed as follows Equation (3):

$$Cross\ Part = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=i+1}^{m} < v_i, v_j > x_i x_j \quad (3)$$

v_i and v_j represent the vector embeddings of features x_i and x_j, respectively, and $< v_i, v_j >$ denotes their inner product. The output of the FM component is the sum of the linear part and the interaction part, as expressed in Equation (4):

$$y_{FM}(x) = \sum_{i=1}^{m} w_i x_i + \frac{1}{2}\sum_{i=1}^{m}\sum_{j=i+1}^{m} < v_i, v_j > x_i x_j \quad (4)$$

The DNN component is a feedforward neural network whose input consists of the outputs from the Embedding layer. The processing in the DNN component involves forward propagation, where it is assumed that $a^{(0)} = (e_1, e_2, \cdots, e_m)$ represents the output of the Embedding layer. The DNN processes these inputs through multiple layers of linear transformations and nonlinear activations to produce the final output. The mathematical formulation of the feedforward process is given by Equation (5):

$$a^{(l+1)} = \sigma(W^{(l)}a^{(l)} + b^{(l)}) \quad (5)$$

$W^{(l)}$ and $b^{(l)}$ denote the weights and biases of the l-th layer, respectively, and σ represents the activation function. The output layer of DeepFM combines the results from both the FM and DNN components by summing them. This combined result is then passed through a sigmoid activation function to perform a nonlinear transformation, yielding the final prediction probability. The mathematical formulation is as follows Equation (6):

$$\hat{y}(x) = \sigma(y_{FM}(x) + y_{DNN}(x)) \quad (6)$$

σ denotes the sigmoid activation function. $\hat{y}(x)$ is the model's predicted probability, and $y_{DNN}(x)$ is the output of the DNN part.

The DeepFM model features end-to-end training capabilities, which simplify the feature engineering process and enhance training efficiency. Both the FM and DNN components share the same input and embedding vectors, reducing redundant computations. By combining FM and DNN, DeepFM effectively captures second-order interactions among first-order features and complex combinations of higher-order features, thereby improving recommendation accuracy and personalization.

Here, an attention mechanism is introduced to automatically learn the importance of each feature interaction during the weighting process. The core idea behind the attention mechanism is to allocate different weights to features based on their importance when processing multiple feature representations. This approach optimizes the representation of feature interactions. In the DeepFM model, the attention mechanism enhances the precision and personalization of feature interactions by weighting and summing interaction vectors. Specifically, the attention scores for feature interactions are calculated as follows Equation (7):

$$a_{ij} = softmax(w^T \cdot (x_i \circ x_j) + b) \quad (7)$$

The attention score a_{ij} for feature interactions reflects the importance of features x_i and x_j in the prediction task. The symbol $\circ$ denotes the dot product operation between feature interactions, with w and b representing model parameters. By calculating these attention scores, this work quantifies the relative importance of each feature pair in the prediction. To address convergence issues related to weight parameters caused by data sparsity, a Multi-Layer Perceptron (MLP) is introduced to further parameterize the attention scores. The input to the attention network is the interaction vector z of the two features, and its encoded information is represented by Equation (8):

$$Attention\ Network(z) = softmax(W_{att} \cdot ReLU(W_{mlp} \cdot z + b_{mlp}) + b \quad (8)$$

W_{att}, W_{mlp}, and b_{mlp} are model parameters, where ReLU is the activation function, and the softmax function is used to normalize the attention scores. This network effectively addresses the sparsity of feature interactions and maps the attention scores to a standardized range. Finally, a weighted interaction vector is output from the attention-based aggregation layer, represented by Equation (9):

$$f_{att} = \sum_{(i,j)\in R} a_{ij} \cdot (x_i \circ x_j) \quad (9)$$

By performing a weighted sum of all feature interactions, the model can focus on more significant interactions, thereby enhancing prediction accuracy. To train and optimize the attention-based DeepFM model, the cross-entropy loss function is used to measure the discrepancy between the model's predictions and the actual labels. For a binary classification problem, the loss function is defined as follows Equation (10):

$$L(\theta) = -\frac{1}{N}\sum_{n=1}^{N}[y_n \log(\hat{y}_n) + (1 - y_n)\log(1 - \hat{y}_n)] \quad (10)$$

y_n is the true label, $\hat{y}_n$ is the model's predicted value, and θ represents the model parameters. N is the total number of samples. The model parameters are optimized using gradient descent. This work employs the Adam optimizer, which combines momentum and adaptive learning rate adjustments, providing enhanced convergence and stability. The update rule for the Adam optimizer is as follows Equation (11):

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \cdot \hat{m}_n \quad (11)$$

η is the learning rate. $\hat{m}_n$ and $\hat{v}_t$ represent the moving averages and variance estimates of the gradients, respectively, and ϵ is a small constant to prevent division by zero errors. To prevent overfitting, L2 regularization is incorporated into the model, adding a regularization term to the loss function. This is expressed as follows Equation (12):

$$L_{reg} = L(\theta) + \lambda\|\theta\|^2 \quad (12)$$

λ represents the regularization strength hyperparameter. Regularization helps improve the model's generalization ability.

3.4 Algorithm fusion

This work introduces the IFP-DeepFM model, which integrates the IFP-growth algorithm with the DeepFM algorithm to leverage the strengths of both approaches. The IFP-growth algorithm is used to mine frequent patterns from user behavior data, while the DeepFM algorithm utilizes its deep learning capabilities for precise recommendations. Figure 4 illustrates the recommendation model based on IFP-DeepFM.

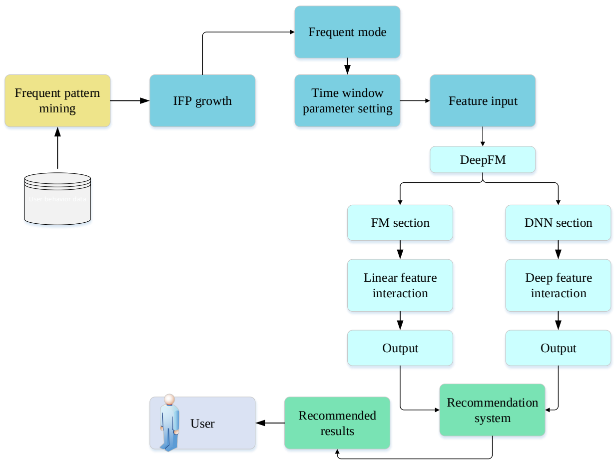

Figure 4: Recommendation model based on IFP-DeepFM.

In practical implementation, the IFP-growth algorithm first segments user behavior data using time windows of 7 or 14 days, and then mines frequent interaction patterns within each window, such as repeated browsing or purchasing of a specific category of products. The extracted patterns are subsequently transformed into additional input features for DeepFM. These features can be represented in the form of binary indicators, frequency counts, or embedding vectors. To ensure consistency with other DeepFM inputs, the pattern features are normalized prior to fusion; categorical patterns are encoded using one-hot encoding or embeddings, and then concatenated with user–item features to form the final input vector. The DeepFM model, which combines the strengths of Factorization Machines and Deep Neural Networks, further enhances feature-interaction representation through the incorporated attention mechanism.

3.5 Experimental design

To verify the effectiveness and robustness of the proposed IFP-DeepFM model in e-commerce recommendation scenarios, experiments are conducted using real user behavior data from a major e-commerce platform. The dataset covers user browsing, clicking, add-to-cart, and purchasing behaviors. The data span a continuous three-month period and contain approximately 1.2 million behavior logs, 80,000 active users, and 100,000 products. Each record includes a user ID, product ID, behavior type, timestamp, and several auxiliary attributes such as product category, price range, and device type.

After data collection, preprocessing is performed by removing missing values, outliers, and duplicate records. The timestamp field is standardized and sorted in chronological order to preserve behavioral sequence characteristics. Categorical features are processed using one-hot encoding, while continuous features are normalized via Min–Max scaling. For frequent pattern mining, time windows of 7 and 14 days are applied to extract high-frequency user interaction patterns. The resulting patterns are incorporated into the model input as follows: for each user–product pair, the presence of a pattern is represented using a binary indicator (0/1), or alternatively using count-based or embedding-based encodings, which are then fused with DeepFM's feature vectors. For example, if a user repeatedly purchases product A within a 7-day window, the pattern "purchase of product A" is encoded as 1 or accumulated as a count in the input vector. This process generates the user–product interaction matrix and the corresponding input feature vectors. To prevent data leakage and ensure reproducibility, a time-aware data split is adopted, dividing the dataset chronologically into 70% training, 15% validation, and 15% testing. A group-wise split is additionally applied to ensure that all sessions belonging to the same user remain within a single subset. All experiments are repeated five times using a fixed random seed (seed = 42), and the mean ± standard deviation (SD) of the key metrics are reported. During validation, five-fold cross-validation is employed to assess the generalization capability of the model.

The experiments are conducted on a Windows 10 operating system with an Intel Core i5-10400F processor, 16 GB of RAM, and an NVIDIA GeForce GTX 1650 GPU with 4 GB of memory. The model is implemented using Python 3.8 and the PyTorch 1.x deep learning framework. Hyperparameters are selected through grid search on the validation set. For the IFP-growth component, the minimum support threshold is set to 0.02, the confidence threshold to 0.5, and the time window to 7 days. The mined frequent patterns are incorporated into the model as binary, count-based, or embedding-based features. For the DeepFM component, the embedding dimension is set to 64, and the DNN architecture consists of layers with sizes [128, 64, 32]. ReLU is used as the activation function. The model is optimized with Adam, using a learning rate of 0.001, a batch size of 256, and a maximum of 50 training epochs. Early stopping is applied based on validation AUC. The attention module adopts a two-layer attention network with a weight dimension of 16 and uses the softmax function for normalization. Dropout (rate = 0.5) and L2 regularization (λ = 1e–5) are applied between network layers to prevent overfitting.

For comparative experiments, the following baseline algorithms are selected: traditional collaborative filtering (with number of neighbors k = 20); IFP-growth (using the same support and confidence thresholds, performing frequent-pattern-based recommendation only); DeepFM (with the same structure as this study but without time windows or the attention mechanism); and Gradient-Boosted Trees (CatBoost), trained on sparse/tabular features under the same train/validation/test splits [21]. The model output is the predicted purchase probability, and the positive-class threshold is fixed at 0.5 for binary decisions. To address the class imbalance caused by the relatively low proportion of purchase behaviors, class weighting is applied to positive and negative samples during the training of DeepFM and IFP-DeepFM, allowing the models to be more sensitive to positive samples. Model performance is evaluated using Precision, Recall, F1-score, the Area Under the Precision–Recall Curve (PR-AUC), CTR, and CVR. CTR and CVR are computed via offline replay using the same time span as the test set. All results are reported as mean ± SD, along with 95% confidence intervals to ensure statistical reliability and business interpretability.

4 Experimental results of the IFP-Deepfm-based recommendation model

4.1 Model performance verification

This work conducts comparative experiments between traditional recommendation algorithms and the IFP-DeepFM model to evaluate their performance in analyzing user purchasing behavior on e-commerce platforms. Figure 5 displays the recommendation results of the various algorithms.

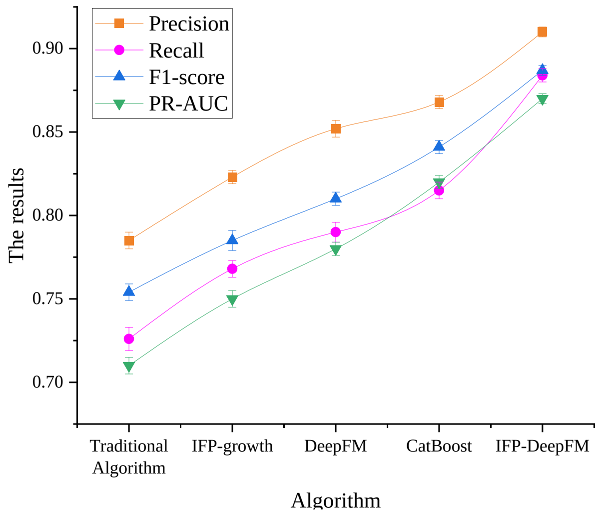

Figure 5: Recommendation performance of different algorithms

Comparative experiments between traditional recommendation algorithms and the IFP-DeepFM model are conducted to evaluate their performance in analyzing user purchasing behavior on e-commerce platforms. In Figure 5, the traditional algorithm shows the weakest performance, with precision, recall, and F1-score of 0.785, 0.726, and 0.754, respectively, and a PR-AUC of 0.71. The 95% confidence interval for the F1-score is [0.749, 0.759]. With the support of frequent pattern mining, IFP-growth achieves a slight improvement. DeepFM further enhances the evaluation metrics. As a mature gradient-boosted tree baseline, CatBoost attains precision, recall, and F1-scores of 0.868, 0.815, and 0.841, respectively, with a PR-AUC of 0.82 and an F1-score confidence interval of [0.837, 0.845], indicating strong performance on sparse, tabular features. In contrast, IFP-DeepFM integrates the frequent pattern mining capability of IFP-growth with the deep feature representation of DeepFM, achieving precision, recall, and F1-scores of 0.91, 0.884, and 0.887, respectively, along with a PR-AUC of 0.87. These metrics represent improvements over DeepFM of 6.81%, 11.9%, 9.51%, and 11.5%, respectively. The 95% confidence interval for the F1-score is [0.883, 0.891], outperforming all baseline methods. Results from paired t-tests show that the performance gains of IFP-DeepFM over the traditional algorithm, IFP-growth, DeepFM, and CatBoost are statistically significant ($p < 0.05$), further demonstrating its substantial enhancement in recommendation effectiveness.

Figure 6 displays the performance of algorithms in terms of CTR and CVR.

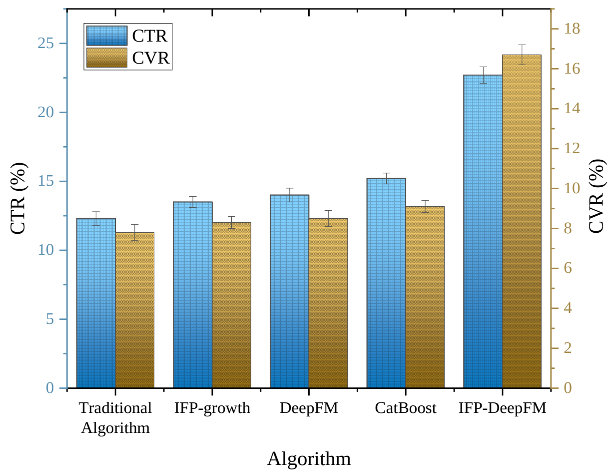

Figure 6: Comparison of CTR and CVR results for different algorithms.

In Figure 6, traditional algorithms report CTR and CVR values of 12.3% and 7.8%, respectively. The IFP-growth algorithm improves these metrics to 13.5% and 8.3%. The DeepFM algorithm further enhances these indicators to 14.0% and 8.5%. As a gradient-boosted tree baseline, CatBoost achieves CTR and CVR values of 15.2% and 9.1%, respectively. In contrast, the IFP-DeepFM model demonstrates the most substantial improvement, achieving a CTR of 22.7% and a CVR of 16.7%, with corresponding 95% confidence intervals of [21.5%, 23.9%] and [15.7%, 17.7%]. Results from paired t-tests indicate that the increases in both CTR and CVR achieved by IFP-DeepFM over all baseline models are statistically significant ($p < 0.01$). These results indicate that the IFP-DeepFM model not only excels in recommendation effectiveness but also significantly boosts user engagement and purchase intent. This is crucial for marketing strategies and user experience on e-commerce platforms.

Figure 7 illustrates the impact of personalized recommendations on user satisfaction.

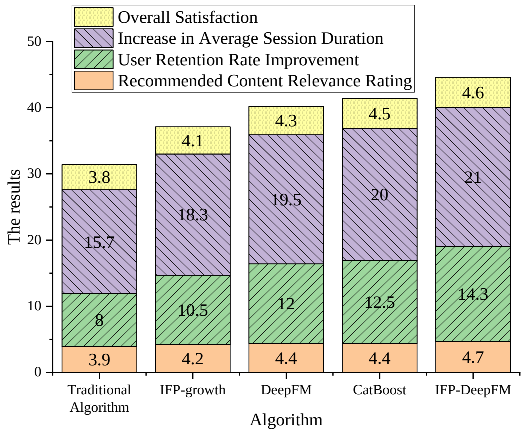

Figure 7: Impact of personalized recommendations on user satisfaction.

In Figure 7, traditional algorithms achieve scores of 3.9 for content relevance, 8.0% for user retention improvement, 15.7 minutes for average session duration, and 3.8 for overall user satisfaction. The IFP-growth algorithm improves these metrics to 4.2, 10.5%, 18.3 minutes, and 4.1, respectively. The DeepFM algorithm further enhances these figures to 4.4, 12.0%, 19.5 minutes, and 4.3. CatBoost performs slightly better than the other baseline models but remains inferior to IFP-DeepFM. Its content relevance score reaches 4.3, user retention improves by 12.5%, the average session duration is 20.0 minutes, and the overall satisfaction score is 4.4. Most notably, the IFP-DeepFM model excels with scores of 4.7 for content relevance, 14.3% for user retention improvement, 21.0 minutes for average session duration, and 4.6 for overall user satisfaction. These results indicate that the personalized recommendation system significantly improves user satisfaction and platform retention rates through more precise capturing of user needs, while also increasing user engagement and interaction time on the platform.

Figure 8 illustrates the effect of targeted advertising on platform sales.

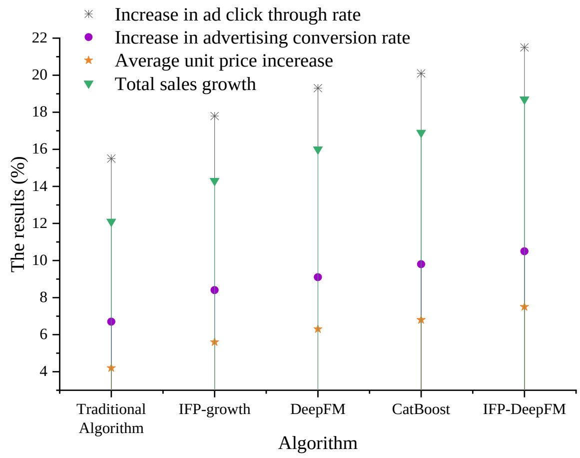

Figure 8: Impact of precise advertising on platform sales.

In Figure 8, traditional algorithms achieve increases of 15.5% in ad CTR, 6.7% in ad conversion rate, 4.2% in average order value, and 12.1% in total sales. The IFP-growth algorithm improves these metrics to 17.8%, 8.4%, 5.6%, and 14.3%, respectively. The DeepFM algorithm further enhances these figures to 19.3%, 9.1%, 6.3%, and 16.0%. Its content relevance score reaches 4.3, user retention improves by 12.5%, the average session duration is 20.0 minutes, and the overall satisfaction score is 4.4. Most notably, the IFP-DeepFM model demonstrates exceptional performance with increases of 21.5%, 10.2%, 7.5%, and 18.7% in these metrics. This indicates that the IFP-DeepFM model not only significantly boosts the attractiveness and conversion efficiency of ads but also enhances the overall sales performance of the platform. This is crucial for marketing strategies and economic benefits in e-commerce.

Figure 9 illustrates the effect of user behavior prediction on various aspects of platform operations and decision support.

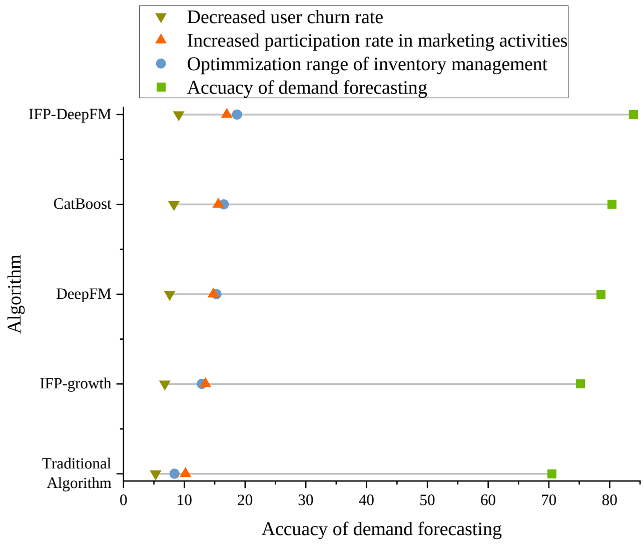

Figure 9: Impact of user behavior prediction on platform operational decision support.

In Figure 9, traditional algorithms achieve performance levels of 70.5% in demand forecast accuracy, 8.4% in inventory management optimization, 10.2% in marketing activity participation, and a 5.3% reduction in user churn rate. The IFP-growth algorithm improves these metrics to 75.2%, 12.9%, 13.5%, and 6.8%, respectively. The DeepFM algorithm further enhances these values to 78.6%, 15.3%, 14.8%, and 7.6%. CatBoost demonstrates robust performance on structured behavioral data, achieving a demand forecasting accuracy of 80.4%, a 16.5% improvement in inventory management, a 15.6% increase in marketing campaign engagement, and an 8.3% reduction in user churn. Most notably, the IFP-DeepFM model achieves exceptional results with 83.9% in demand forecast accuracy, 18.7% in inventory management optimization, 17.0% in marketing activity participation, and a 9.1% reduction in user churn rate. These results demonstrate that the IFP-DeepFM model significantly improves platform operational efficiency and decision support capabilities, while also reducing user churn and enhancing market competitiveness.

To further evaluate the scalability and computational complexity of the proposed model, we conducted a comparative analysis of training time, inference time, and peak memory usage across all models. The results are summarized in Table 3.

Table 3: Comparison of model training time and memory usage.

Algorithm	**Training Time (s)**	**Inference Time per 1000 Samples (ms)**	**Peak Memory Usage (GB)**
Traditional Algorithm	185	6.3	1.2
IFP-growth	294	8.1	1.8
DeepFM	412	9.5	2.6
CatBoost	376	8.7	2.3
IFP-DeepFM	528	11.2	3.1

As shown in Table 3, the traditional algorithm exhibits the shortest training and inference time but also the lowest recommendation performance. Due to the incorporation of frequent pattern mining, IFP-growth incurs slightly higher computational overhead. DeepFM and CatBoost achieve a balanced trade-off between accuracy and efficiency, with CatBoost delivering relatively fast training on medium-scale datasets. Although the IFP-DeepFM model requires more training time and memory—28.2% longer training time than DeepFM and a peak memory usage of 3.1 GB—it achieves substantial improvements in F1-score and PR-AUC. Overall, IFP-DeepFM strikes a reasonable balance between accuracy and computational cost, making it suitable for e-commerce recommendation scenarios involving medium-to-large-scale datasets.

To validate the model's generalizability and cross-platform applicability, two additional datasets—Amazon Product Reviews and the MovieLens user rating dataset—were incorporated into the experiments. All models were evaluated under the same data-splitting scheme (70% training, 15% validation, 15% testing) and identical hyperparameter settings to ensure comparability. The results are presented in Table 4.

Table 4: Generalization performance across different datasets.

Algorithm	**Amazon F1-score (±SD)**	**Amazon PR-AUC**	**MovieLens F1-score (±SD)**	**MovieLens PR-AUC**
Traditional Algorithm	0.742 ± 0.006	0.68	0.725 ± 0.007	0.66
IFP-growth	0.771 ± 0.005	0.72	0.758 ± 0.006	0.70
DeepFM	0.804 ± 0.005	0.77	0.782 ± 0.006	0.74
CatBoost	0.829 ± 0.004	0.80	0.808 ± 0.005	0.77
IFP-DeepFM	0.878 ± 0.004	0.85	0.865 ± 0.005	0.83

As shown in Table 4, the IFP-DeepFM model consistently achieves the highest F1-score and PR-AUC across both external datasets. Its F1-score reaches 0.878 on the Amazon dataset (95% CI: [0.870, 0.886]) and 0.865 on the MovieLens dataset (95% CI: [0.856, 0.874]), representing improvements of 9.2% and 10.6% over DeepFM, respectively. Paired t-test results confirm that the improvements of IFP-DeepFM over the four baseline models are statistically significant ($p < 0.05$). These findings demonstrate that the proposed IFP-DeepFM model exhibits strong generalization capability and robustness across platforms and across different types of user behavior data.

Furthermore, a synthetic cold-start scenario was constructed on the original e-commerce dataset. Active users were randomly sampled and grouped by historical interaction counts. For each user, only the most recent 1, 2, or 5 interactions were retained to simulate severe, moderate, and mild cold-start situations, respectively. A similar truncation strategy was applied to items. The model training still followed a time-aware splitting strategy, and the test set included these truncated cold-start users/items. All models (Traditional, IFP-growth, DeepFM, CatBoost, IFP-DeepFM) were run five times under the same settings. F1-score and PR-AUC were reported, and paired t-tests were used to evaluate the significance of differences between IFP-DeepFM and the baseline models. The results are shown in Table 5.

Table 5: Performance comparison under synthetic cold-start scenarios.

Scenario	Model	F1-score (±SD)	PR-AUC
Severe Cold Start (1 history record retained)	Traditional	0.432 ± 0.010	0.41
	IFP-growth	0.458 ± 0.009	0.44
	DeepFM	0.482 ± 0.008	0.47
	CatBoost	0.493 ± 0.007	0.49
	IFP-DeepFM	0.539 ± 0.006	0.55
Moderate Cold Start (2 history records retained)	Traditional	0.514 ± 0.009	0.50
	IFP-growth	0.541 ± 0.008	0.53
	DeepFM	0.566 ± 0.007	0.56
	CatBoost	0.578 ± 0.006	0.58
	IFP-DeepFM	0.621 ± 0.005	0.63
Mild Cold Start (5 history records retained)	Traditional	0.643 ± 0.007	0.62
	IFP-growth	0.669 ± 0.006	0.65
	DeepFM	0.694 ± 0.005	0.69
	CatBoost	0.707 ± 0.005	0.71
	IFP-DeepFM	0.738 ± 0.004	0.74

The results in Table 5 indicated that IFP-DeepFM consistently outperformed all baseline models under the three levels of cold-start severity, with the most substantial advantage observed when only a single historical interaction was retained. In this scenario, the model achieved a 9.33% improvement in F1-score over the best-performing baseline, CatBoost, demonstrating the largest relative gain. Paired t-tests further showed that the improvement of IFP-DeepFM over CatBoost and DeepFM was statistically significant across all cold-start conditions ($p < 0.05$). These findings suggested that the time-window-based frequent patterns extracted by IFP-growth provided valuable behavioral cues for users and items with extremely sparse histories. Meanwhile, the attention-enhanced DeepFM component more effectively exploited these sparse pattern features, thereby improving predictive performance under cold-start settings.

Table 6 reports the attention-weight distribution and the top-ranked frequent patterns identified by the IFP-DeepFM model. The results showed that short-term repeated browsing and purchase behaviors had the strongest influence on recommendation outcomes. For example, the pattern "repeated browsing of electronic product A followed by the purchase of accessory B" received the highest attention weight of 0.215. Other patterns involving consecutive add-to-cart actions, item collection, and repeated searches also exhibited high attention weights, indicating the model's capability to capture influential behavioral interactions. These high-impact patterns provided actionable insights for e-commerce marketing strategies—for instance, pushing bundle recommendations to users who frequently added items to the cart without purchasing, or delivering time-sensitive advertisements for categories with high browsing intensity. Overall, the integration of attention mechanisms and frequent-pattern features not only enhanced predictive performance but also improved model interpretability, offering data-driven support for marketing decision-making.

Table 6: Top 10 most important IFP frequent patterns and attention weights in IFP-DeepFM.

Rank	Frequent Pattern	Avg. Attention Weight
1	Repeated browsing of "Electronic Product A" and purchase of "Accessory B"	0.215
2	Adding "Clothing C" to cart and immediately purchasing "Shoes D" within 7 days	0.198
3	Browsing "Home Product E" more than three times	0.176
4	Purchasing "Book F" followed by browsing "Stationery G"	0.162
5	Browsing "Cosmetics H" for two consecutive days	0.149
6	Browsing "Food I" and adding "Beverage J" to cart	0.135
7	Collecting "Sports Equipment K"	0.128
8	Browsing "Digital Product L" more than five times	0.121
9	Purchasing "Toy M" during weekends	0.114
10	Repeatedly searching for "Maternal and Infant Product N"	0.107

4.2 Ablation study

To assess the contribution of each component of the IFP-DeepFM model, an ablation study was conducted by evaluating several model variants: DeepFM without the attention mechanism, DeepFM without the IFP-growth frequent-pattern features, and DeepFM without the time-window module. All models were trained and tested under identical data splits, and each experiment was repeated five times. The results, including the mean ± standard deviation of Precision, Recall, F1-score, and PR-AUC, are shown in Table 7.

Table 7: Ablation study results.

Model Variant	Precision ± SD	Recall ± SD	F1-score ± SD	PR-AUC
DeepFM w/o Attention	0.876±0.004	0.848±0.005	0.862±0.004	0.85±0.005
DeepFM w/o IFP Features	0.881±0.003	0.854±0.004	0.867±0.003	0.86±0.004
DeepFM w/o Time Window	0.887±0.003	0.860±0.004	0.873±0.003	0.865±0.004
Full IFP-DeepFM	0.910±0.003	0.884±0.004	0.887±0.003	0.87±0.003

As shown in Table 7, each component contributed meaningfully to overall model performance. Removing the attention mechanism reduced the F1-score to 0.862, confirming the importance of attention in modeling higher-order feature interactions. Excluding the IFP-growth frequent-pattern features decreased the F1-score to 0.867, demonstrating the value of frequent patterns in enhancing personalized recommendations. Eliminating the time-window parameter resulted in an F1-score of 0.873, further validating the effectiveness of temporal constraints in capturing time-sensitive user behavior. The complete IFP-DeepFM model achieved the best performance across all metrics, with Precision, Recall, F1-score, and PR-AUC reaching 0.910, 0.884, 0.887, and 0.87, respectively. Paired t-tests showed that improvements over the model variants were statistically significant ($p < 0.01$), indicating that each component made a substantial contribution to overall performance. These ablation results quantitatively demonstrated the importance of the attention mechanism, IFP-based features, and the time-window strategy, providing empirical support for the model architecture. Overall, the proposed IFP-DeepFM model exhibited significant advantages in e-commerce recommendation tasks. By integrating high-frequency behavioral patterns extracted by IFP-growth with DeepFM's deep feature-interaction capability, the model achieved notable improvements in accuracy, recall, and F1-score, and outperformed traditional collaborative filtering, IFP-growth, DeepFM, and CatBoost baselines in CTR, CVR, and user-satisfaction metrics. The ablation study further confirmed the complementary contributions of frequent-pattern features, temporal modeling, and the attention mechanism, demonstrating the effectiveness and robustness of the proposed approach.

4.3 Discussion

This study evaluated the effectiveness and robustness of the IFP-DeepFM model for recommendation tasks using real user behavior data from a large-scale e-commerce platform. Experimental results demonstrated that IFP-DeepFM significantly outperformed traditional collaborative filtering, IFP-growth, and DeepFM baselines across key metrics, including Precision, Recall, F1-score, PR-AUC, CTR, and CVR. The model particularly excelled at capturing short-term user behavior patterns and high-frequency interaction features, indicating that the integration of IFP-growth with attention-enhanced DeepFM substantially improves recommendation performance in sparse data environments and enables more accurate prediction of user purchase behaviors. Shi et al. proposed a quantum-enhanced recommendation system for e-commerce platforms. Their approach leveraged quantum principal component analysis and quantum similarity computation, achieving an 87.3% reduction in execution time and a 15.8% improvement in recommendation accuracy across three datasets. The system also demonstrated logarithmic scalability as the data volume increased [22]. Although their approach offers advantages in execution time and precision, it relies on quantum computing resources, whereas IFP-DeepFM operates efficiently on conventional hardware and captures user behavior patterns effectively via time windows and attention mechanisms, achieving comparable gains in recommendation accuracy. Nguyen et al. developed an implicit personalized recommendation system combining collaborative filtering, popularity ranking, and Bayesian personalized ranking, using a lightweight gradient boosting machine and DNN for candidate evaluation. The gradient boosting component achieved a MAP@K of 0.06, significantly outperforming the DNN's 0.02, effectively addressing cold-start issues and large-scale personalized recommendation challenges [23]. However, their method remains limited in handling large-scale behavioral sequences and complex feature interactions. In contrast, IFP-DeepFM effectively enhances higher-order feature representations by integrating frequent-pattern features with deep networks. Karabila et al. proposed a recommendation system combining collaborative filtering with sentiment analysis. By fine-tuning a BERT model, they achieved 91% accuracy in sentiment classification and leveraged these insights to optimize product selection, significantly improving recommendation accuracy and personalization [24]. Nevertheless, this approach has limited capacity for mining pure behavioral patterns, whereas IFP-DeepFM extracts high-weight patterns across multiple behavior types—such as browsing, add-to-cart, and purchasing—providing a more comprehensive understanding of user behavior for recommendation purposes.

Despite its strong performance, IFP-DeepFM has several limitations. Prediction accuracy may decrease for extremely sparse new users or new items. The model's computational complexity is relatively high, which could extend training times in resource-constrained environments. Additionally, its cross-platform generalization capability requires further validation. Future work could focus on improving computational efficiency, enhancing cross-platform adaptability, and incorporating additional multimodal user features—such as social interactions and textual reviews—to further improve recommendation accuracy and personalization.

5 Conclusion

This study proposes the IFP-DeepFM model, which demonstrates significant advantages in e-commerce recommendation systems. By combining high-frequency user behavior patterns mined through IFP-growth with DeepFM's deep feature-interaction capabilities, the model effectively enhances recommendation accuracy, Recall, and F1-score, while outperforming traditional collaborative filtering, IFP-growth, DeepFM, and CatBoost baselines in CTR, CVR, and user satisfaction metrics. Ablation studies further confirmed the contribution of frequent-pattern features, the time-window module, and the attention mechanism to overall model performance. Nevertheless, some limitations remain. The model's robustness under extremely sparse data scenarios requires improvement, its cross-platform generalization remains partially unverified, and high computational complexity restricts real-time deployment feasibility. Future work will focus on three directions: (1) optimizing the model architecture and algorithms to reduce computational costs and support low-latency, real-time recommendations; (2) enhancing generalization capability through validation on multi-platform and multi-type datasets; and (3) improving interpretability, for example, via attention-weight visualization or SHAP analysis, to provide data-driven guidance for e-commerce marketing strategies. These enhancements will further enable IFP-DeepFM to contribute to improved user experience and platform economic performance.

Acknowledgment

The Collaborative Education Program between Industry and Education of the Ministry of Education, “Exploring the Path of Cultivating College Students'" Double Innovation "Abilities in New Media Marketing Courses Based on Career Needs Orientation"” (No. 231003230310505)

References

[1] Q. Zhang and Y. Xiong, "Harnessing AI potential in E-Commerce: improving user engagement and sales through deep learning-based product recommendations," *Current Psychology*, vol. 43, no. 38, pp. 30379–30401, Sep. 2024. DOI: 10.1007/s12144-024-06649-3

[2] W. Yang and J. Guo, "Consumers' Purchase Behavior Preference in E-Commerce Platform Based on Data Mining Algorithm," *International Journal of Circuits, Systems and Signal Processing*, vol. 16, no. 1, pp. 603–609, Jan. 2022. DOI: 10.46300/9106.2022.16.75

[3] F. T. Abdul Hussien, A. M. S. Rahma, and H. B. Abdulwahab, "An e-commerce recommendation system based on dynamic analysis of customer behavior," *Sustainability*, vol. 13, no. 19, p. 10786, Sep. 2021. DOI: 10.3390/su131910786

[4] I. Hariyale and M. Raghuwanshi, "Advanced Hybrid Recommender Systems: Enhancing Precision and Addressing Cold-Start Challenges in E-Commerce and Content Streaming," *International Journal of Computing and Digital Systems*, vol. 16, no. 1, pp. 1737–1753, Nov. 2024. DOI: 10.12785/ijcds/1601129

[5] L. Liu, "e-Commerce Personalized Recommendation Based on Machine Learning Technology," *Mobile Information Systems*, vol. 2022, no. 1, p. 1761579, Apr. 2022. DOI: 10.1155/2022/1761579

[6] Y. Gulzar, A. A. Alwan, R. M. Abdullah, et al., "OCA: ordered clustering-based algorithm for E-commerce recommendation system," *Sustainability*, vol. 15, no. 4, p. 2947, Feb. 2023. DOI: 10.3390/su15042947

[7] R. Shrivastava, D. S. Sisodia, and N. K. Nagwani, "Deep ensembled multi-criteria recommendation system for enhancing and personalizing the user experience on e-commerce platforms," *Knowledge and Information Systems*, vol. 66, no. 12, pp. 7799–7836, Sep. 2024. DOI: 10.1007/s10115-024-02187-3

[8] D. R. K. Raja, G. H. Kumar, S. M. Basha, et al., "Recommendations based on integrated matrix time decomposition and clustering optimization," *International Journal of Performability Engineering*, vol. 18, no. 4, p. 298, Mar. 2022. DOI: 10.23940/ijpe.22.04.p8.298306

[9] X. Zhou, Y. Li, and W. Liang, "CNN-RNN based intelligent recommendation for online medical pre-diagnosis support," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 912–921, May 2020. DOI: 10.1109/TCBB.2020.2994780

[10] M. Ma, G. Wang, and T. Fan, "Improved DeepFM recommendation algorithm incorporating deep feature extraction," *Applied Sciences*, vol. 12, no. 23, p. 11992, Nov. 2022. DOI: 10.3390/app122311992

[11] S. Zhou, H. Liu, B. Chen, et al., "Status set sequential pattern mining considering time windows and periodic analysis of patterns," *Entropy*, vol. 23, no. 6, p. 738, Jun. 2021. DOI: 10.3390/e23060738

[12] T. Widiyaningtyas, I. Hidayah, and T. B. Adji, "User profile correlation-based similarity (UPCSim) algorithm in movie recommendation system," *Journal of Big Data*, vol. 8, no. 1, p. 52, Mar. 2021. DOI: 10.1186/s40537-021-00425-x

[13] S. Bhaskaran and R. Marappan, "Design and analysis of an efficient machine learning based hybrid

recommendation system with enhanced density-based spatial clustering for digital e-learning applications," *Complex & Intelligent Systems*, vol. 9, no. 4, pp. 3517–3533, Aug. 2023. DOI: 10.1007/s40747-021-00509-4

[14] C. Cui, J. Qin, and Q. Ren, "Deep Collaborative Recommendation Algorithm Based on Attention Mechanism," *Applied Sciences*, vol. 12, no. 20, p. 10594, Oct. 2022. DOI: 10.3390/app122010594

[15] P. Anitha and M. M. Patil, "RFM model for customer purchase behavior using K-Means algorithm," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 5, pp. 1785–1792, May 2022. DOI: 10.1016/j.jksuci.2019.12.011

[16] A. Shankar, P. Perumal, M. Subramanian, et al., "An intelligent recommendation system in e-commerce using ensemble learning," *Multimedia Tools and Applications*, vol. 83, no. 16, pp. 48521–48537, May 2024. DOI: 10.1007/s11042-023-17415-1

[17] K. Tabianan, S. Velu, and V. Ravi, "K-means clustering approach for intelligent customer segmentation using customer purchase behavior data," *Sustainability*, vol. 14, no. 12, p. 7243, Jun. 2022. DOI: 10.3390/su14127243

[18] M. Shawkat, M. Badawi, S. El-ghamrawy, et al., "An optimized FP-growth algorithm for discovery of association rules," *The Journal of Supercomputing*, vol. 78, no. 4, pp. 5479–5506, Mar. 2022. DOI: 10.1007/s11227-021-04066-y

[19] A. Pirayre, P. Michel, S. S. Rodriguez, et al., "Driving behavior identification and real-world fuel consumption estimation with crowdsensing data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18378–18391, May 2022. DOI: 10.1109/TITS.2022.3169534

[20] H. Guo, R. Tang, Y. Ye, et al., "DeepFM: a factorization-machine based neural network for CTR prediction," *arXiv preprint arXiv:1703.04247*, Mar. 2017. DOI: 10.48550/arXiv.1703.04247

[21] A. Hajjouz and E. Avksentieva, "Evaluating the effectiveness of the CatBoost classifier in distinguishing benign traffic, FTP BruteForce and SSH BruteForce traffic," in *2024 9th International Conference on Signal and Image Processing (ICSIP)*, IEEE, 2024, pp. 351–358. DOI: 10.1109/ICSIP61881.2024.10671552

[22] J. Shi, F. Shang, S. Zhou, et al., "Applications of quantum machine learning in large-scale e-commerce recommendation systems: enhancing efficiency and accuracy," *Journal of Industrial Engineering and Applied Science*, vol. 2, no. 4, pp. 90–103, Sep. 2024. DOI: 10.5281/zenodo.13117899

[23] D. N. Nguyen, V. H. Nguyen, T. Trinh, et al., "A personalized product recommendation model in e-commerce based on retrieval strategy," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 10, no. 2, p. 100303, Jun. 2024. DOI: 10.1016/j.joitmc.2024.100303

[24] I. Karabila, N. Darraz, A. EL-Ansari, et al., "BERT-enhanced sentiment analysis for personalized e-commerce recommendations," *Multimedia Tools and Applications*, vol. 83, no. 19, pp. 56463–56488, Jun. 2024. DOI: 10.1007/s11042-023-17689-5