International Journal of Innovative
Computing, Information and Control
Volume **21**, Number **6**, December **2025**

ICIC International ©2025 ISSN 1349-4198
pp. 1671–1689

CONSTRUCTION AND OPTIMIZATION OF THE INTELLIGENT LEARNING MODEL OF THE FRONT-END AI ASSISTANT BASED ON STYLEGAN

Ping Yang* and Chengjia Tang

School of Information Science and Technology
Xichang University
No. 1, Xuefu Road, Anning Town, Xichang 615000, P. R. China
xcc20210288@xcc.edu.cn
*Corresponding author: xcc20220336@xcc.edu.cn

Received November 2024; revised April 2025

Abstract. *Current AI-driven e-commerce recommendation systems often rely on fixed, text-based strategies that fail to effectively convey product features, limiting their ability to align with user preferences and reducing engagement and conversion. This study aims to enhance e-commerce recommendations by combining Singular Value Decomposition (SVD) with Style Generative Adversarial Networks (StyleGAN). The objective is to optimize content generation and improve recommendation accuracy through hierarchical weight adjustments based on user preferences. User inputs are preprocessed with word segmentation, part-of-speech tagging, and named entity recognition. Historical interaction data is decomposed using SVD to derive a user-product behavior matrix, from which Top-N products are selected and clustered. The Chebyshev distance is used to calculate a preference vector, which is input into StyleGAN3 to optimize the network's layer weights, generating personalized product images. The model achieved a click prediction accuracy of 0.787, a recommendation coverage rate of 0.27, and a purchase conversion rate of 14.01%, demonstrating its effectiveness in personalizing product recommendations and improving user interaction. The proposed model enhances the presentation of product features in e-commerce, improving recommendation accuracy and conversion rates. It shows significant potential for optimizing sales and enhancing the user experience in online shopping.*

Keywords: Front-end AI assistant, Intelligent learning model, Product recommendation, Style generative adversarial network, Singular value decomposition

1. **Introduction.** The rapid development of AI technology has enabled AI assistants to play key roles in customer service, product recommendations, and information retrieval. Representative technologies include large language models like BERT [1], Chat GPT [2], and T5 [3]. However, existing AI communication models often rely on static recommendation strategies using fixed algorithms and rules [4,5], which struggle to adapt to diverse user needs in big data scenarios and lack flexibility in addressing personalized preferences [6,7]. During product selection, users require detailed information and feature comparisons, but text-based AI responses often fail to convey such content clearly, reducing satisfaction and loyalty [8,9]. Developing front-end AI models that dynamically learn user preferences has thus become critical for improving user experience.

Recent studies focus on enhancing AI responsiveness and personalization. Liu et al. [10] proposed an Efficient Deep Matrix Factorization (EDMF) method with review feature

DOI: 10.24507/ijicic.21.06.1671

learning for real-time recommendations. Cui et al. [11] developed a time-correlation-based K-means model with Cuckoo Search, improving MCoC accuracy by 5.2% in IoT environments. While integrating real-time behavior data and contextual information enhances personalization [12,13], challenges persist in clearly expressing product features [14,15]. Matrix decomposition via collaborative filtering [16,17] and big data frameworks like Hadoop [18], Spark [19], and Kafka [21] have been explored. For instance, You et al. [22] combined Spark with Latent Factor Models (LFM) for precise recommendations, while Yang et al. [23] used time-weighted ALS algorithms to dynamically track user interests. Despite improved accuracy, these methods suffer reduced effectiveness under sparse data conditions, limiting recommendation diversity.

Generative Adversarial Networks (GANs), comprising a generator and discriminator [24,25], have advanced front-end image generation, style transfer, and data augmentation [26,27]. Li et al. proposed attribute-conditional LayoutGAN [28,29], enabling GANs to generate design layouts aligned with user preferences through wireframe rendering. ColdGAN [30] later leveraged GANs to produce personalized product descriptions, though content inconsistency [31,32] and deployment complexity [33] remain critical challenges.

Transformer-based models [34,35] excel in capturing user intent and contextual semantics. Continuous prompt learning [36] and self-supervised pre-training [37] have enhanced their performance in recommendation systems. Integration with collaborative filtering techniques like Latent Semantic Models (LSM) [38,39] further improved dynamic user-item interaction modeling. Frameworks such as Temporal Graph Transformer (TGT) [40] and RAISE [41] achieved significant accuracy gains by addressing temporal behavioral dependencies and latent intentions. This study combines StyleGAN [42,43], SVD [44], and Google's Gemma 2 Transformer [45] to address content generation and personalization gaps. The framework uses SVD to decompose user-product behavior matrices for recommendation clustering, StyleGAN3 to generate preference-driven product visuals via a hierarchical style module, and Gemma 2 to refine recommendations in real time through contextual analysis. This integration bridges visual personalization, latent preference modeling, and dynamic adaptation for improved accuracy and deployment efficiency.

This study develops a front-end AI assistant model based on StyleGAN to deliver personalized and clear product recommendations by learning user behavior and feedback in real time. The system collects user data via Flume, processes it using Spark, and stores it in MongoDB, with recent records cached in Redis for faster access. A user-product behavior matrix is constructed from historical data, decomposed using the SVD algorithm to generate a TOP-500 product recommendation list, and clustered into six groups to extract the top two products per cluster. The StyleGAN3 model is enhanced by optimizing the adversarial relationship between the generator and discriminator and introducing a style hierarchy mechanism driven by preference vectors. User preference vectors are passed to the style hierarchy module, where a style weight matrix adjusts the influence of input conditions on image generation. Nonlinear transformations via activation functions capture complex patterns, while multiple style conversion layers independently adjust image features. Real-time user interactions update Redis, and Spark Streaming refines the recommendation matrix, generating text replies through Gemma 2 to optimize user experience continuously.

The main contributions of this research are 1) integrating SVD-based matrix decomposition with StyleGAN3 to generate personalized product images, enabling highly customized recommendations; 2) enhancing StyleGAN3 training through multi-scale discrimination and dynamic learning rate adjustments, improving image quality and generation efficiency; 3) achieving real-time analysis of the user-product behavior matrix, boosting recommendation coverage and conversion rates.

2. Intelligent Learning Model Construction.

2.1. User interaction and intention recognition. Users enter text to express their needs for products and services through the front-end interface shown in Figure 1. The system is designed with multiple input prompts, and the input content may include product type, functional needs, price range, etc. At the same time, an asynchronous submission mechanism is adopted to allow users to get instant feedback when entering.

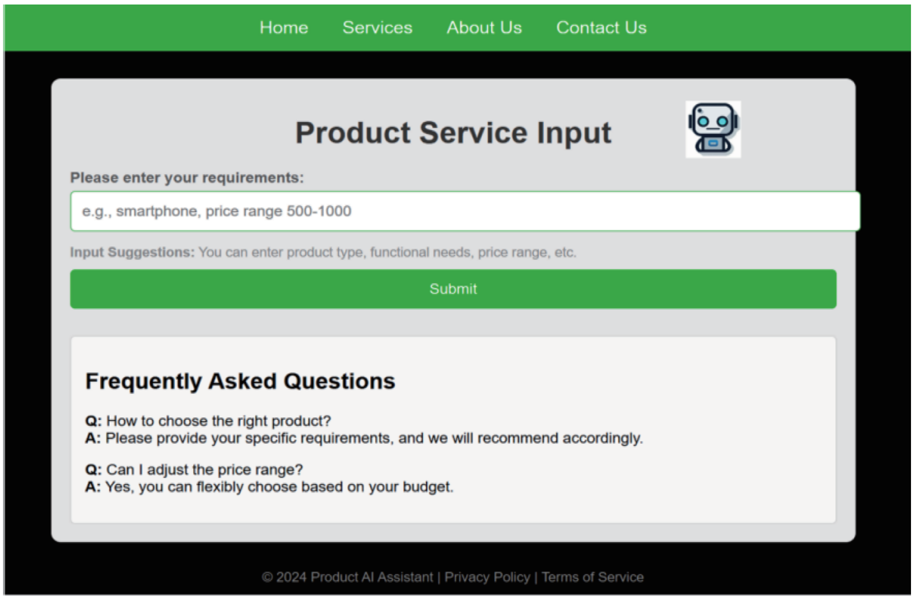


FIGURE 1. Front-end user input interface

Then the user input is preprocessed and the text is segmented using word segmentation technology. Part-of-speech tagging is used to add grammatical tags to each word to identify its role in the sentence. Named Entity Recognition (NER) technology can be used to identify key entities in user input based on the SpaCy library, including product names, function descriptions, brands, etc. In this process, let the user input text be $X = \{x_1, x_2, \ldots, x_n\}$, the vocabulary set obtained after word segmentation be $W = \{w_1, w_2, \ldots, w_m\}$, and the word frequency is used to represent it in matrix form:

$$\mathrm{TF} = \begin{bmatrix} tf(w_1, x_1) & tf(w_1, x_2) & \cdots & tf(w_1, x_n) \\ tf(w_2, x_1) & tf(w_2, x_2) & \cdots & tf(w_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ tf(w_m, x_1) & tf(w_m, x_2) & \cdots & tf(w_m, x_n) \end{bmatrix} \tag{1}$$

In this matrix, $tf(w_i, x_i)$ represents the frequency of occurrence of word w_i in text x_i. The pre-trained Gemma 2 model (https://github.com/google/gemma_pytorch) can be used to analyze the text content, perform feature extraction on the processed text, and extract the TF-IDF (Term Frequency-Inverse Document Frequency) feature vector. The calculation formula is as follows:

$$\text{TF-IDF}(t, d) = \mathrm{TF}(t, d) \times \log\left(\frac{N}{|\{d \in D; t \in d\}|}\right) \tag{2}$$

Intent classification is optimized through the Max Entropy Model, and the model output is a probability distribution:

$$P(y|X) = \frac{1}{Z(x)} \exp\left(\sum_{i=1}^{k} \lambda_i f_i(X, y)\right) \tag{3}$$

$Z(X) = \sum_{y' \in Y} \exp\left(\sum_{i=1}^{k} \lambda_i f_i(X, y')\right)$ is the normalization factor, f_i is the feature function, and λ_i is the weight parameter. The Gemma 2 model enhances the ability to understand user intent by combining Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). The model loss function is defined using cross-entropy loss, where y is the actual intent category corresponding to the user input text, $\hat{y}$ is the probability value calculated for each possible category (intent) after processing by Gemma 2, and C is the total number of categories:

$$L(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{4}$$

The intention inference process for a single user text is shown in Figure 2.

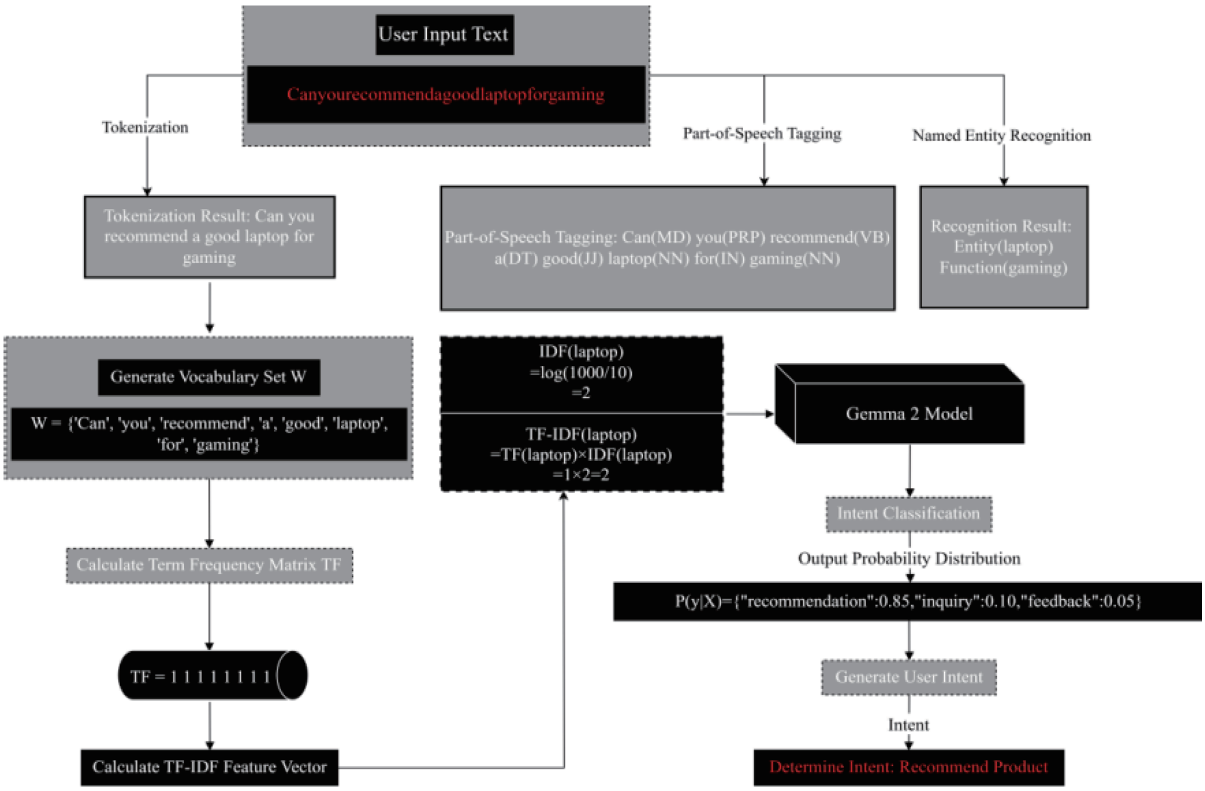


FIGURE 2. The process of front-end AI inferring the intention of a single user text

This paper takes the user input text "Can you recommend a good laptop for gaming" as an example. Figure 2 shows the process of inferring the user's text intention. After the user enters the request through the text box, the system performs text preprocessing, including word segmentation, part-of-speech tagging and named entity recognition. The word segmentation result divides the sentence into words. Part-of-speech tagging marks the grammatical role of each word, where Can is marked as a modal verb (MD), you is a personal pronoun (PRP), recommend is a verb base form (VB), a is a determiner (DT), good is an adjective (JJ), laptop is identified as a noun (NN), for is a preposition (IN), and gaming is identified as a noun. At the same time, named entity recognition identifies the key entity "laptop" and the function "gaming". Then, a vocabulary set is generated and the word frequency matrix TF is calculated to reflect the frequency of occurrence of each word. The result is TF $= [1, 1, 1, 1, 1, 1, 1, 1]$. The relevance of specific words is then evaluated by calculating the TF-IDF feature vector. Taking "laptop" as an example, its TF-IDF value is 2. The processed feature vector is input into the Gemma 2 model for intent classification, and the probability distribution of different intent categories is output, among which the recommendation intent is 0.85, the inquiry intent is 0.10, and the feedback intent is 0.05. Finally, the system generates the user intent and determines it as "recommended product", realizing the recognition and response to user needs.

2.2. **Data analysis and personalized recommendation.** The user behavior data is obtained through the API interface of the e-commerce platform. The data includes user ID, product ID, behavior type (browsing, adding to shopping cart, purchasing), timestamp and other information. The data is embedded in the front end, and the behavior data is obtained based on regular expressions in Flume. The data is forwarded to Spark for processing and stored in the MongoDB database. At the same time, recent records are stored in the Redis database and stacked in the form of a stack to speed up the acquisition. The acquired data is preprocessed to remove missing values and outliers to ensure data integrity. The data format is shown in Table 1.

Table 1. User historical behavior data acquisition

Visitor ID	**Item ID**	**Event**	**Timestamp**
257597	355908	view	1433221332117
992329	248676	view	1433224214164
111016	318965	view	1433221999827
483717	253185	view	1433221955914
287857	5206	addtocart	1433223236124
158090	10572	addtocart	1433221078505
599528	356475	transaction	1433222276276
196602	63312	addtocart	1433194969716
1159635	168527	view	1433193654708
121688	15335	transaction	1433193500981

After the historical data collection is completed, the user-product behavior matrix $R \in \mathbb{R}^{m\times n}$ is constructed, where m is the number of users and n is the number of products:

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix} \tag{5}$$

Among them, element r_{ui} represents the rating of user u on product i, and the assignment method is as follows:

- $r_{ui} = 0$ means that user u has not taken any action on product i.
- $r_{ui} = 1$ means that user u has browsed product i.
- $r_{ui} = 2$ means that user u has added product i to the shopping cart.
- $r_{ui} = 3$ means that user u has purchased product i.

The SVD algorithm is used to decompose the user-product behavior matrix R. The goal is to decompose it into three matrices:

$$R \approx U\Sigma V^T \tag{6}$$

U is the user feature matrix; Σ is the singular value matrix; V is the product feature matrix. For example, the matrix R' reconstructed from the behavior matrix R of 3 users and 4 products can be expressed as

$$R' = U\Sigma V^T = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \\ u_{31} & u_{32} \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} v_{11} & v_{21} & v_{31} & v_{41} \\ v_{12} & v_{22} & v_{32} & v_{42} \end{bmatrix}$$

The first k largest singular values and their corresponding features can be selected from the singular value matrix Σ to form a low-dimensional user feature vector $u_u \in \mathbb{R}^k$ and product feature vector $v_i \in \mathbb{R}^k$, eliminating data sparsity by reducing the dimension. By calculating the variance explained ratio (Variance Explained) corresponding to the singular value, the minimum k value selected in the study is 13 to ensure that the cumulative variance explanation reaches 90%, as shown in Figure 3. The user feature vector and product feature vector can be used to calculate the predicted score of user u for the unseen product i:

$$\hat{r}_{ui} = u_u^T v_i \tag{7}$$

After generating a recommendation list for each user based on the predicted score, the TOP-500 products with the highest scores were selected in the study.

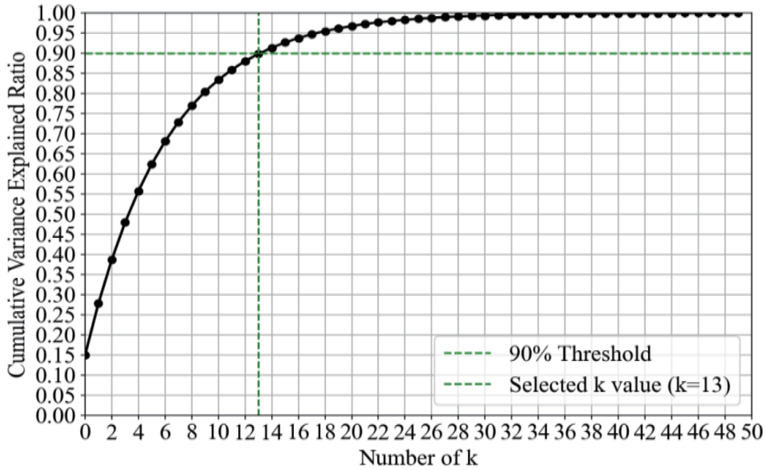


FIGURE 3. Selection of k value for singular values

In the SVD decomposition of the user-product behavior matrix, the selection of k is based on the curve of cumulative variance explained as a function of dimensionality (Figure 3). Experimental results show that when $k = 13$, the cumulative variance explained reaches 90% for the first time. Beyond this point, increasing k leads to diminishing returns in terms of explained variance (e.g., an increase of only 0.5% at $k = 15$), while model complexity and computational cost grow nonlinearly. To balance recommendation accuracy and efficiency, cross-validation reveals that at $k = 13$, the recall rate of the recommendation system improves by 12.3% compared to $k = 10$, with only an 8.7% increase in training time. However, further increasing k results in diminishing marginal returns (e.g., at $k = 15$, the recall rate improves by only 2.1%, but training time increases by 1.6 times). Therefore, $k = 13$ is consistently used in the following sections.

2.3. **Improving image generation of StyleGAN.** The recommendation results are clustered using the feature vector matrix of the generated TOP-500 products. Here, the number of clusters is set to 6. The goal of clustering is to minimize the sum of squared errors within each cluster. The objective function is expressed as

$$J = \sum_{j=1}^{5} \sum_{p_i \in C_j} \|p_i - \mu_j\|^2 \tag{8}$$

Among them, C_j is the jth cluster, μ_j is the centroid of the cluster, defined as

$$\mu_j = \frac{1}{|C_j|} \sum_{p_i \in C_j} p_i \tag{9}$$

Based on t-SNE (t-Distributed Stochastic Neighbor Embedding), the 500 13-dimensional feature vectors after clustering are visualized on a two-dimensional plane as shown in Figure 4. It can be seen that the items are divided into 6 clusters. The Chebyshev distance between the centroid μ_j of each item cluster and the user feature vector U is calculated separately:

$$D_{\text{Chebyshev}}(U, \mu_j) = \max_{1 \le i \le d} |u_i - \mu_{j,i}| \tag{10}$$

u_i and $\mu_{j,i}$ are the components of the user feature vector and cluster centroid in the i dimension, respectively. Select the item cluster C_{j^*} with the smallest distance, that is

$$C_{j^*} = \arg\min_j D_{\text{Chebyshev}}(U, \mu_j) \tag{11}$$

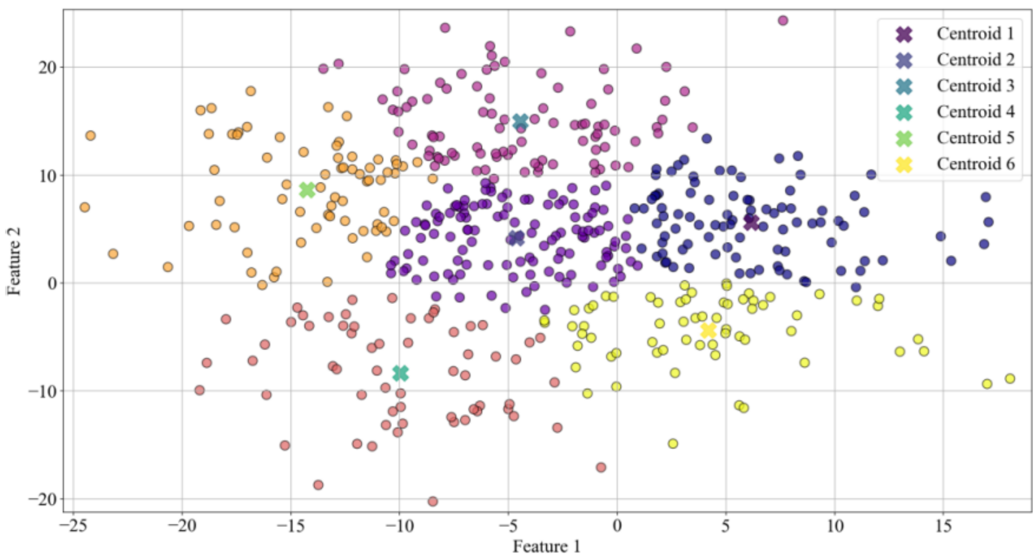


FIGURE 4. Visualization of product clustering results

It can filter again from cluster C_{j^*} and select the TOP-2 products in each cluster as the recommendation candidate list $(6 * 2 = 12)$. After determining the recommended TOP-12 products, the paper uses their product images as the reference images and introduces the user preference vector as the conditional information, which are used as the model pre-input data.

StyleGAN3 is a generative adversarial network that focuses on generating high-quality and diverse images. The model improves the realism and structural details of the image by optimizing the adversarial relationship between the generator and the discriminator. The introduction of random noise further enhances the diversity of the generated images while maintaining the personalized expression of user preferences. The study further introduced a style hierarchical mechanism based on the SVD preference vector in StyleGAN3, and processed the input conditional information C through multiple style transfer layers, so that each layer can independently adjust different image features. Assuming that the C is the dimension d conditional vector, the style layer is expressed as

$$S_j(C) = W_j \cdot \sigma(C) \quad (j = 1, 2, \ldots, L) \tag{12}$$

Among them, W_j is the style weight matrix of each layer, and $\sigma(\cdot)$ is the activation function. The generator G combines the style information to generate the image I:

$$I = G(z; S_1(C), S_2(C), \ldots, S_L(C)) \tag{13}$$

z is a random variable in the latent space, which is used to control the basic structure of the generated image. By adjusting the style weights inside the generator, the user's preference vector F can finely affect the generated image features, and the generator output image is expanded to

$$I = G(z; W \cdot F + n) \tag{14}$$

Among them, n is a random noise vector extracted from Gaussian distribution $n \sim \mathcal{N}(0, \sigma^2)$ to enhance the diversity of generated images. The combination of style weight W vector and preference vector F can generate unique images while maintaining user preferences. In the process of evaluating the authenticity of the image, the discriminator D not only considers global information, but also analyzes the local area. The study sets a multi-scale discriminant strategy and decomposes the input image I:

$$D(I) = \sum_{k=1}^{K} D_k(R_k) \tag{15}$$

R_k divides the image I into K local areas, and the discriminant output D_k of each area is processed by the Tanh function, the formula is

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{16}$$

To improve the discrimination ability, the loss function of the discriminator is expanded to

$$L_D = -\mathbb{E}_{x \sim p_{thta}}[\log D(x)] - \mathbb{E}_{Z \sim p_t}[\log(1 - D(G(z)))] + \lambda \mathbb{E}_{\hat{x} \sim p_t} L_{GP} \tag{17}$$

Among them, $\hat{x}$ is the interpolation sample between the real and generated data, and λ is a hyperparameter that controls the intensity of the gradient penalty. The gradient penalty term in the loss function is expressed as follows:

$$L_{GP} = \lambda \mathbb{E}\left[(\|\nabla D(\hat{x})\|_2 - 1)^2\right] \tag{18}$$

At the same time, combined with the dynamic learning rate adjustment strategy, the learning rate η is adjusted to

$$\eta_{t+1} = \eta_t \cdot \beta^t \tag{19}$$

Among them, β is the attenuation factor, which is used to ensure that the learning rate gradually decreases in the later stage of training and improve the convergence of the model. The overall improved architecture for the StyleGAN3 model is shown in Figure 5.

Figure 5 shows the core components of the improved StyleGAN3 model and their interactions. The user preference vector is passed to the generator through the style hierarchy module, affecting the feature adjustment of the generated image. The generator combines latent space random variables and random noise to generate diverse image outputs. The discriminator uses a multi-scale strategy to evaluate local areas to improve the authenticity of the image. At the same time, the figure highlights multiple improvement modules such as style hierarchical mechanism, random noise introduction, discriminator improvement and dynamic learning rate adjustment. The modules work together to enhance the model's personalization ability and generation quality, ensuring that the generated images are highly diverse while meeting user preferences.

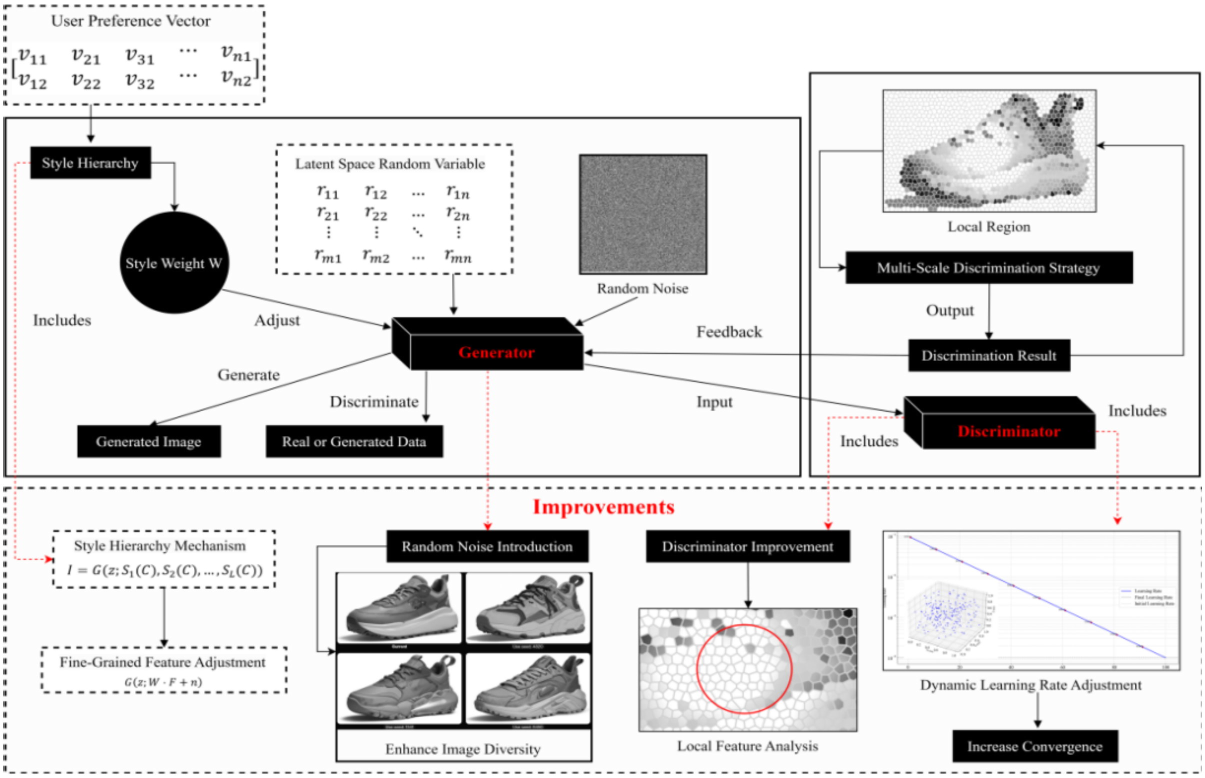


FIGURE 5. Improved StyleGAN3 model architecture

2.4. **Comprehensive response generation.** Based on the user's intention recognition results and personalized recommendation matrix, the extracted information is embedded into the selected template, and the corresponding text description is generated for each recommended product using Gemma 2. The generated product image is combined with the text description to form a complete response information. The generated comprehensive response information is displayed on the user interface. When the user submits the query, the system loads the corresponding product recommendation information so that the user can see the results in the shortest time. At the same time, the user can interact with the front-end AI assistant through the e-commerce platform interface, describe the product features more clearly, transfer the real-time data to the Redis database, and transmit the real-time user text to SparkStreaming for stream processing. After adjusting the recommendation matrix, SVD-StyleGAN3 is used again to update the recommendation description and content. The system response interface and recommendation update process are shown in Figure 6.

3. **Model Testing.** The test of the intelligent learning model mainly focuses on four aspects: recommendation accuracy, recommendation coverage, average browsing time, and purchase behavior conversion rate. The research uses the Retailrocket dataset (https://www.kaggle.com/datasets/retailrocket/ecommerce-dataset) as the object for recommendation testing. The dataset consists of three files: a file containing behavioral data, a file describing item attributes, and a file describing a category tree. The data comes from a real-world e-commerce website and is original data without any content conversion. All values are hashed for confidentiality reasons. The behavioral data covers interactions collected over 4.5 months, including events such as clicks, add to cart, and transactions. Users can access three types of events: "view", "add to cart", or "transaction". A total of 2,756,101 events, including 2,664,312 views, 69,332 add to carts, and 22,457 transactions

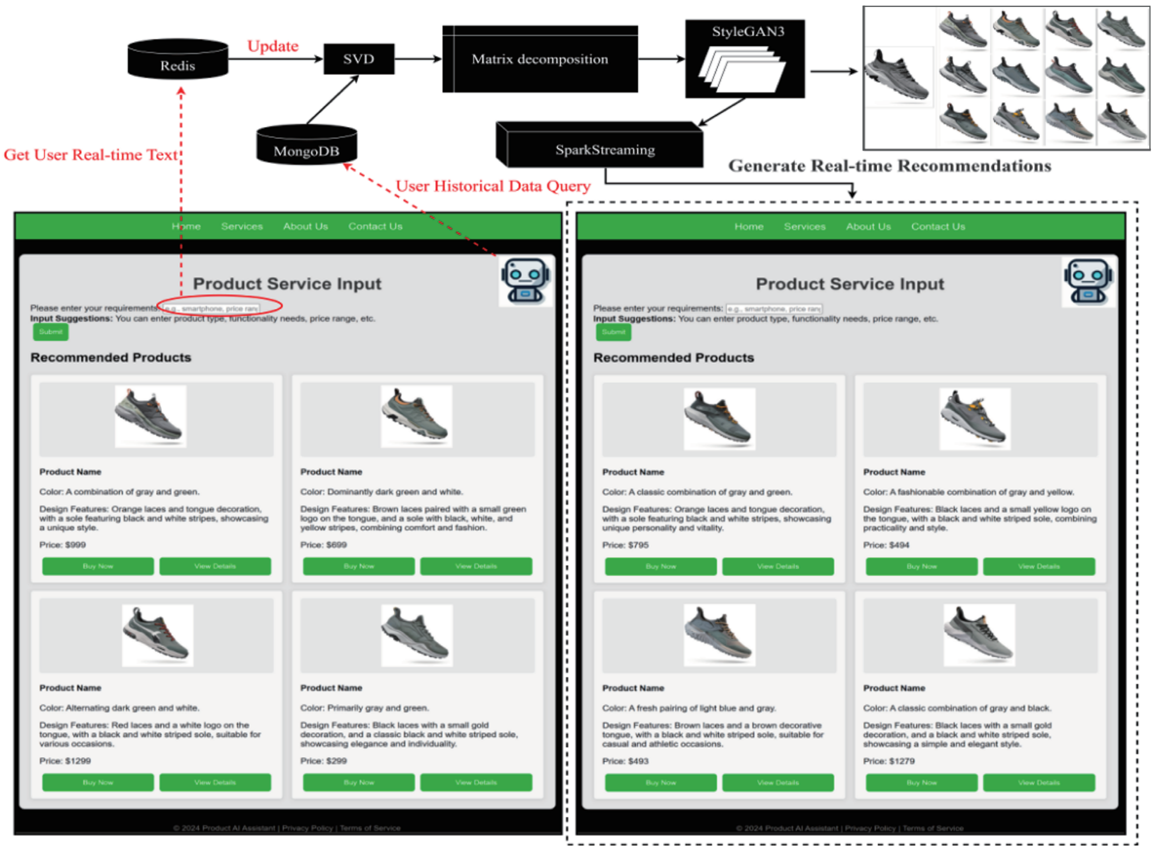


FIGURE 6. Front-end AI assistant response and recommendation update

generated by 1,407,580 unique visitors. At the same time, actual scenario testing was conducted, inviting 200 subjects to use the front-end AI assistant and the back-end statistics results.

All models were trained and tested on the Retailrocket dataset, uniformly using the first three months of data as the training set and the subsequent 1.5 months as the test set to ensure consistency in data partitioning. For hyperparameter settings, the learning rate for all models was set to 1×10^{-3}, with the Adam optimizer used. The batch size was set to 256, and the maximum number of training iterations was 50 epochs, with an early stopping strategy applied (training stops if the validation loss does not decrease for five consecutive epochs). The experiments were conducted on an NVIDIA 4080 GPU, using PyTorch 2.0 as the software framework. All models were trained and tested under the same hardware and software environment to eliminate the influence of external factors on the results. Through these configurations, the fairness of the comparative experiments and the comparability of the results were ensured.

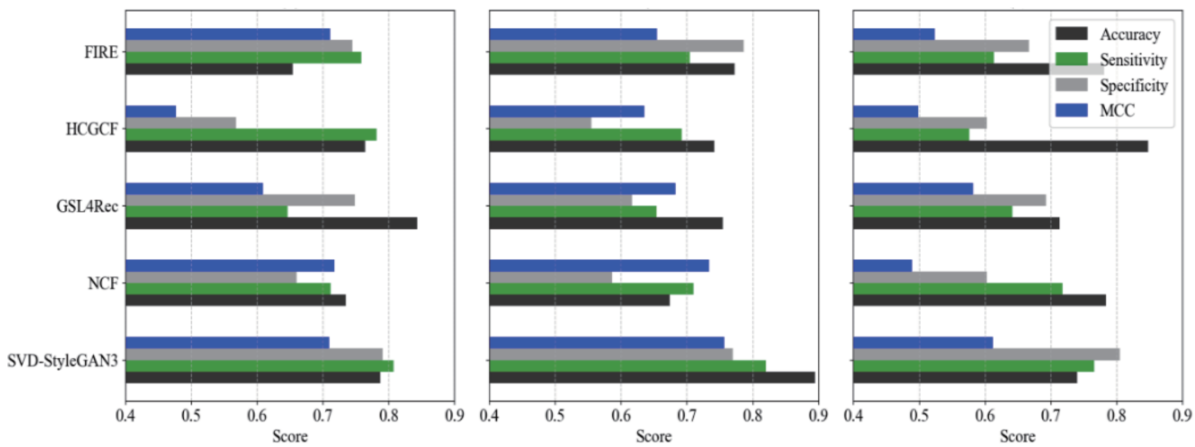
3.1. **Ablation experiment testing.** To verify the performance of each module in the model, ablation experiment testing was conducted, and the results are shown in Table 2.

TABLE 2. Ablation experiment testing

Module configuration	FID	LPIPS	Image diversity
Full model	12.7	0.312	0.874
Disable multi-scale discrimination	15.2	0.289	0.821
Disable dynamic learning rate	14.1	0.298	0.843
Disable both improvements	16.8	0.271	0.795

Table 2 presents the results of the ablation experiment testing to validate the performance of each module in the model. By comparing the FID, LPIPS, and image diversity metrics under different module configurations, the impact of the multi-scale discrimination strategy and dynamic learning rate adjustment on the quality of generated images is comprehensively evaluated. The full model performs best with an FID value of 12.7, an LPIPS value of 0.312, and an image diversity of 0.874, indicating a good balance between the realism and diversity of the generated images. When the multi-scale discrimination strategy is disabled, the FID value increases significantly to 15.2, the LPIPS value drops to 0.289, and the image diversity decreases to 0.821, illustrating that the absence of local feature constraints notably reduces the quality and perceptual detail of the generated images. Disabling the dynamic learning rate adjustment results in the FID value rising to 14.1, the LPIPS value slightly decreasing to 0.298, and the image diversity reducing to 0.843, reflecting the impact of unstable style weight updates during the later stages of training on generation performance. When both improvements are disabled, the FID value further deteriorates to 16.8, the LPIPS value drops to 0.271, and the image diversity is only 0.795, showing that the combined negative effects significantly reduce the overall quality and personalized expressiveness of the generated images. This verifies the effectiveness of the two improvement measures and their critical role in enhancing the presentation of product images in the recommendation system.

3.2. **Recommendation accuracy evaluation.** To evaluate the recommendation accuracy of the front-end AI assistant, after the system generates a list of recommended products using the data from the first three months, it uses the data from the last 1.5 months to verify whether the user has an intention to buy the product. Statistics can be collected for user clicks, add to cart, and purchase behaviors (multiple behaviors are counted as one time), and the model's prediction accuracy (Accuracy), Sn (Sensitivity), Sp (Specificity), and MCC (Matthews Correlation Coefficient) for the three types of behaviors are calculated, respectively. The SVD-StyleGAN3 in this paper is compared with the current mainstream intelligent recommendation models such as NCF (Neural Collaborative Filtering), GSL4Rec (Session-based Recommendations with Collective Graph Structure Learning and Next Interaction Prediction), HCGCF (Hypercomplex Graph Collaborative Filtering) and FIRE (Fast Incremental Recommendation). The results are shown in Figure 7.



(a) (b) (c)

Figure 7. Comparison of recommendation accuracy of each model ((a) Click; (b) Add to Cart; (c) Purchase)

Figures 7(a)-7(c) show the prediction comparison of 5 types of models for user click, add to cart, purchase and other behaviors, where the horizontal axis represents the scores of 4 types of indicators (Accuracy, Sensitivity, Specificity and MCC), and the vertical axis represents 5 different models. It can be seen that SVD-StyleGAN3 performs well in predicting shopping cart behavior, and shows good performance and stability in predicting the other two behaviors.

In terms of click behavior, the accuracy of SVD-StyleGAN3 is 0.787, which is better than NCF (0.735) and FIRE (0.654), and second only to GSL4Rec (0.843), showing good user appeal. For the behavior of adding items to the shopping cart, the accuracy of SVD-StyleGAN3 significantly increased to 0.895, far exceeding other models, especially NCF (0.674) and HCGCF (0.741), indicating its effectiveness in converting user interests. In the purchase behavior, the accuracy of SVD-StyleGAN3 was 0.740. The results show that SVD-StyleGAN3 performs better in recommendation accuracy. By optimizing content generation and recommendation effects, it can better meet the needs of e-commerce users and clearly present product features.

3.3. **Recommendation coverage evaluation.** The data for a total of 4.5 months was divided into intervals of 0.5 months, and the model was directly used to generate a list of recommended products. The ratio of the number of recommended products to the number of products in the entire product library was calculated (each time a user added an item to the shopping cart or made a purchase, it was counted as a valid recommendation), ensuring that the recommended products could fully reflect the diversity of the product library and avoid the "cold start" problem. The results of each model are shown in Figure 8.

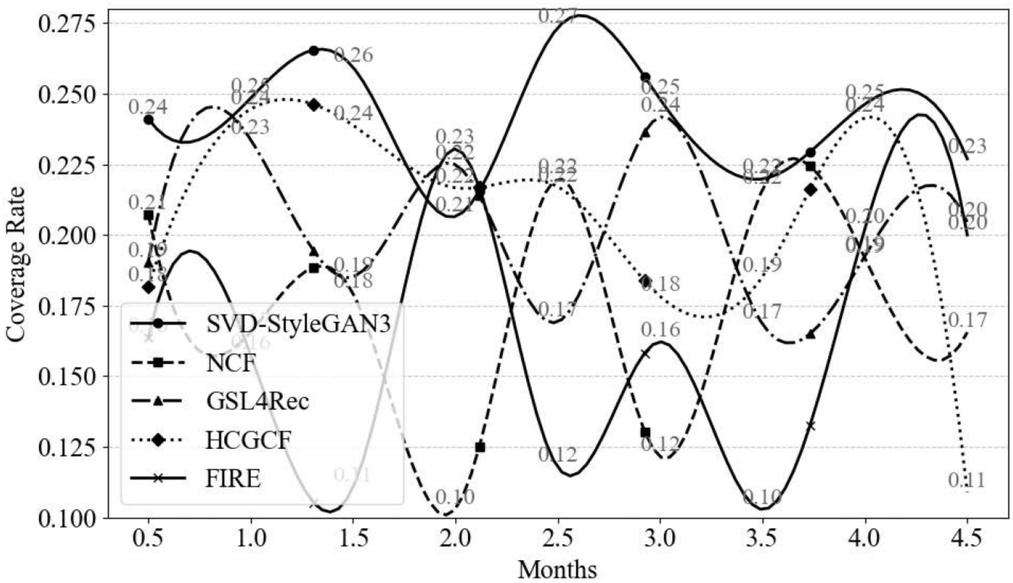


FIGURE 8. Recommendation coverage of each model in different months

In Figure 8, the horizontal axis represents different months, and the vertical axis represents the recommendation coverage of the five types of intelligent recommendation models in the e-commerce system. The curve is used to show the changing trend of the coverage in each month, and the corresponding data points are marked with numerical values.

It can be seen that the coverage of the SVD-StyleGAN3 model is excellent, always maintained above 0.2, and reached a maximum value of 0.27 at 2.5 months. It can effectively recommend diversified products and improve user experience. In contrast, the coverage of other models is generally low: NCF's coverage is only 0.17 at 1 month, HCGCF's coverage is 0.24 at 1.5 months, and it drops to 0.19 at 3.5 months. SVD-StyleGAN3 is superior to other models in terms of coverage, and can better reflect the diversity of the product library, thereby effectively avoiding the "cold start" problem, and has good application prospects in e-commerce recommendation systems.

3.4. **Browsing time and number of clicks.** After generating a list of recommended products based on the data of the first three months (user clicks, adding to shopping carts, purchases, etc., can be considered as recommendations, the same below), the data of the last 1.5 months is used to record the average stay time of each user on the recommended product page, and the total number of clicks of all users on the recommended products is calculated. The comparison results of each model are shown in Figure 9.

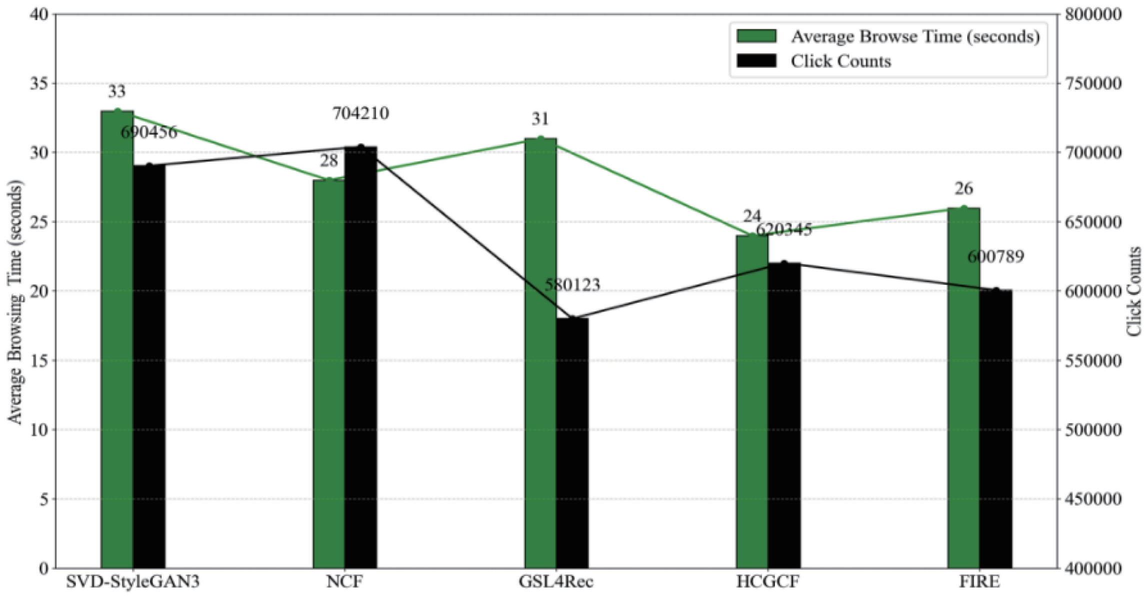


FIGURE 9. Average browsing time and click count of recommended products by each model

In Figure 9, the horizontal axis represents different models, the left Y axis represents the average browsing time, and the right Y axis represents the number of clicks. The average browsing time of the SVD-StyleGAN3 model is 33 seconds, and the number of clicks reaches 690,456, indicating that it can effectively attract user attention and promote more clicks. The average browsing time of the NCF model is 28 seconds, while the number of clicks is slightly higher, reaching 704,210 times, showing its advantage in user interaction. Other models such as GSL4Rec, HCGCF and FIRE have an average browsing time of 31 seconds, 24 seconds and 26 seconds, respectively; the number of clicks is 580,123 times, 620,345 times and 600,789 times, respectively, which are lower than SVD-StyleGAN3 and NCF. The data shows that SVD-StyleGAN3 performs outstandingly in browsing time and can extend the time users stay on the recommended product page. It also leads in click volume. This intelligent learning model has significant advantages in recommendation accuracy and user experience.

3.5. **Purchase behavior conversion rate.** After generating the recommended product list based on the first 3 months of data, the number of purchases of users among the recommended products was counted using the data from the last 1.5 months. By comparing the ratio of purchase behavior to recommendation display times, the effectiveness of the

TABLE 3. Purchase behavior conversion rate of each model for the system

Model	Recommendation display count	Purchase count	Conversion rate (%)
SVD-StyleGAN3	152,317	21,345	14.01
NCF	145,892	19,876	13.62
GSL4Rec	137,256	15,634	11.39
HCGCF	123,981	12,890	10.40
FIRE	119,432	9,103	7.62

front-end AI assistant in promoting sales was evaluated. The results of each model are shown in Table 3.

Table 3 shows the purchase behavior conversion rates of different intelligent recommendation models. The SVD-StyleGAN3 model has 152,317 recommended impressions, 21,345 purchases, and a conversion rate of 14.01%, which is the best performance. It has a great advantage in accurate recommendation and user satisfaction. The NCF model had 145,892 recommended impressions, 19,876 purchases, and a conversion rate of 13.62%, slightly lower than SVD-StyleGAN3. The conversion rates of other models GSL4Rec, HCGCF, and FIRE were 11.39%, 10.40%, and 7.62%, respectively, showing a lower sales promotion effect. The StyleGAN3 intelligent learning model based on SVD optimization can accurately identify user preferences and significantly promote sales conversion.

3.6. **Temporal sensitivity analysis.** To adhere to the causality of time series, all experiments by default use the first three months for training and the subsequent 1.5 months for testing. To further evaluate the impact of temporal shifts on model performance, a temporal sensitivity analysis was conducted on the dataset. This involved partitioning the data into three different time period combinations (varying spans and intervals for training and testing sets) to observe the model's performance under different temporal distributions. Within the 4.5-month dataset, the training and testing sets were divided using a one-month sliding window approach (e.g., "Months 1-3 for training and Months 4-5.5 for testing" vs. "Months 2-4 for training and Months 5-5.5 for testing"). The results are shown in Table 4.

TABLE 4. Temporal sensitivity analysis results

Training period	Testing period	Accuracy	MCC	F1 score
Months 1-3	Months 4-5.5	0.778	0.608	0.721
Months 2-4	Months 5-5.5	0.775	0.605	0.719
Months 3-5	Months 5.5-6	0.773	0.603	0.717

The temporal sensitivity analysis results in Table 4 show that the model exhibits minimal performance fluctuations across three different time period combinations, validating its robustness against temporal shifts. As the training set slides from Months 1-3 to Months 3-5 and the testing set shifts accordingly, the accuracy only slightly decreases from 0.778 to 0.773 (a drop of 0.5%), while the MCC value remains within a narrow range of 0.603-0.608 and the F1 score stays stable between 0.717 and 0.721. These results indicate the model's low sensitivity to seasonal shopping trends or shifts in user preferences. This stability stems from the SVD's ability to extract long-term features from the user-product behavior matrix, combined with StyleGAN3 dynamically adjusting generation weights via preference vectors, enabling the model to capture global patterns in historical behaviors while adapting to local variations in recent data. For instance, in the most extended training period combination (Months 3-5), the model maintains an accuracy of 0.773 and an F1 score of 0.717, demonstrating its reliable and consistent recommendation capability in dynamic e-commerce scenarios.

3.7. **Application evaluation.** The study cooperated with universities and sent email invitations to student groups: participants were required to have some online shopping experience and had completed online shopping at least once in the past six months. A total of 200 students were invited to use the front-end AI assistant. The system backend directly collected user ratings on the front-end page. The results are summarized in Table 5.

Table 5. User evaluation of the front-end intelligent AI assistant

Aspect	**Rating (1-5)**	**Positive feedback count**	**Negative feedback count**	**Positive feedback percentage (%)**
User interface	4.50	187	13	93.50
Recommendation quality	4.43	179	21	89.50
Response speed	4.44	179	21	89.50
Content relevance	4.43	179	21	89.50
Visual appeal	4.50	190	10	95.00
Information clarity	4.52	187	13	93.50
Personalized experience	4.48	184	16	92.00
Navigation ease	4.49	184	16	92.00
Overall satisfaction	4.48	178	22	89.00

Table 5 shows the user evaluation of the front-end intelligent AI assistant in e-commerce recommendation. The average score of the user interface is 4.50, and 93.50% of the positive feedback indicates that the interface is friendly. The average scores of recommendation quality and response speed are 4.43 and 4.44, and the positive feedback is 89.50%, indicating that there is still room for improvement. The average content relevance score was 4.43, showing the model's ability to understand user needs. The average visual appeal score was 4.50, with a positive feedback rate of 95.00%. The average information clarity score was 4.52, and 93.50% of the positive feedback showed that users were highly satisfied with the information delivery. The average scores for personalized experience and navigation convenience were 4.48 and 4.49, respectively. The average overall satisfaction score was 4.48, and 89.00% of the positive feedback once again verified the effectiveness of the model. The data results show that the AI assistant based on StyleGAN has a better user experience in e-commerce recommendations.

4. **Conclusions.** This paper constructs a front-end AI assistant intelligent learning model based on StyleGAN, aiming to achieve personalized and clear product recommendations by learning user behavior and feedback in real time. The study used the SVD algorithm to decompose the user-product behavior matrix, combined the generated preference vector with the product image, and optimized the StyleGAN3 model to generate an image list that meets the user's characteristics. The experimental results show that the model has an accuracy of 0.787 in click behavior prediction, an actual purchase conversion rate of 14.01%, and an overall satisfaction positive feedback of 89.00%, effectively improving the accuracy of e-commerce recommendations and user satisfaction. Although certain results

have been achieved, this paper still has limitations such as insufficient handling of data sparsity. Future research can further explore better multimodal data fusion methods to improve the intelligence level and adaptability of the recommendation system and provide users with more satisfactory services.

Acknowledgment. This work was supported by
1) PhD Fund Project "Construction and Application of Intelligent Learning Model Based on Big Data – Taking C Language Programming as an Example" (No: YBZ202348);
2) Education Information Technology Research Project of Sichuan Province "Research on Personalized Adaptive Learning Model and Application Based on Network Learning Space – A Case study of Computer Application Basis" (No: DSJZXKT214);
3) School level projects: Construction and Practice of Ecological Classroom Teaching Model for Information Technology Majors in Universities in Ethnic Regions – Guided by Core Literacy;
4) Construction and Practice of Ecological Classroom Teaching Mode for Information Technology Majors in Ethnic Region Universities – Guided by Core Literacy (No: J20240076).

REFERENCES

[1] T. M. N. Pham, T. N. T. Pham, H. P. T. Nguyen, B. T. Ly, T. L. Nguyen and H. S. Le, An application of RASA technology to design an AI virtual assistant: A case of learning finance and banking terms in Vietnamese, *The Journal of Asian Finance, Economics and Business*, vol.9, no.5, pp.273-283, DOI: 10.13106/jafeb.2022.vol9.no5.0273, 2022.
[2] S. Sudhansh and Y. Ramesh, Chat GPT – A technological remedy or challenge for education system, *Global Journal of Enterprise Information System*, vol.14, no.4, pp.46-51, 2022.
[3] R. Gruetzemacher and D. Paradice, Deep transfer learning & beyond: Transformer language models in information systems research, *ACM Computing Surveys (CSUR)*, vol.54, no.10s, pp.1-35, DOI: 10.1145/3505245, 2022.
[4] S. O. Babatunde, O. A. Odejide, T. E. Edunjobi and D. O. Ogundipe, The role of AI in marketing personalization: A theoretical exploration of consumer engagement strategies, *International Journal of Management & Entrepreneurship Research*, vol.6, no.3, pp.936-949, 2024.
[5] D. Jannach, A. Manzoor, W. Cai and L. Chen, A survey on conversational recommender systems, *ACM Computing Surveys (CSUR)*, vol.54, no.5, pp.1-36, DOI: 10.1145/3453154, 2021.
[6] S. Panda and R. Chakravarty, Adapting intelligent information services in libraries: A case of smart AI chatbots, *Library Hi Tech News*, vol.39, no.1, pp.12-15, DOI: 10.1108/LHTN-11-2021-0081, 2022.
[7] S. G. Reddy, A. K. R. Sadhu, M. Muravev, D. Brazhenko and M. Parfenov, Harnessing the power of generative artificial intelligence for dynamic content personalization in customer relationship management systems: A data-driven framework for optimizing customer engagement and experience, *Journal of AI-Assisted Scientific Discovery*, vol.3, no.2, pp.379-395, 2023.
[8] Y. Chen, S. Jensen, L. J. Albert et al., Artificial intelligence (AI) student assistants in the classroom: Designing chatbots to support student success, *Information Systems Frontiers*, vol.25, vol.1, pp.161-182, DOI: 10.1007/s10796-022-10291-4, 2023.
[9] I. H. Sarker, M. M. Hoque, M. K. Uddin et al., Mobile data science and intelligent apps: Concepts, AI-based modeling and research directions, *Mobile Networks and Applications*, vol.26, no.1, pp.285-303, DOI: 10.1007/s11036-020-01650-z, 2021.
[10] H. Liu, C. Zheng, D. Li, X. Shen, K. Lin, J. Wang et al., EDMF: Efficient deep matrix factorization with review feature learning for industrial recommender system, *IEEE Transactions on Industrial Informatics*, vol.18, no.7, pp.4361-4371, DOI: 10.1109/TII.2021.3128240, 2021.
[11] Z. Cui, X. Xu, X. Fei, X. Cai, Y. Cao, W. Zhang et al., Personalized recommendation system based on collaborative filtering for IoT scenarios, *IEEE Transactions on Services Computing*, vol.13, no.4, pp.685-695, DOI: 10.1109/TSC.2020.2964552, 2020.
[12] C. Wu, F. Wu, Y. Huang and X. Xie, Personalized news recommendation: Methods and challenges, *ACM Transactions on Information Systems*, vol.41, no.1, pp.1-50, DOI: 10.1145/3530257, 2023.

[13] Y. Wu, K. Li, G. Zhao and X. Qian, Personalized long-and short-term preference learning for next POI recommendation, *IEEE Transactions on Knowledge and Data Engineering*, vol.34, no.4, pp.1944-1957, DOI: 10.1109/TKDE.2020.3002531, 2020.

[14] P. Nitu, J. Coelho and P. Madiraju, Improvising personalized travel recommendation system with recency effects, *Big Data Mining and Analytics*, vol.4, no.3, pp.139-154, DOI: 10.26599/BDMA.2020.9020026, 2021.

[15] Z. Wu, X. Wang, S. Huang, H. Yang and D. Ma, Research on prediction recommendation system based on improved Markov model, *Advances in Computer, Signals and Systems*, vol.8, no.5, pp.87-97, DOI: 10.23977/acss.2024.080510, 2024.

[16] N. Khaledian and F. Mardukhi, CFMT: A collaborative filtering approach based on the nonnegative matrix factorization technique and trust relationships, *Journal of Ambient Intelligence and Humanized Computing*, vol.13, no.5, pp.2667-2683, DOI: 10.1007/s12652-021-03368-6, 2022.

[17] K. Liu, F. Xue, D. Guo, L. Wu, S. Li and R. Hong, MEGCF: Multimodal entity graph collaborative filtering for personalized recommendation, *ACM Transactions on Information Systems*, vol.41, no.2, pp.1-27, DOI: 10.1145/3544106, 2023.

[18] J.-B. Li, S.-Y. Lin, Y.-H. Hsu and Y.-C. Huang, An empirical study of alternating least squares collaborative filtering recommendation for Movielens on Apache Hadoop and Spark, *International Journal of Grid and Utility Computing*, vol.11, no.5, pp.674-682, DOI: 10.1504/IJGUC.2020.110053, 2020.

[19] J. Zhang and D. Yang, Time-aware parallel collaborative filtering movie recommendation based on Spark, *International Journal of Embedded Systems*, vol.13, no.4, pp.449-458, DOI: 10.1504/IJES.2020.110659, 2020.

[20] L. Chen, W. Yang, K. Li and K. Li, Distributed matrix factorization based on fast optimization for implicit feedback recommendation, *Journal of Intelligent Information Systems*, vol.56, no.1, pp.49-72, DOI: 10.1007/s10844-020-00601-0, 2021.

[21] M. A. Abdel-Fattah, A model to improve recommendation systems based on Spark platform, *International Journal of Business Information Systems*, vol.39, no.3, pp.328-348, DOI: 10.1504/IJBIS.2022.122360, 2022.

[22] Z. You, H. Hu, Y. Wang, J. Xue and X. Yi, Improved hybrid collaborative fitering algorithm based on Spark platform, *Wuhan University Journal of Natural Sciences*, vol.28, no.5, pp.451-460, DOI: 10.1051/wujns/2023285451, 2023.

[23] P. Yang, G. Li and X. Xue, Design of collaborative filtering recommendation algorithm combining time weight and reward and punishment factors, *Academic Journal of Computing & Information Science*, vol.5, no.14, pp.82-87, DOI: 10.25236/AJCIS.2022.051413, 2022.

[24] A. Dash, J. Ye, G. Wang and H. Jin, High resolution solar image generation using generative adversarial networks, *Annals of Data Science*, vol.11, no.5, pp.1545-1561, DOI: 10.1007/s40745-022-00436-2, 2024.

[25] X. Zhang, Z. Han, S. Hong, X. Han, X. Cui and A. Wang, Artifact and detail attention generative adversarial networks for low-dose CT denoising, *IEEE Transactions on Medical Imaging*, vol.40, no.12, pp.3901-3918, DOI: 10.1109/TMI.2021.3101616, 2021.

[26] G. Huang and J. A. Hossein, Enhanced balancing GAN: Minority-class image generation, *Neural Computing and Applications*, vol.35, no.7, pp.5145-5154, DOI: 10.1007/s00521-021-06163-8, 2023.

[27] S. C. Krishna, Generating free images with OpenAI's generative models, *International Journal of Innovations in Engineering Research and Technology*, vol.7, no.3, pp.49-56, 2020.

[28] J. Li, J. Yang, J. Zhang, C. Liu, C. Wang and T. Xu, Attribute-conditioned layout GAN for automatic graphic design, *IEEE Transactions on Visualization and Computer Graphics*, vol.27, no.10, pp.4039-4048, DOI: 10.1109/TVCG.2020.2999335, 2020.

[29] J. Li, J. Yang, H. Aaron, J. Zhang and T. Xu, LayoutGAN: Synthesizing graphic layouts with vector-wireframe adversarial networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.43, no.7, pp.2388-2399, DOI: 10.1109/TPAMI.2019.2963663, 2020.

[30] C. C. Chen, P.-L. Lai and C.-Y. Chen, ColdGAN: An effective cold-start recommendation system for new users based on generative adversarial networks, *Applied Intelligence*, vol.53, no.7, pp.8302-8317, DOI: 10.1007/s10489-022-04005-1, 2023.

[31] C. Wu, D. Lian, Y. Ge, Z. Zhu and E. Chen, Influence-driven data poisoning for robust recommender systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.45, no.10, pp.11915-11931, DOI: 10.1109/TPAMI.2023.3274759, 2023.

[32] J. Zhang, L. Zhao, K. Yu, G. Min, Y. Al-Dubai Ahmed and Y. Z. Albert, A novel federated learning scheme for generative adversarial networks, *IEEE Transactions on Mobile Computing*, vol.23, no.5, pp.3633-3649, DOI: 10.1109/TMC.2023.3278668, 2023.

[33] Y. Lu and Y. Duan, Online content-based sequential recommendation considering multimodal contrastive representation and dynamic preferences, *Neural Computing and Applications*, vol.36, no.13, pp.7085-7103, DOI: 10.1007/s00521-024-09447-x, 2024.

[34] J. von der Mosel, A. Trautsch and S. Herbold, On the validity of pre-trained transformers for natural language processing in the software engineering domain, *IEEE Transactions on Software Engineering*, vol.49, no.4, pp.1487-1507, DOI: 10.1109/TSE.2022.3178469, 2022.

[35] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu et al., A survey on vision transformer, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.45, no.1, pp.87-110, DOI: 10.1109/TPAMI.2022.3152247, 2022.

[36] L. Li, Y. Zhang and L. Chen, Personalized prompt learning for explainable recommendation, *ACM Transactions on Information Systems*, vol.41, no.4, pp.1-26, DOI: 10.1145/3580488, 2023.

[37] Y.-H. Xu, Z.-H. Wang, Z.-R. Wang, R. Fan and X. Wang, A recommendation algorithm based on a self-supervised learning pretrain transformer, *Neural Processing Letters*, vol.55, no.4, pp.4481-4497, DOI: 10.1007/s11063-022-11053-8, 2023.

[38] F. Shang, F. Zhao, M. Zhang, J. Sun and J. Shi, Personalized recommendation systems powered by large language models: Integrating semantic understanding and user preferences, *International Journal of Innovative Research in Engineering and Management*, vol.11, no.4, pp.39-49, 2024.

[39] V. W. Anelli, T. Di Noia, E. Di Sciascio, A. Ragone and J. Trotta, Semantic interpretation of top-n recommendations, *IEEE Transactions on Knowledge and Data Engineering*, vol.34, no.5, pp.2416-2428, DOI: 10.1109/TKDE.2020.3010215, 2020.

[40] L. Xia, C. Huang, Y. Xu and J. Pei, Multi-behavior sequential recommendation with temporal graph transformer, *IEEE Transactions on Knowledge and Data Engineering*, vol.35, no.6, pp.6099-6112, DOI: 10.1109/TKDE.2022.3175094, 2022.

[41] Z. Lin, S. Zang, R. Wang, Z. Sun, J. Senthilnath, C. Xu et al., Attention over self-attention: Intention-aware re-ranking with dynamic transformer encoders for recommendation, *IEEE Transactions on Knowledge and Data Engineering*, vol.35, no.8, pp.7782-7795, DOI: 10.1109/TKDE.2022.320 8633, 2022.

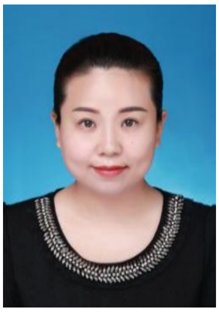
[42] Y. Nitzan, K. Aberman, Q. He, O. Liba, M. Yarom, Y. Gandelsman et al., MyStyle: A personalized generative prior, *ACM Transactions on Graphics (TOG)*, vol.41, no.6, pp.1-10, DOI: 10.1145/3550 454.3555436, 2022.

[43] W. Su, H. Ye, S.-Y. Chen, L. Gao and H. Fu, DrawinginStyles: Portrait image generation and editing with spatially conditioned styleGAN, *IEEE Transactions on Visualization and Computer Graphics*, vol.29, no.10, pp.4074-4088, DOI: 10.1109/TVCG.2022.3178734, 2022.

[44] S. Y. Chang, H.-C. Wu, K. Yan, X. Chen, S. C.-H. Huang and Y. Wu, Personalized multimedia recommendation systems using higher-order tensor singular-value-decomposition, *IEEE Transactions on Broadcasting*, vol.70, no.1, pp.148-160, DOI: 10.1109/TBC.2023.3278111, 2024.

[45] Gemma Team: T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak et al., Gemma: Open models based on Gemini research and technology, *arXiv Preprint*, arXiv: 2403.08295, 2024.

Author Biography



Ping Yang received her Ph.D. from Sungshin Women's University, Korea, in 2017. She currently works at the School of Information Science and Technology, Xichang University, China. Her research interests include smart learning and data analysis. She has led or participated in multiple provincial and municipal research projects, compiled and authored several textbooks and related monographs, and holds multiple patents and awards.

Chengjia Tang earned a Ph.D. from East China Normal University, China, in 2012. Currently with the School of Information Science and Technology, Xichang University, China, Dr. Tang's research interests include computational intelligence, image processing, and big data analytics. Dr. Tang has led or participated in multiple provincial- and municipal-level research projects, compiled and authored several textbooks and related monographs, and holds multiple patents and awards.