AIMS Mathematics, 11(2): 3221–3242.
DOI: 10.3934/math.2026130
Received: 25 September 2025
Revised: 29 December 2025
Accepted: 05 January 2026
Published: 02 February 2026

https://www.aimspress.com/journal/Math

Research article

Data-driven neural network dynamics for customer behavior modeling and personalized E-commerce marketing

Long Li[1] and Yuanyuan Jiang[2,*]

[1] School of Digital Economy, Nanning Vocational and Technical University, NanNing 530000, China
[2] School of Business Administration, Guangxi Vocational Normal University, NanNing 530000, China

* **Correspondence:** Email: jiangyuanyuan8105@163.com.

Abstract: Accurate personalization in e-commerce is challenged by high-dimensional, time-varying customer data and rapidly shifting behavioral patterns. We proposed a dynamics-aware neural modeling framework that integrated a gated recurrent unit (GRU) encoder with a temporal attention mechanism (TAM) to capture learning dynamics, stability, and generalization across business stages. Sparse, multi-source interaction streams were embedded to reduce feature dimensionality before sequence modeling. The GRU extracted long- and short-term dependencies, while TAM assigned time-step-specific weights to highlight behaviorally salient periods for prediction. To handle distributional drift, model parameters were updated dynamically, ensuring alignment with evolving customer behavior. We evaluated convergence during training, temporal prediction stability, and cross-period generalization. For customer-demand forecasting, the method achieved an accuracy of 0.924, a recall of 0.910, and an F1-score of 0.914. In recommendation tasks, the overall click-through rate reached 87.1%, and forecasting accuracy remained stable between 92.4% and 90.9% across business stages, demonstrating robustness to temporal regime changes. Attention-weight analyses further provided interpretability by revealing dominant behavioral windows and influential features. These results indicated that explicitly modeling neural network dynamics—through sequence encoders, temporal attention, and adaptive updating—enhanced prediction accuracy, recommendation effectiveness, and stability under non-stationary conditions, offering a practical and scalable pathway for personalized e-commerce marketing.

Keywords: neural network dynamics; temporal attention mechanism; GRU; customer behavior modeling; non-stationary sequences; stability and generalization; personalized E-commerce marketing;

dynamic feature learning
Mathematics Subject Classification: 62M10, 68T07, 91B60

1. Introduction

With the rapid expansion of the digital economy, e-commerce platforms have become a primary interface between firms and consumers. Customer behavior data in this context are massive, high-dimensional, multi-source, and non-stationary, exhibiting pronounced temporal dynamics and heterogeneity [1–3]. Accurately modeling such behavior and translating it into personalized marketing strategies are critical for improving user experience and platform revenue [4–6]. Data-driven intelligent modeling has therefore emerged as a central pathway to enhance prediction accuracy and recommendation effectiveness, and it is of practical significance for advancing personalized e-commerce systems [7,8].

Despite substantial progress, customer behavior modeling still faces several challenges [9,10]. High-dimensional multi-source signals often contain redundancy and noise, which complicates feature selection and representation learning [11,12]. Behavioral trajectories are temporally dependent and dynamically evolving, and static models struggle to capture such dependencies [13,14]. Furthermore, heterogeneity and abrupt regime shifts—such as promotions, seasonal effects, and external shocks—complicate modeling and can degrade the stability of traditional approaches under distribution drift [15,16]. Collectively, these issues constrain both predictive accuracy and out-of-distribution generalization in real-world e-commerce scenarios.

Prior studies have explored a spectrum of solutions [17,18]. Collaborative filtering leverages user–item similarity but is vulnerable to sparsity and cold-start issues [19,20]. Matrix factorization and deep factorization machines introduce latent interactions yet remain limited for long sequential behaviors and rapidly shifting preferences [21,22]. Recurrent neural networks improve sequence prediction but may suffer from vanishing gradients on long horizons and can fail to capture sudden behavioral changes [23,24]. Attention-based methods enhance feature selection; however, many adopt static attention allocation and insufficiently reflect the time-varying nature of customer behavior [25]. Consequently, existing approaches only partially address the joint problem of key behavior selection and temporal dependency modeling under non-stationarity. These limitations highlight the need for a unified approach that adapts to temporal shifts while preserving interpretability. To address this gap, this article introduces a dynamics-aware neural modeling framework that integrates gated recurrent unit (GRU) with a temporal attention mechanism for enhanced behavior modeling under non-stationary conditions.

To bridge these gaps, we propose a dynamics-aware neural modeling framework that couples a GRU temporal encoder with a temporal attention mechanism (TAM). Multi-source interaction streams are first embedded to map sparse, high-dimensional inputs into compact dense representations, improving feature efficiency. The GRU then extracts long- and short-term dependencies, while TAM assigns time-step-specific weights to emphasize behaviorally salient windows for prediction. To cope with distributional drift, model parameters are updated dynamically so that the learned representation remains synchronized with evolving customer behavior. This design explicitly leverages the learning dynamics of neural networks—covering convergence behavior, stability over time, and generalization across business stages—thereby aligning with the theme of this special issue on the dynamics and

applications of artificial neural networks.

In summary, this work makes four contributions:

- It formulates a dynamics-aware neural architecture that jointly addresses temporal dependency modeling and dynamic key-behavior selection via the integration of GRU and TAM under non-stationarity.
- It incorporates dynamic parameter updating to maintain robustness and stability across shifting business regimes, thereby improving temporal generalization.
- It enhances interpretability by analyzing attention weights over time and across features, clarifying dominant behavioral windows and drivers that inform marketing decisions.
- It provides empirical evidence covering accuracy, stability, and recommendation effectiveness across business stages, demonstrating resilience to temporal regime changes and practicality for real-world personalized e-commerce marketing.

The remainder of this article is organized as follows. Section 2 reviews related work on customer behavior modeling, sequential recommendation, and dynamic neural architectures. Section 3 presents the proposed dynamics-aware neural framework, detailing data preprocessing, GRU-based temporal encoding, temporal attention modeling, and dynamic parameter updating. Section 4 describes the experimental design, datasets, parameter settings, and training strategy. Section 5 reports and analyzes the experimental results from forecasting accuracy, recommendation effectiveness, dynamic adaptability, and attention-based interpretability perspectives. Section 6 concludes the study and discusses limitations and future research directions.

2. Related work

Research on customer behavior modeling has progressed from classical machine learning baselines to dynamics-aware deep architectures tailored for high-dimensional, time-varying data. Comparative evidence indicates that gradient-boosting families such as XGBoost and CatBoost handle complex feature interactions and large-scale settings more effectively than several conventional learners [26]. To better reflect temporal structure, segmentation and prediction pipelines have incorporated time-series features with clustering, while counterfactual analysis has been used to probe the mechanism of marketing interventions [27].

Deep learning has reshaped recommendation by coupling sequence modeling with preference estimation. A time-aware deep collaborative filtering framework demonstrates sustained gains by unifying dynamic user preference modeling with score prediction across multiple datasets [28]. Complementarily, neural matrix factorization fusing explicit and implicit feedback enriches the representation of user-item interactions and improves downstream accuracy [29].

Beyond single-domain settings, recent surveys highlight that deep models alleviate sparsity and cold-start issues in cross-domain recommendation, and identify transfer learning and reinforcement learning as promising directions for scalable personalization [30]. In dynamic e-commerce scenarios, personalized recommendation based on evolving user portraits—integrating profiles, behaviors, and domain knowledge with refined clustering—has reported measurable improvements in platform-level metrics [31]. From the user-behavior perspective, empirical studies grounded in the stimulus-organism-response model (S – O – R) framework reveal that AI-driven recommendation experiences shape click intentions, with privacy concerns and technology acceptance acting as critical moderators [32–35].

A growing body of work further connects learning dynamics with marketing objectives. Data-driven strategy optimization links behavior modeling to segmentation and targeting under non-stationarity [36,37]. Convolutional neural network (CNN)/ long short-term memory network (LSTM) pipelines—which leverage convolutional neural networks for local feature denoising and long short-term memory networks for capturing long-term dependencies—have been explored for understanding e-commerce behavior, emphasizing temporal feature extraction in noisy environments [38,39]. Attention-augmented sequence models integrate temporal salience with representation learning to improve purchase prediction and decision support [40].

At the system level, big-data analytics frameworks seek to translate behavioral signals into business decisions, stressing pipeline robustness and deployability under evolving traffic patterns [41]. Real-time platforms for customer retention combine behavioral tracking with deep predictive analytics to support intervention timing and policy design [42]. Collectively, these advances improve accuracy and recommendation effectiveness, yet persistent gaps remain in unified treatment of non-stationarity, stability, and interpretable generalization across business stages—gaps our GRU-based temporal encoder with temporal attention and dynamic parameter updating is designed to address.

Recent advances in neural recommenders explore large language models and temporal graph networks for dynamic behavior modeling. For example, the hybrid knowledge-augmented news recommender (HKNR) framework utilizes LLaMA-2 for semantic candidate recall and temporal graph neural networks (TGNNs) for user encoding. Specifically, HKNR integrates the deep semantic representations from the LLM with the dynamic relational structures of the TGNN to capture both high-level content meaning and evolving user-item interactions over time. This approach effectively captures structural semantics and long-range dependencies for ephemeral content [43]. However, such methods typically lack an explicit mechanism to identify and interpret the most influential short-term behavioral windows within a sequence. Our work addresses this gap by integrating a dedicated TAM with GRU encoding, which provides fine-grained, time-step-specific importance weighting. This design offers a complementary pathway to achieve stability and interpretable generalization under non-stationarity.

3. Methods

3.1. Data preprocessing and embedding representation

Multi-source customer interaction data exhibits high dimensionality, sparsity, and heterogeneity, posing direct challenges to subsequent sequence modeling. To ensure the validity and consistency of the input data, missing values are processed and outliers are removed to eliminate noise. Index encoding is then used for discrete features, converting non-numeric information into computable integer indices. Numerical features are normalized to ensure that features of different dimensions are within a comparable range, thus avoiding the uneven impact of feature scale differences on modeling. Through these preprocessing steps, customer behavior information on browsing, clicking, purchasing, and multi-dimensional interactions is fully preserved, forming a unified and structured data representation at the input layer, as shown in Table 1.

The unit of analysis is user-level sequences, where each sample represents a chronological sequence of events for a single user. Raw interaction logs were filtered to remove bots and invalid entries, then sessionized into sequences with a maximum length of 30 time steps. Duplicate events

within a session were removed, and sequences were aggregated per user. The data were split chronologically by time stamp to prevent leakage. Table 2 summarizes the record counts after each processing stage.

Table 1. Statistics of customer behavior characteristics.

Feature category	Number of features	Data type	Sample size	Sparsity ratio
Browsing Behavior	1200	Discrete	500,000	83.2%
Clicking Behavior	950	Discrete	500,000	76.5%
Purchasing Behavior	680	Discrete	500,000	69.1%
User Attributes	150	Numerical	500,000	12.3%
Temporal Features	80	Continuous	500,000	5.7%

Table 2. Data flow, filtering steps, and final record counts.

Processing stage	Record count	Description
Raw Logs	550,000	Original interaction data
After Filtering	525,000	Invalid entries removed
After Sessionization	500,000	Sequences per user
Train/Val/Test Split	48,000/12,000/20,000	Chronological split

The data was reduced from 500,000 session-based user sequences to 80,000 sequences ultimately used for modeling through filtering and sampling operations. Removing short-term interactions with sequence lengths less than 5 resulted in approximately 150,000 fewer records. Removing user records with a large number of missing values for key behavioral features resulted in approximately 220,000 fewer records. Stratified random downsampling was performed to manage computational complexity and balance the sample size for behavioral categories, resulting in approximately 50,000 fewer records. This ultimately yielded 80,000 high-quality user sequences for the experiment. After completing data normalization, in order to avoid the inefficiency and overfitting caused by directly inputting high-dimensional sparse features into the neural network, an embedding layer is introduced to densely map the sparse features. To bridge these gaps, we propose a dynamics-aware neural modeling framework that couples a GRU temporal encoder with a TAM. As illustrated in Eq (1), an embedding layer is introduced to densely map high-dimensional sparse features $x \in \mathbb{R}^d$ (where d denotes the input dimension) into a compact representation $z \in \mathbb{R}^k$ through the embedding matrix $E \in \mathbb{R}^{d \times k}$.

$$z=xE. \tag{1}$$

In Eq (1), $z \in R^k$ is the embedded feature after mapping, k and is the embedding dimension. The parameter matrix E is dynamically updated during training, allowing the embedded vector to retain the original semantic information while reducing sparsity and dimensionality. This process makes the input data more compact in vector space, providing efficient and expressive input for the subsequent GRU temporal encoding layer. The embedded representation demonstrates advantages in dimensionality compression, sparsity reduction, storage efficiency, and preservation of semantic

similarity, as shown in Table 3.

Table 3. Comparison of feature embedding representation effects.

Metric	Original features	Embedded features
Feature dimension	3060	128
Sparsity ratio	72.8%	9.6%
Average similarity	0.12	0.47
Storage cost (MB)	1850	220
Training convergence epochs	35	18

3.2. Time series modeling and GRU encoding

After embedding, customer behavior sequence data is organized into a multidimensional time series tensor. Let the input sequence be $X=\{x_1,x_2,\ldots,x_T\}$, where $x_t \in R^d$ represents embedded feature vector T at time step, t is the sequence length, and d is the embedding dimension. Time series modeling uses GRU to encode the sequence to preserve long-term dependency information in the hidden state while reflecting short-term preference changes.

The state update of GRU is controlled by two gate structures, namely the update gate z_t and the reset gate r_t. The update gate determines the fusion ratio between the previous hidden state and the current candidate state, and its calculation method is as shown in Eq (2):

$$z_t=\sigma(W_z x_t+U_z h_{t-1}+b_z). \tag{2}$$

In Eq (2), $\sigma(\cdot)$ denotes the Sigmoid function. The weight matrices for the update gate are represented by $W_z \in \mathbb{R}^{d_h \times d}$ and $U_z \in \mathbb{R}^{d_h \times d_h}$, while $b_z \in \mathbb{R}^{d_h}$ is the corresponding bias vector, d_h is the hidden state dimension , and $h_{t-1} \in R^{d_h}$ is the hidden state at the previous moment.

The reset gate is used to control the degree of retention of the previous hidden state in the generation of the current candidate state. Its calculation method is as shown in Eq (3):

$$r_t=\sigma(W_r x_t+U_r h_{t-1}+b_r). \tag{3}$$

In Eq (3), $W_r \in R^{d_h \times d}$, $U_r \in R^{d_h \times d_h}$, and $b_r \in R^{d_h}$ are the weight matrices and bias vector of the reset gate, respectively.

The update gate and the reset gate, the candidate hidden state is generated $\tilde{h}_t$, and its expression is as shown in Eq (4):

$$\tilde{h}_t=\tanh(W_h x_t+U_h(r_t \odot h_{t-1})+b_h)\cdot \tag{4}$$

In Eq (4), $W_h \in R^{d_h \times d}$, $U_h \in R^{d_h \times d_h}$, $b_h \in R^{d_h}$ are the parameter sets of candidate states, $\odot$ representing element-by-element multiplication operations. The final hidden state is controlled by the update gate, which is a weighted combination of the candidate state and the historical state. The calculation method is as shown in Eq (5):

$$h_t=(1-z_t)\odot h_{t-1}+z_t\odot \tilde{h}_t. \tag{5}$$

In Eq (5), $h_t\in R^{d_h}$ is the time step t. The hidden state sequence $H=\{h_1,h_2,\ldots,h_T\}$ obtained through recursive calculation captures both long-term dependencies and short-term dynamic information, forming a high-dimensional representation of the evolution of customer behavior.

The design of GRU effectively alleviates the vanishing gradient problem through a gating mechanism, ensuring that key behavioral features can be propagated and retained in long sequence data, thereby providing dynamic sequence encoding input for the subsequent temporal attention mechanism.

3.3. Introduction of temporal attention mechanism

In the model, TAM performs a learnable weighting of the temporal representation output by the GRU encoder, so as to highlight the behavioral moments that contribute most to the results during the prediction phase and introduce the distinguishable influence of temporal information. Let the GRU hidden state sequence obtained in Section 3.2 be $H=[h_1,h_2,\ldots,h_T]$, where $h_i\in R^{d_h}$ represents i the hidden state vector at the moment. Temporal information is embedded through vectorized temporal encoding $\tau_i\in R^{d_\tau}$ and participates in attention scoring. The calculation of attention energy is completed by parameterized projection and nonlinear mapping, as shown in Eq (6):

$$e_i=v^T\tanh(W_h h_i+W_t\tau_i+b). \tag{6}$$

In Eq (6), $e_i\in R$ represents i the attention score scalar at the moment, is the projection matrix $W_h\in R^{d_a\times d_h}$ denotes the projection matrix from the hidden state to the attention space, and $W_t\in R^{d_a\times d_\tau}$ is the temporal encoding projection matrix, $v\in R^{d_a}$ is the score vector (mapping the projected representation to a scalar), is $b\in R^{d_a}$ the bias vector, $\tan h(\cdot)$ and is the element-wise hyperbolic tangent activation function. The attention weight is obtained by normalizing the energy and is defined as in Eq (7):

$$\alpha_i=\frac{\exp(e_i)}{\sum_{j=1}^{T}\exp(e_j)}. \tag{7}$$

In Eq (7), $\alpha_i\in(0,1)$ is the normalized position weight, satisfying $\sum_{i=1}^{T}\alpha_i=1$. The context vector is obtained by summing the weighted hidden states as a compressed representation of the overall information of the sequence, defined as Eq (8):

$$c=\sum_{i=1}^{T}\alpha_i h_i. \tag{8}$$

In Eq (8), $c\in R^{d_h}$ is the context vector obtained by attention aggregation. The context vector is concatenated with the current or final hidden state of the GRU and input into the prediction layer to fuse the instantaneous state and the full sequence contribution. The prediction mapping form is as shown in Eq (9):

$$\hat{y}=\phi(W_o[c;h_T]+b_o). \tag{9}$$

In Eq (9), $[c;h_T]\in R^{2d_h}$ represents vector concatenation, $W_o\in R^{d_y\times 2d_h}$ and $b_o\in R^{d_y}$ are the output mapping matrix and bias, respectively, $\phi(\cdot)$ are the activation functions corresponding to the

task, and $\hat{y} \in R^{d_y}$ represents the model output. To maintain a balance between the model's responsiveness to the evolution of customer behavior and training stability, parameter updates employ an online/incremental optimization strategy based on attention-weighted loss and temporal smoothing regularization. The loss function t is defined at the time step as follows:

$$L_t(\theta) = \sum_{n \in B_t} \alpha^{(n)} l(\hat{y}^{(n)}, y^{(n)}) + \lambda \|\theta - \theta_{t-1}\|_2^2. \tag{10}$$

In Eq (10), θ represents all trainable parameters of the model, represents B_t the set of samples participating in the update $\alpha^{(n)}$ at time, t is the attention weight coefficient corresponding to the sample n in its sequence, $l(\cdot,\cdot)$ is the loss function at the sample level, $\lambda \geq 0 \lambda \geq 0$ is the temporal smoothing regularization coefficient, and θ_{t-1} is the last parameter snapshot. The parameters based on this loss are iteratively updated in the gradient direction, and the update rule is:

$$\theta_t = \theta_{t-1} - \eta \nabla_\theta L_t(\theta). \tag{11}$$

In Eq (11), $\eta > 0 \eta > 0$ represents the learning rate. The attention weights redistribute the importance of moments within the sample in the loss, and the temporal smoothing term constrains parameter changes, thereby balancing responsiveness and stability when tracking behavioral distribution drift. The interpretability of TAM is reflected in the weight distribution, which not only reflects the relative contribution of different behavioral moments to the predicted target but also provides weight feedback to the subsequent strategy layer to support personalized marketing strategy adjustments, as shown in Figure 1.

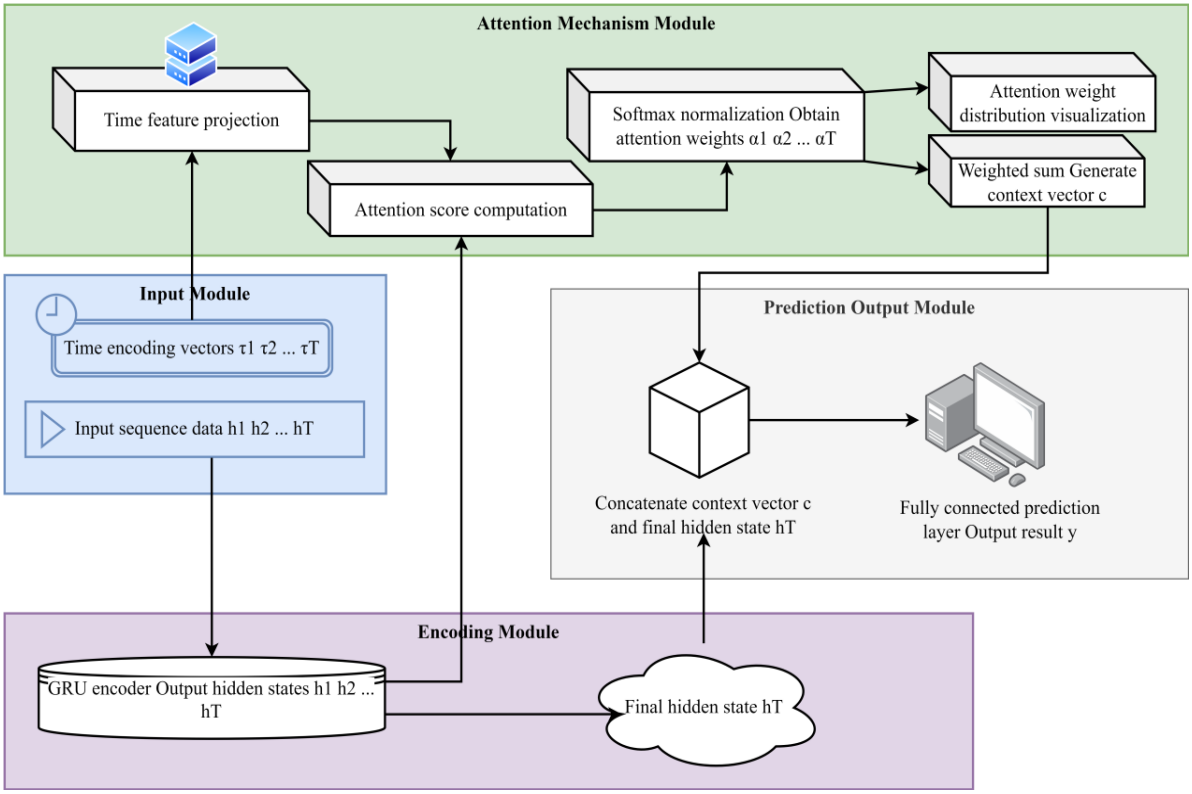

Figure 1. TAM weight distribution.

3.4. Dynamic update and prediction output

This section describes the implementation details of the model's parameter dynamic update strategy and prediction output mechanism, focusing on the aggregation method of temporal attention weights, prediction operator design, loss definition and online/batch parameter update rules, as well as parameter smoothing constraints introduced to prevent historical knowledge from being forgotten. Temporal attention h_t calculates time step scores on the temporal encoder output sequence. The scores use a trainable feedforward mapping and are normalized by a nonlinear transformation. Let the score be e_t, which is defined as:

$$e_t=u^{\top}\tanh(W_a h_t+b_a). \tag{12}$$

In Eq (12), e_t denotes the raw attention score for time step t. $W_a \in \mathbb{R}^{d_a \times d_h}$ and $b_a \in \mathbb{R}^{d_a}$ represent the trainable weight matrix and bias vector of the attention layer, where $u \in \mathbb{R}^{d_a}$ is the score mapping vector. Additionally, d_h corresponds to the hidden dimension of the GRU encoder, whereas d_a denotes the dimensionality of the attention space.

$$\alpha_t=\frac{\exp(e_t)}{\sum_{j=1}^{T}\exp(e_j)}. \tag{13}$$

The denominator in T in Eq (13) is the exponential sum of the sequence length, which is used to ensure $\sum_{t=1}^{T}\alpha_t=1$ that the weight has a probabilistic interpretation. The aggregate representation is obtained by weighted summation of the time series representation based on the normalized weight $\tilde{h}$:

$$\tilde{h}=\sum_{t=1}^{T}\alpha_t h_t. \tag{14}$$

In Eq (14), $\tilde{h} \in R^{d_h}$ is the final sequence representation after integrating temporal attention, which serves as the input of the prediction layer.

The prediction layer uses a fully connected mapping to output the behavior probability distribution $\hat{y}$, and the classification case uses a linear mapping with a normalization function:

$$\hat{y}=\mathrm{softmax}(W_o\tilde{h}+b_o). \tag{15}$$

In Eq (15), $\hat{y} \in R^{C}$ represents C the predicted probability distribution of class behavior, $W_o \in R^{C\times d_h}$ and $b_o \in R^{C}$ is the output layer parameter. The regression or single target probability scoring case will softmax be replaced by σ or identity mapping, but the output still comes from the same weighted representation $\tilde{h}$.

The training loss is defined between the supervision signal y and the prediction $\hat{y}$, and the cross entropy loss is used to evaluate the classification error:

$$L_t=-\sum_{c=1}^{C} y_{t,c}\log\hat{y}_{t,c}+\lambda R(\Theta). \tag{16}$$

In Eq (16), L_t is the loss scalar $y_{t,c}$ at the time instant , tis the true category indicator, $\hat{y}_{t,c}$ is the predicted probability Θ representing the set of all trainable parameters of the model, $R(\Theta)$ is the regularization term used to constrain the parameter norm, λ and is the regularization coefficient. The model uses this loss as the optimization target to update parameters on batch or streaming data.

An adaptive optimizer based on first-order and second-order moment estimation to balance convergence speed and stability. The update rule is as follows (taking Adam as an example):

$$\begin{aligned} m_t &= \beta_1 m_{t-1}+(1-\beta_1)g_t, \\ v_t &= \beta_2 v_{t-1}+(1-\beta_2)g_t^2, \\ \hat{m}_t &= \frac{m_t}{1-\beta_1^t}, \hat{v}_t=\frac{v_t}{1-\beta_2^t}, \\ \theta_{t+1} &= \theta_t-\alpha\frac{\hat{m}_t}{\sqrt{\hat{v}_t+\epsilon}}. \end{aligned} \tag{17}$$

In Eq (17), θ_t represents the value of the parameter vector at the current update step, $g_t=\nabla_\theta L_t$ represents the current gradient, m_t, v_t is the first-order and second-order moment estimate , β_1,β_2 is the momentum decay hyperparameter, α is the learning rate, ϵ is the numerical stability term, and symbolic operations and vector dimensions are defined according to the parameter vector. To maintain the model's robustness to behavioral distribution drift and suppress the rapid forgetting of historical information, a parameter sliding average is used as the long-term steady-state parameter representation:

$$\bar{\theta}_t=\rho\bar{\theta}_{t-1}+(1-\rho)\theta_t. \tag{18}$$

In Eq (18), $\bar{\theta}_t$ is the sliding average parameter, ρ and is the smoothing coefficient. The initial value $\bar{\theta}_0$ can be set to random initialization or pre-trained parameters. The sliding average parameter is used as a model deployment parameter during the online prediction phase to improve stability. During the training phase, the real-time parameter is used to respond to new information. Both are compared in parallel during the evaluation phase to measure the trade-off between dynamic adaptability and stability.

The online update mechanism is executed in a streaming scenario based on preset trigger conditions. The trigger conditions are determined by the sample size and loss growth rate within the time window. When triggered, a small batch gradient update is applied to the new sample and the sliding average parameters are updated at the same time. If batch training is used, the same optimization and parameter smoothing process is performed after each round of iteration. The output end sorts the predicted probabilities in descending order and makes them available to downstream recommendation and marketing decision modules. The probability threshold and ranking strategy are

automatically selected based on the business indicators on the validation set to ensure accurate targeting and frequency control of marketing delivery. The entire dynamic update and prediction output process forms a closed loop in the system-level view. Weight allocation, aggregate representation, loss feedback, and parameter smoothing constitute the core links of continuous adaptation, as shown in Figure 2.

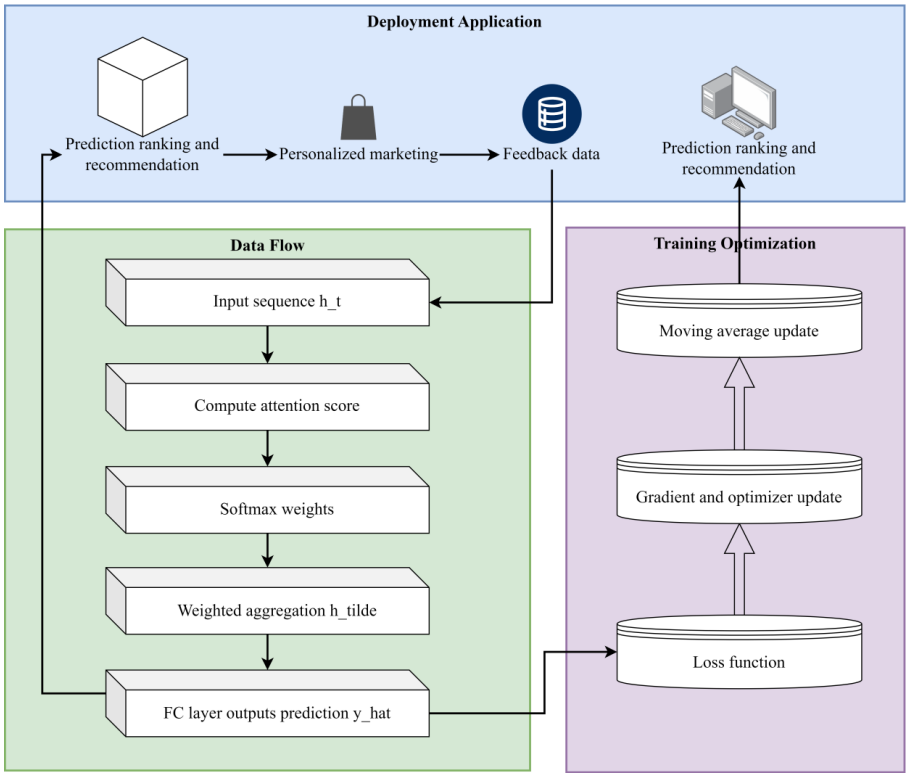

Figure 2. Dynamic update and prediction output flow chart.

4. Experimental design and implementation

4.1. Dataset and experimental environment

During the experimental design phase, the raw customer interaction data needs to be rationally partitioned to ensure sufficient sample coverage during training, while retaining some data for testing the model's generalization performance. The dataset was split chronologically by time stamp into periods T1 to T5, with no random shuffling. A rolling-window evaluation protocol was adopted: for each period Ti, training used data from Ti-1 and earlier, while validation and testing used data from Ti. Feature computations (e.g., windowed aggregates) were strictly based on historical data up to the prediction time to avoid future information leakage. Results in Section 5 are reported under this protocol. To avoid overfitting and improve the effectiveness of parameter adjustment, the research divided the dataset into training, validation, and test sets based on different functions. Additionally, subsets of behavioral logs and transaction records were constructed to support analysis of temporal characteristics and consumer habits, ensuring the scientific and complete evaluation of the model under multi -dimensional conditions. Table 4 shows the data partitioning.

Table 4. Dataset division table.

Dataset type	Sample size	Proportion	Feature description
Training set	48,000	60%	Covers main behavior features
Validation set	12,000	15%	Used for hyperparameter tuning
Test set	20,000	25%	Evaluates generalization performance
Behavior log subset	15,000	-	Contains click, browse, add-to-cart
Transaction subset	10,000	-	For purchase and repurchase analysis

In terms of data distribution, the training set consists of 48,000 records, accounting for 60% of the total data. This is primarily because the model's sequence modeling requires a large number of samples to support parameter learning. The validation set consists of 12,000 records, representing 15% of the total data. This proportion is determined to ensure model stability during hyperparameter tuning and to avoid bias caused by too few samples. The test set consists of 20,000 records, representing 25% of the total data. This large data size allows for evaluation of generalization capabilities in real-world application scenarios and ensures more representative results. The behavioral log subset consists of 15,000 records. This high number stems from the fact that customer browsing and clicking data are much more frequently generated than transaction records. The transaction record subset consists of only 10,000 records because actual purchasing behavior in e-commerce scenarios is much more sparse than browsing behavior. This overall distribution ensures sufficient training, stable validation, and reliable testing.

4.2. Parameter setting and training strategy

In the experimental design, to ensure the stability and repeatability of the proposed model during training, a parameter system covering network structure configuration, training strategy, and regularization measures was constructed. The network structure component focuses on expressing sequence modeling capabilities, the training strategy component defines the optimization path and iteration method, and the regularization component is used to mitigate the risk of overfitting. The overall parameter configuration directly affects the model 's performance under high-dimensional behavioral data, so it is systematically organized as shown in Table 5.

Table 5. Experimental parameter configuration table.

Category	Parameter	Value	Description
Structure	Hidden Size	128	GRU hidden state dimension
Structure	Time Steps	30	Max input sequence length
Training	Batch Size	64	Samples per batch
Training	Learning Rate	0.001	Initial rate for Adam
Training	Optimizer	Adam	Adaptive optimization
Training	Epochs	100	Max training iterations
Regularization	Dropout	0.3	Drop ratio to reduce overfitting
Regularization	L2 Penalty	1e-5	L2 regularization strength

This table summarizes the experimental parameters from three perspectives. The network structure configuration shows the hidden layer dimensions and time step length, demonstrating the model's ability to model the sequential dependencies between customer behaviors. The training strategy section displays the batch size, learning rate, optimizer type, and number of iterations; these factors determine the model's convergence speed and training efficiency. Regularization measures, including the dropout ratio and L2 regularization coefficient, are used to reduce instability caused by model complexity. Through the rational coordination of multiple parameter types, the experimental training process achieved consistency in both structure and strategy.

All baseline models and the proposed method used identical input features, preprocessing, sequence length (30), and padding rules. Hyperparameter search included learning rate [0.001, 0.01], hidden size [64, 128], and dropout [0.2, 0.5], with 50 trials per model using Bayesian optimization. Early stopping was based on validation loss with a patience of 10 epochs. Results report mean and standard error over 5 runs with fixed random seeds. Training used a single NVIDIA Tesla V100 GPU, with comparable time (2–4 hours per model) across methods.

5. Results analysis

5.1. Analysis of customer demand forecast accuracy

During the experiment, the research focused on customer demand forecasting. Four models were compared and analyzed: collaborative filtering, matrix factorization, GRU, and GRU+TAM (a temporal attention mechanism) to comprehensively evaluate their performance in the forecasting task. The experiment measured performance across multiple dimensions, examining not only static metrics such as accuracy, precision, recall, and F1 score, but also plotting receiver operating characteristic (ROC) curves to assess the classification capabilities of different models. Training convergence curves were used to examine the convergence speed and stability during the iterative process. The results, obtained after systematic data modeling and visualization, are shown in Figure 3.

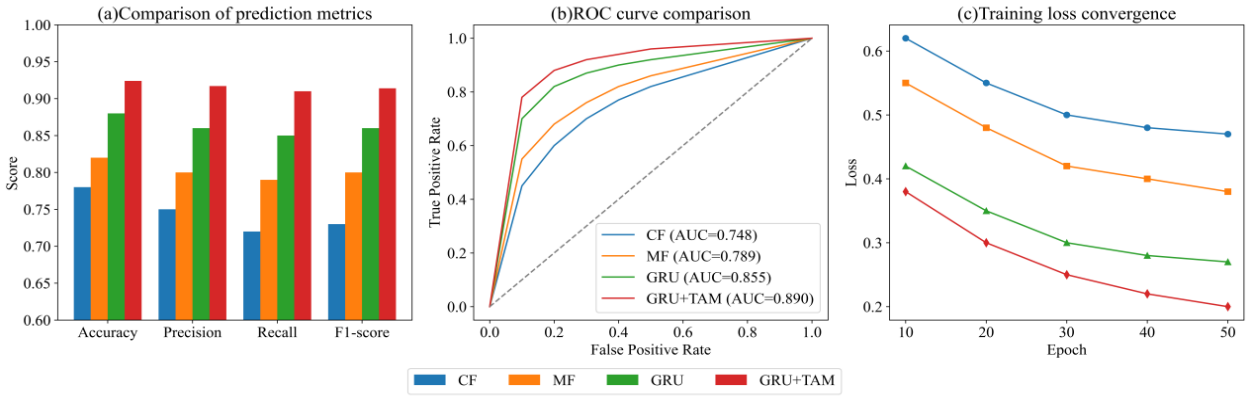

Figure 3. Multi-dimensional comparative analysis of customer demand forecasts. (a) Comparison of prediction indicators. (b) ROC curve comparison. (c) Training convergence curve.

The results show that GRU+TAM achieves the highest performance across all four core metrics,

with accuracy of 0.924, precision of 0.917, recall of 0.910, and F1 score of 0.914. This result is closely related to TAM's weighting mechanism for key behavioral time steps, as this mechanism enhances the utilization of key features that influence prediction, while GRU, which relies solely on time series modeling, has limitations in weight allocation. The ROC curve further shows that GRU+TAM maintains a lead in AUC, achieving a TPR of 0.88 at an FPR of 0.2, compared to 0.82 for GRU. This difference stems from the attention mechanism, which enables the model to capture more true positive examples while maintaining a low false positive rate. The training convergence curve reflects the stability of different models during the iteration process. The loss of GRU+TAM drops to 0.25 at the 30th epoch, lower than the 0.30 of GRU and the 0.42 of MF. This is primarily due to the attention mechanism optimizing the efficiency of sequence modeling during gradient propagation. Overall, the introduction of TAM significantly improves the model's advantages in prediction accuracy and training efficiency.

5.2. Analysis of recommendation click-through rate performance

In the experimental design for recommendation click-through rate, this part of the work verified the model's applicability in complex interactive scenarios by comparing various modeling methods in terms of overall performance, user grouping, and recommendation placement. Click-through rate is defined as the ratio of clicked items to total recommended items in an offline evaluation, computed as the hit rate on held-out test data. Click-through rate lift measures the relative improvement in click-through rate over a baseline model. These metrics are derived from the same classification task as accuracy and F1-score, where the label space includes user click actions on items. The prediction task is binary classification of click events, and all metrics are computed offline without online exposure. The experiment first examined the performance of different recommendation algorithms in terms of overall click-through rate. It also introduced user activity grouping to analyze the model's adaptability to different usage habits. It further compared the click-through rate distribution of each method at different recommendation list positions, comprehensively demonstrating the differences in recommendation effectiveness across multiple dimensions. The experimental results are shown in Figure 4.

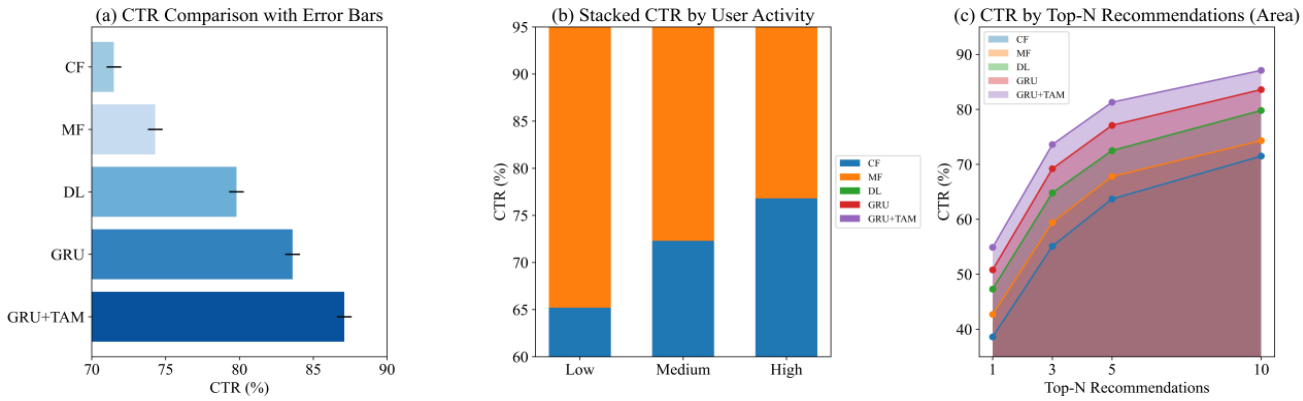

Figure 4. Comparison of recommended click-through rate performance. (a) Overall click-through rate comparison of models. (b) Click-through rates at different user activity levels. (c) Click-through rates of different recommendation positions.

The results show that the GRU+TAM approach achieved an overall click-through rate of 87.1%, significantly outperforming collaborative filtering's 71.5% and matrix factorization's 74.3%. This is due to the introduction of the temporal attention mechanism, which allows the model to better capture key behaviors. In user segmentation experiments, highly active users generally achieved higher click-through rates than less active users. For example, with GRU+TAM, the click-through rate for the highly active group reached 90.2%, while that for the less active group was only 78.9%. This is because high-frequency interactions provide the model with richer temporal features, enhancing its ability to discern preferences. In the recommendation position experiment, the click-through rate for the top-1 recommendation was 54.9% with GRU+TAM, gradually increasing to 87.1% for the top-10 recommendation. However, the collaborative filtering approach achieved a mere 38.6% click-through rate for the top-1 recommendation. This difference is due to the sequential modeling approach more accurately capturing the user's immediate needs in the top recommendations, thereby increasing the relevance of the top recommendations. Overall, the results demonstrate that the combination of temporal modeling and the attention mechanism improves the performance of recommendation systems across multiple dimensions.

5.3. Model dynamic adaptability analysis

In a dynamic e-commerce environment, customer behavior distribution exhibits phased variations over time. If modeling methods fail to respond to temporal evolution, predictive performance will gradually decline. To validate the proposed method's adaptability across different business phases, this study compared the performance of a static model with that of a dynamic neural network across five key time periods. Stability and generalizability were analyzed by combining error distribution with user group segmentation. These time periods were the initial launch of the system (T1), the user expansion phase in the third month (T2), the peak promotional activity phase in the sixth month (T3), the market stabilization phase in the ninth month (T4), and the end of the year in the twelfth month (T5). The experimental process covered three perspectives: overall accuracy, error fluctuation, and group differences. The results are shown in Figure 5.

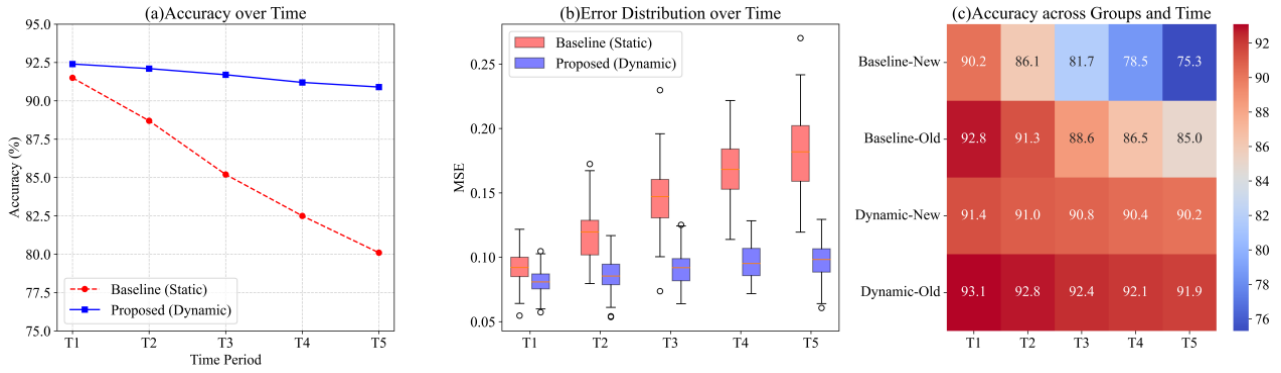

Figure 5. Comprehensive results of model dynamic adaptability. (a) Accuracy trends across different business stages. (b) Time evolution of prediction error distribution. (c) Prediction accuracy of user groups in different time periods.

As shown in Figure 5(a), the static model achieved an accuracy of 91.5% at the beginning of the

system's launch, but by the end of the year, it had dropped to 80.1%, a decrease of 11.4 percentage points. This indicates that its performance degraded significantly under changing temporal distributions. The dynamic model maintained an accuracy range of 92.4% to 90.9% across the five phases, fluctuating by only 1.5 percentage points. This difference stems from the dynamic update mechanism continuously capturing new behavioral patterns during parameter iteration, thus preventing performance degradation over time. Figure 5(b) further shows that the mean MSE distribution of the static model at the end of the year was 0.184, a significant increase from 0.094 at the beginning of the system's launch, and the variance increased, indicating that the uncertainty of the prediction results increased with changes in the data distribution. The dynamic model achieved an MSE mean of 0.097 at the end of the year, only increasing by 0.015 from the beginning, maintaining a relatively stable error range. This is due to the temporal attention mechanism effectively filtering key behavioral features and suppressing the noise accumulation effect. Figure 5(c) shows the results for different user groups. For new users, the accuracy of the static model dropped from 90.2% at the beginning of the launch to 75.3% at the end of the year, while the accuracy of the dynamic model remained stable between 91.4% and 90.2% during the same period. A similar trend was observed for the established user group. This indicates that static modeling methods struggle to adapt to groups with rapidly changing behavior patterns, while the key reason dynamic methods maintain stable output is their more flexible weighting mechanism for short-term behavioral features. In summary, the dynamic neural network combined with the temporal attention mechanism demonstrates strong adaptability and robustness across different business phases.

5.4. Feature importance and attention weight analysis

In this experiment, we modeled the dynamic characteristics of customer behavior sequences, focusing on the distribution of attention across different types of interactions over time. By constructing a temporal attention mechanism model, we first weighted the continuous interaction sequence to reveal the varying importance of each behavior to the prediction results over different time periods. We then calculated the global average attention value for each feature to measure its long-term contribution. Finally, we aggregated the data across the stage dimension to compare feature importance across different time intervals. This process yielded the visualization shown in Figure 6.

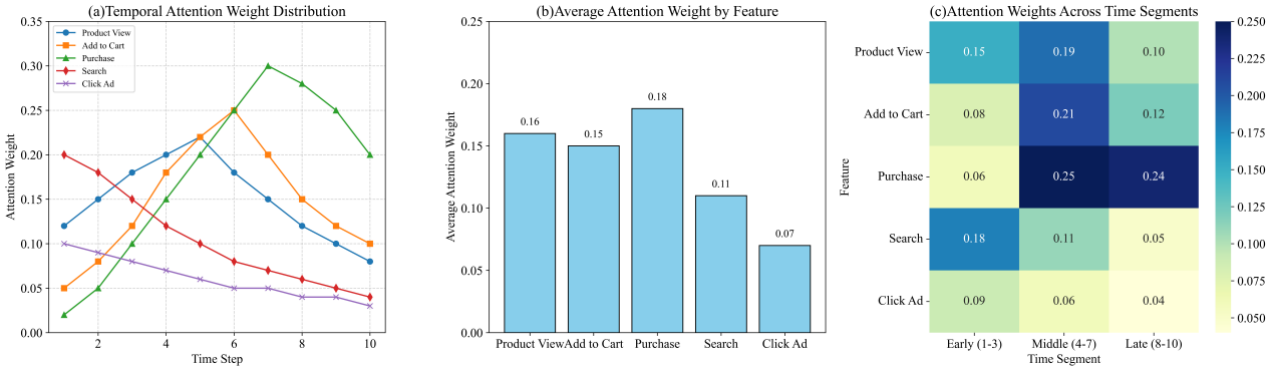

Figure 6. Attention weight analysis of customer behavior characteristics. (a) Time attention weight distribution. (b) Average attention weight of each feature. (c) Attention weights across different time periods.

The results show that the average attention value for purchase is 0.18, substantially higher than the 0.07 assigned to click ads. This discrepancy reflects the stronger direct relevance of purchase behavior to the prediction target, whereas ad clicks exhibit greater randomness, resulting in lower attention weighting. The attention weight for product view reaches 0.20 in the early stages but declines to 0.08 later, indicating that browsing behavior is highly informative at the initial stages of the consumer journey but is gradually overshadowed by conversion-oriented actions. The weight for add to cart peaks at 0.25 in the middle stages, underscoring the role of carting behavior as a key signal of preference conversion. Meanwhile, search receives a weight of 0.18 early on but decreases to 0.05, consistent with search behavior serving as an exploratory action that becomes less informative at later stages. These differentiated patterns demonstrate the sensitivity of the temporal attention mechanism in capturing evolving customer behavior dynamics.

5.5. Stability and generalization analysis

This section evaluates the model's stability and generalization ability from multiple perspectives. By comparing datasets from different e-commerce platforms, we assess the consistency of model performance across scenarios. Experiments conducted over multiple time periods examine month-to-month fluctuations in prediction accuracy. Segmenting users into new and active groups further reveals how different interaction patterns influence prediction performance. Finally, key metrics—accuracy, click-through rate improvement, recall, and F1-score—are jointly analyzed through a performance matrix to provide a comprehensive evaluation across dimensions. These results are summarized in Figure 7, Model Stability and Generalization Performance Comparison.

The results show that the accuracy of different data sets ranged from 91.5 to 92.4, and the click-through rate improvement ranged from 86.5 to 87.3. The main reason for the differences is the different user behavior characteristics and interaction frequencies of each platform, but the model structure's capture of sequential information reduces the instability caused by such differences. In cross-time period experiments, the accuracy remained between 91.8 and 92.2, and the click-through rate improvement ranged from 86.9 to 87.2. The slight fluctuations were caused by changes in seasonal user activity. In user group experiments, the accuracy of new users was concentrated at 91.0, while that of active users was around 93.0. This difference is due to the longer interaction sequences of active users, which provide more complete temporal characteristics. The comprehensive heat map shows that the F1-score remains stable between 91.3 and 92.0, reflecting the high reliability of the model in different scenarios and evaluation dimensions.

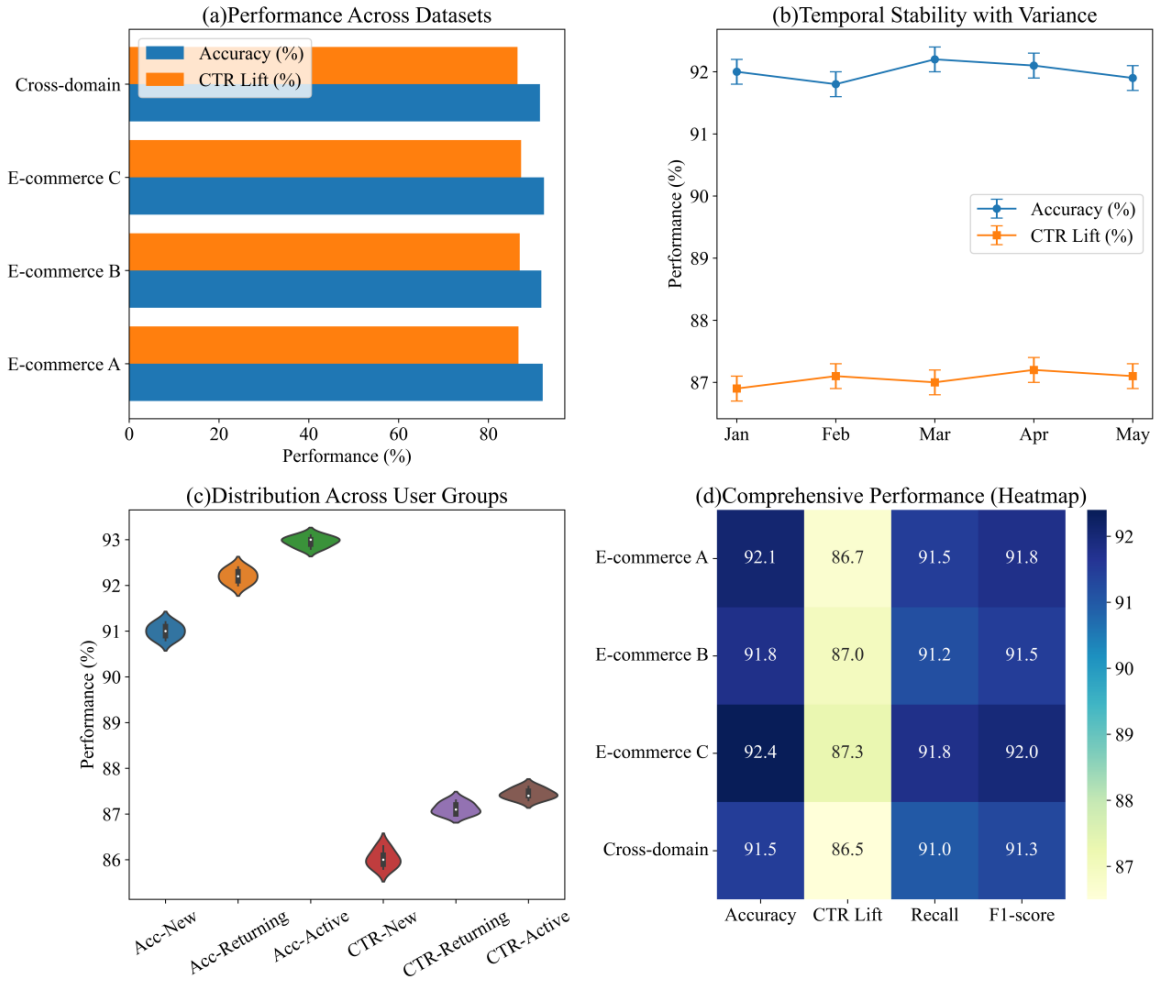

Figure 7. Model stability and generalization results. (a) Performance on different datasets. (b) Time series stability. (c) User group distribution. (d) Comprehensive performance across multiple indicators.

6. Conclusions

This study tackled dynamic modeling of e-commerce customer behavior by proposing a dynamics-aware neural architecture that couples a gated recurrent unit (GRU) temporal encoder with a temporal attention mechanism (TAM) and employs dynamic parameter updating. Multi-source, high-dimensional interaction streams are first embedded to obtain compact representations; the GRU then captures long- and short-term dependencies, while TAM assigns time-step-specific weights that highlight behaviorally salient windows and features. Empirically, the framework achieves an accuracy of 0.924, a recall of 0.910, and an F1-score of 0.914 on customer-demand forecasting, outperforming collaborative filtering and matrix-factorization baselines. In recommendation evaluation, the overall click-through rate (CTR) reaches 87.1% versus ~70% for the baseline, indicating stronger relevance. Under dynamic adaptability tests across annual business stages, accuracy remains within 92.4%–90.9%, whereas a static comparator degrades by more than 11 percentage points, evidencing robustness to distributional shifts. Attention analyses further reveal that purchase and add-to-cart signals are dominant drivers of preference conversion. The proposed method improves prediction accuracy, recommendation

effectiveness, and temporal stability, supporting personalized marketing in evolving user environments.

Study limitations: This study has several limitations. The dataset is from a single e-commerce platform, which may limit generalizability to other domains. Although we adopted a time-based evaluation protocol, offline CTR metrics may not fully reflect online performance. The reproducibility is constrained by the use of proprietary data, though aggregate statistics are provided. External factors such as promotions were not explicitly controlled.

Future work: Future research will include validation across diverse domains and platforms, integration of causal modeling to isolate preference shifts, and development of uncertainty estimation methods for risk-aware decision making. Continual and meta-learning approaches will be explored for faster adaptation, alongside multimodal data fusion and reinforcement learning for real-time interventions.

Author contributions

Long Li: Conceptualization, methodology, software, writing – original draft; Yuanyuan Jiang: Supervision, funding acquisition, writing – review & editing. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the Guangxi Vocational Education and Teaching Reform Research Project "A study on the cultivation model of innovation and entrepreneurship-oriented professional undergraduate talents in cross-border E-commerce for ASEAN from the perspective of industry-education integration" (Grant No. GXGZJG2025A047).

Data availability

The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request. Aggregate statistics necessary to reproduce the reported results are provided in the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. A. A. Alsmadi, A. Shuhaiber, M. Al-Okaily, A. Al-Gasaymeh, N. Alrawashdeh, Big data analytics and innovation in e-commerce: Current insights and future directions, *J. Financ. Serv. Mark.*, **29** (2023), 1635–1652. https://doi.org/10.1057/s41264-023-00235-7

2. T. Wang, N. Li, H. Wang, J. Xian, J. Guo, Visual analysis of e-commerce user behavior based on log mining, *Adv. Multimedia*, **2022** (2022), 4291978. https://doi.org/10.1155/2022/4291978
3. L. Theodorakopoulos, A. Theodoropoulou, Leveraging big data analytics for understanding consumer behavior in digital marketing: A systematic review, *Hum. Behav. Emerg. Technol.*, **2024** (2024), 3641502. https://doi.org/10.1155/2024/3641502
4. S. Chandra, S. Verma, W. M. Lim, S. Kumar, N. Donthu, Personalization in personalized marketing: trends and ways forward, *Psychol. Mark.*, **39** (2022), 1529–1562. https://doi.org/10.1002/mar.21670
5. Z. Liu, Prediction model of e-commerce users' purchase behavior based on deep learning, *Front. Bus. Econ. Manag.*, **15** (2024), 147–149. https://doi.org/10.54097/p22ags78
6. Y. D. Al-Otaibi, Enhancing e-commerce strategies: A deep learning framework for customer behavior prediction, *Eng. Technol. Appl. Sci. Res.*, **14** (2024), 15656–15664. https://doi.org/10.48084/etasr.7945
7. A. T. Rosário, J. C. Dias, How has data-driven marketing evolved: challenges and opportunities with emerging technologies, *Int. J. Inf. Manag. Data Insights*, **3** (2023), 100203. https://doi.org/10.1016/j.jjimei.2023.100203
8. S. C. Necula, Exploring the impact of time spent reading product information on e-commerce websites: A machine learning approach to analyze consumer behavior, *Behav. Sci.*, **13** (2023), 439. https://doi.org/10.3390/bs13060439
9. S. De, P. Prabu, Predicting customer churn: A systematic literature review, *J. Discrete Math. Sci. Cryptogr.*, **25** (2022), 1965–1985. https://doi.org/10.1080/09720529.2022.2133238
10. S. Gupta, P. S. Kushwaha, U. Badher, P. Chatterjee, E. D. R. S. Gonzalez, Identification of benefits, challenges, and pathways in e-commerce industries: An integrated two-phase decision-making model, *Sustain. Oper. Comput.*, **4** (2023), 200–218. https://doi.org/10.1016/j.susoc.2023.08.005
11. Y. Li, L. Hu, W. Gao, Multi-label feature selection with high-sparse personalized and low-redundancy shared common features, *Inf. Process. Manag.*, **61** (2024), 103633. https://doi.org/10.1016/j.ipm.2023.103633
12. H. Cai, J. Meng, S. Yuan, J. Ren, A robust sequential recommendation model based on multiple feedback behavior denoising and trusted neighbors, *Neural Process. Lett.*, **56** (2024), 1. https://doi.org/10.1007/s11063-024-11438-x
13. Z. Ren, X. He; D. Yin; M. de Rijke, Information discovery in e-commerce, *Found. Trends Inf. Retr.*, **18** (2024), 417–690. https://doi.org/10.1561/1500000097
14. Y. Song, W. Wang, S. Yao, Customer acquisition via explainable deep reinforcement learning, *Inf. Syst. Res.*, **36** (2025), 534–551. https://doi.org/10.1287/isre.2022.0529
15. Y. Zhong, J. Zhou, P. Li, J. Gong, Dynamically evolving deep neural networks with continuous online learning, *Inf. Sci.*, **646** (2023), 119411. https://doi.org/10.1016/j.ins.2023.119411
16. L. Tian, X. Wang, A dynamic prediction neural network model of cross-border e-commerce sales for virtual community knowledge sharing, *Comput. Intell. Neurosci.*, **2022** (2022), 2529372. https://doi.org/10.1155/2022/2529372
17. Y. H. Alfaifi, Recommender systems applications: Data sources, features, and challenges, *Information*, **15** (2024), 660. https://doi.org/10.3390/info15100660
18. V. Norouzi, Predicting e-commerce CLV with neural networks: The role of NPS, ATV, and CES, *J. Econ. Technol.*, **2** (2024), 174–189. https://doi.org/10.1016/j.ject.2024.04.004

19. P. Magron, C. Févotte, Neural content-aware collaborative filtering for cold-start music recommendation, *Data Min. Knowl. Discov.*, **36** (2022), 1971–2005. https://doi.org/10.1007/s10618-022-00859-8
20. H. Papadakis, A. Papagrigoriou, C. Panagiotakis, E. Kosmas, P. Fragopoulou, Collaborative filtering recommender systems taxonomy, *Knowl. Inf. Syst.*, **64** (2022), 35–74. https://doi.org/10.1007/s10115-021-01628-7
21. T. F. Boka, Z. Niu, R. B. Neupane, A survey of sequential recommendation systems: techniques, evaluation, and future directions, *Inf. Syst.*, **125** (2024), 102427. https://doi.org/10.1016/j.is.2024.102427
22. M. Ma, G. Wang, T. Fan, Improved DeepFM recommendation algorithm incorporating deep feature extraction, *Appl. Sci.*, **12** (2022), 11992. https://doi.org/10.3390/app122311992
23. M. I. Stan, O. Rhodes, Learning long sequences in spiking neural networks, *Sci. Rep.*, **14** (2024), 21957. https://doi.org/10.1038/s41598-024-71678-8
24. J. Smith, S. De Mello, J. Kautz, S. Linderman, W. Byeon, Convolutional state space models for long-range spatiotemporal modeling, *Adv. Neural Inf. Process. Syst.*, **36** (2023), 80690–80729.
25. Y. Zhu, S. Yao, X. Sun, Feature interaction dual self-attention network for sequential recommendation, *Front. Neurorobot.*, **18** (2024), 1456192. https://doi.org/10.3389/fnbot.2024.1456192
26. J. Lin, Application of machine learning in predicting consumer behavior and precision marketing, *PLoS One*, **20** (2025), e0321854. https://doi.org/10.1371/journal.pone.0321854
27. M. E. Jalal, A. Elmaghraby, Analyzing the dynamics of customer behavior: A new perspective on personalized marketing through counterfactual analysis, *J. Theor. Appl. Electron. Commer. Res.*, **19** (2024), 1660–1681. https://doi.org/10.3390/jtaer19030081
28. R. Wang, Z. Wu, J. Lou, Y. Jiang, Attention-based dynamic user modeling and deep collaborative filtering recommendation, *Expert Syst. Appl.*, **188** (2022), 116036. https://doi.org/10.1016/j.eswa.2021.116036
29. H. Liu, W. Wang, Y. Zhang, R. Gu, Y. Hao, Neural matrix factorization recommendation for user preference prediction based on explicit and implicit feedback, *Comput. Intell. Neurosci.*, **2022** (2022), 9593957. https://doi.org/10.1155/2022/9593957
30. M. O. Ayemowa, R. Ibrahim, Y. A. Bena, A systematic review of the literature on deep learning approaches for cross-domain recommender systems, *Decis. Anal. J.*, **13** (2024), 100518. https://doi.org/10.1016/j.dajour.2024.100518
31. J. Li, S. Bao, Personalized recommendation method of e-commerce products based on in-depth user interest portraits, *Int. J. Inf. Technol. Web Eng.*, **19** (2024), 15. https://doi.org/10.4018/ijitwe.335123
32. J. Yin, X. Qiu, Y. Wang, The impact of AI-personalized recommendations on clicking intentions: Evidence from Chinese e-commerce, *J. Theor. Appl. Electron. Commer. Res.*, **20** (2025), 21. https://doi.org/10.3390/jtaer20010021
33. O. R. Amosu, P. Kumar, A. Fadina, Y. M. Ogunsuji, S. Oni, O. Faworaja, et al., Data-driven personalized marketing: deep learning in retail and e-commerce, *World J. Adv. Res. Rev.*, **23** (2024), 788–796. https://doi.org/10.30574/wjarr.2024.23.2.2395

34. F. U. Ojika, W. O. Owobu, O. A. Abieba, O. J. Esan, B. C. Ubamadu, A. I. Daraojimba, Enhancing user interaction through deep learning models: A data-driven approach to improving consumer experience in e-commerce, *J. Front. Multidiscip. Res.*, **4** (2023), 126–137. https://doi.org/10.54660/.ijfmr.2023.4.1.126-137
35. G. Kumar, G. Devayani, C. Saraswathi, R. M. Kani, S. Dongre, M. S. M. Mallick, Exploring consumer behavior in e-commerce with AI personalization and market trends analysis models, In: *2025 3rd International conference on integrated circuits and communication systems (ICICACS)*, India: IEEE, 2025, 1–6. https://doi.org/10.1109/icicacs65178.2025.10967904
36. B. Sun, Data-driven personalized marketing strategy optimization based on user behavior modeling and predictive analytics: sustainable market segmentation and targeting, *PLoS One*, **20** (2025), e0328151. https://doi.org/10.1371/journal.pone.0328151
37. A. G. Aishwarya, H. K. Su, W. K. Kuo, Personalized e-commerce: Enhancing customer experience through machine learning-driven personalization, In: *2024 IEEE International conference on information technology, electronics and intelligent communication systems (ICITEICS)*, India: IEEE, 2024, 1–5. https://doi.org/10.1109/iciteics61368.2024.10624901
38. T. Mahmud, R. Chakma, R. Akther, M. T. Aziz, T. Rahman, M. S. Hossain, K. Andersson, Leveraging data-driven decision making for e-commerce growth: a machine learning framework, In: *Intelligent computing and optimization*, Cham: Springer, **1167** (2023), 249–261. https://doi.org/10.1007/978-3-031-73318-5_21
39. L. Singla, S. Bansal, P. Banyal, S. Abrol, A. Dhiman, Understanding e-commerce consumer behavior with CNN-LSTM: A data-driven approach, In: *2025 5th International conference on expert clouds and applications (ICOECA)*, India: IEEE, 2025, 872–876. https://doi.org/10.1109/icoeca66273.2025.00153
40. B. Ye, Predicting e-commerce customer purchase behavior using LSTM-attention neural networks and data optimization strategies, *Informatica*, **49** (2025). https://doi.org/10.31449/inf.v49i31.8875
41. P. Pande, A. K. Kulkarni, V. Ramalingam, Big data analytics in e-commerce driving business decisions through customer behavior insights, *ITM Web Conf.*, **76** (2025), 05001. https://doi.org/10.1051/itmconf/20257605001
42. M. Ononiwu, T. I. Azonuche, O. F. Okoh, J. O. Enyejo, AI-driven predictive analytics for customer retention in e-commerce platforms using real-time behavioral tracking, *Int. J. Sci. Res. Mod. Technol.*, **2** (2023), 17–31. https://doi.org/10.38124/ijsrmt.v2i8.561
43. J. Liu, Knowledge-augmented news recommendation via LLM recall, temporal GNN encoding, and multi-task ranking, In: *2025 6th International conference on big data & artificial intelligence & software engineering (ICBASE)*, Malaysia: IEEE, 2025, 141–144. https://doi.org/10.1109/icbase66587.2025.11181275