International Scientific Technical and Economic Research **(ISSN: 2959-1309)**
Vol. 4, No. 2, (June 2026), pp. 98-123
Journal homepage: https://www.istaer.online

RESEARCH ARTICLE **DOI: 10.71451/ISTAER2617**

Research on the Fusion Model of DeepFM and XGBoost for Digital Consumer Behavior Prediction

Haotian Zhou*

School of Business Administration/School of Marxism, China University of Petroleum-Beijing at Karamay, Karamay, Xinjiang, China

Abstract: To leverage the complementary characteristics of deep models and tree models in feature interaction modeling for digital consumer behavior prediction, this paper proposes a dual channel fusion model of DeepFM and XGBoost. In this model, an optimized DeepFM branch and an enhanced XGBoost branch are constructed using a feature shunting mechanism, and the dynamic weighted fusion and attention mechanism based on sample features are introduced to realize the adaptive combination of the two branch outputs. At the same time, a feature interaction enhancement algorithm is designed, which combines the depth implicit representation with the rule features of the tree model by multiplication, and further improves the depiction ability of high-order interaction. Experiments on real e-commerce user behavior and ad click through rate data sets show that the AUC of this model reaches 0.879, LogLoss drops to 0.342, which is 5.4% higher and 11.6% lower than DeepFM, and 8.3% higher and 14.7% lower than XGBoost, respectively. Ablation experiments verify the effectiveness of the dynamic weighted fusion and feature enhancement module, and the performance degradation is 6.7% and 10.5%, respectively. The robustness test showed that the AUC remained at 0.839 and 0.851 under the proportion of 30% missing features and 1% positive samples, and the click-through rate in online simulations increased to 4.93%, which was 1.6% higher than that of the industrial reference system. The proposed model has significant advantages in prediction accuracy and stability.

Keywords: DeepFM; XGBoost; Fusion model; Consumer behavior prediction; Dynamic weighted fusion

How to Cite: Zhou, H. (2026). Research on the Fusion Model of DeepFM and XGBoost for Digital Consumer Behavior Prediction. *International Scientific Technical and Economic Research, 4*(2), 98–123. **https://doi.org/10.71451/ISTAER2617**

Article history: Received: 23 Jan 2026; **Revised:** 05 Mar 2026; **Accepted:** 15 Apr 2026; **Published:** 23 Apr 2026

1. INTRODUCTION

Digital consumer behavior prediction has become the core task in the field of recommendation system and precision marketing [1],[2]. With the rapid development of e-commerce platform and mobile Internet, click, purchase, favorite/add-to-cart, and other

---

***Corresponding author:** Haotian Zhou, School of Business Administration/School of Marxism, China University of Petroleum-Beijing at Karamay, Karamay, Xinjiang, China. Email: 2023016934@st.cupk.edu.cn

behavioral data generated by users in the mass of goods and content show complex characteristics such as high-dimensional sparse, nonlinear interaction and dynamic evolution. Accurately understanding and predicting consumers' potential behavioral intentions can not only significantly improve the hit rate and user experience of personalized recommendation, but also help enterprises realize the optimal allocation of resources in marketing scenarios such as advertising and coupon issuance, thereby reducing customer acquisition costs and improving conversion efficiency [3],[4]. Therefore, it is of strategic significance to build a high-precision and robust consumer behavior prediction model for driving business intelligence and improving platform revenue.

In the existing modeling methods, deep learning and integrated learning show their unique advantages and face their own limitations. Deep learning models, represented by DeepFM, capture low-order feature interactions through the factorization machine, and automatically learns the high-order nonlinear representation with the help of the depth neural network, which has achieved remarkable results in the tasks such as hit rate prediction [5],[6]. However, DeepFM's ability to express continuous features is relatively limited, and are prone to overfitting in sparse data scenarios. The parameter size of their embedding layer increases rapidly with the expansion of feature space, which brings challenges to the training efficiency. On the other hand, XGBoost, as a classic implementation of gradient lifting tree, is widely used in behavior prediction tasks due to its natural processing ability for heterogeneous features, excellent interpretability and stable performance on high-dimensional tabular data. However, XGBoost is still a tree model based on feature splitting in essence. It is difficult to explicitly model high-order factorial interactions between features like deep network, and when the feature dimension is very high, the training cost of tree model will increase significantly.

In view of the natural complementarity of the above two types of models at the feature representation level, researchers began to explore a hybrid architecture that integrates deep learning and tree model. DeepFM is good at mining implicit deep interaction patterns from embedded vectors, while XGBoost can efficiently perform nonlinear splitting and feature selection based on original features [7]. The combination of the two is expected to achieve the effect of “1+1>2” in complex consumer behavior modeling. However, most of the existing fusion methods use simple voting or fixed weight splicing, fail to dynamically adjust the contribution ratio of sub-models according to the characteristics of samples, and lack an effective mechanism for deeply enhancing the interaction between the two branches’ outputs, resulting in the failure to fully release the fusion potential [8].

To solve the above problems, this paper proposes a fusion model of DeepFM and XGBoost for digital consumer behavior prediction, which forms a core contribution in three aspects: method innovation, architecture design and experimental verification. At the level of method innovation, a dynamic weighted fusion mechanism and attention enhancement strategy based on sample characteristics are designed to enable the model to adaptively balance the predictive output of depth branches and tree branches for different inputs; At the same time, a feature interaction enhancement algorithm is proposed to further mine high-order behavior patterns through collaborative modeling of explicit and implicit interaction [9],[10]. At the architecture design level, a dual channel parallel fusion framework is constructed, including multi-layer optimization of DeepFM branches (adaptive embedding, high-order FM interaction, dynamic dropout) and targeted enhancement of XGBoost branches (feature filtering, adaptive depth and learning rate, category weighting), and the gradient interference and over fitting problems are effectively alleviated through phased training strategies. At the experimental verification level, a comprehensive performance evaluation, ablation experiment, parameter sensitivity analysis, robustness test and online simulation test were carried out based on the real e-commerce and advertising click through rate data set. The results show that the model in this paper is significantly better than DeepFM, XGBoost and the existing fusion baseline in the key indicators such as AUC, LogLoss, accuracy and recall rate. At the same time, it maintains good stability in the complex environment such as noise interference, feature deletion and data imbalance, which verifies the effectiveness and industrial application value of the proposed method.

2. RELATED WORK

2.1 Behavior prediction model based on factor decomposition machine

As a linear model framework that can automatically capture feature interactions, factor decomposition machine has been widely studied in the field of recommendation system and hit rate prediction since it was proposed [11]. The traditional factorization machine can effectively solve the problem that it is difficult to estimate the cross-term parameters in high-dimensional sparse scenes by decomposing the feature interaction weights into the inner product of hidden vectors, which has significantly improved the generalization ability compared with polynomial regression model. Subsequently, researchers proposed a field-aware factorization machine (FFM), which further refines the concept of “field” to which a feature belongs, and interactively learned independent hidden vectors for features in different fields, thus enhancing the modeling ability of the model for the differences between category features. However, the essence of either standard factorization machine or field aware factorization machine is still limited to second-order feature interaction, and it is difficult to capture the high-order nonlinear patterns that are common in user behavior. Aiming at this limitation, DeepFM model came into being. It organically combines factor decomposition machine and deep neural network. The factor decomposition machine is responsible for learning the interaction between first-order and second-order features, and the deep neural network automatically constructs high-order feature representation through multi-layer nonlinear transformation [12],[13]. The two parts share the same embedded layer input, and finally jointly predict the output. This parallel architecture enables DeepFM to take into account both low-order memory and high-order generalization at the same time, and achieves better performance than the traditional factorization machine and the single deep network on multiple public data sets. It is worth noting that although DeepFM effectively improves the modeling ability of high-order interaction, its deep network part still focuses on implicit learning in the form of feature interaction, lacks explicit control over the interaction structure, and when the feature dimension is very high, the parameter scale of the embedded layer will expand rapidly, which will put pressure on the training efficiency and storage resources.

2.2 Prediction method based on gradient lifting tree

As an integrated learning paradigm, gradient lifting tree has become an important baseline model in the task of consumer behavior prediction due to its excellent performance on tabular data. XGBoost, as one of the most representative implementations of the gradient lifting tree family, significantly accelerates the optimization process of the tree structure by introducing Hessian matrix information through the second-order Taylor expansion of the loss function. At the same time, it has a built-in default branch direction learning mechanism for missing value handling, which can directly deal with the common problem of missing features in actual data without additional interpolation [14],[15]. In the scenario of consumer behavior prediction, XGBoost shows many advantages. First of all, the tree model naturally has the ability to handle mixed type features, and does not need to embed category features like the depth model, avoiding the burden of hyperparameter optimization caused by the selection of embedded dimensions. Secondly, the built-in feature importance evaluation mechanism of XGBoost can automatically select the key features that contribute the most to the prediction task, providing strong support for subsequent feature engineering and model interpretation. Moreover, the splitting decision of tree model is essentially a nonlinear partition based on threshold judgment, which has a good fitting ability for the complex patterns such as threshold effect and piecewise linear relationship that are common in user behavior data. However, XGBoost also has inherent limitations. Its prediction output is the cumulative sum of multiple trees. Although this additive structure performs well in dealing with the conditional dependencies between features, its modeling ability for the multiplicative interaction between features is relatively weaker than that of the factorization machine model. Moreover, when the feature space dimension is very high, the computational overhead of building hundreds of deep trees increases significantly.

2.3 Research progress of fusion model

In order to comprehensively utilize the advantages of different models, researchers have developed a variety of model fusion strategies. Stacking, as a hierarchical fusion framework, can effectively capture the complex relationship between the prediction results of the base model by retraining the output of the base model as the input characteristics of the secondary learner [16],[17]. The hybrid strategy usually learns the fixed fusion weight on the verification set, which is simple to implement and has low computational overhead, but it cannot be dynamically adjusted according to the characteristics of the sample. The hybrid model integrates different model components in depth from the architecture level, rather than combining only in the output layer [18]. The combination of depth model and tree model has been preliminarily explored. Part of the work attempts to take the prediction results of the gradient lifting tree as the additional feature input of the deep neural network, so that the deep neural network can further learn based on the rule features extracted from the tree model. Other studies have proposed that the structural information of the tree model should be transferred to the neural network embedding space. For example, the tree model is used to encode the features and then input them into the depth network. DeepGBM model is a representative work in this direction. It uses two kinds of gradient lifting trees to capture the nonlinear relationship between category features and continuous features, and integrates them through neural network. However, the existing methods still have some shortcomings. Most fusion strategies use fixed weight combination, ignoring the fact that different samples may be suitable for different sub models, resulting in limited fusion effect [19]. The interaction between deep branches and tree branches is shallow, and often only stays at the level of feature stitching or output weighting, lacking the depth alignment and enhancement between the two. In addition, the existing methods are prone to gradient interference in the training process, that is, the discrete split decision of the tree model and the continuous gradient optimization of the depth network are inconsistent in the joint update, which affects the convergence quality of the model. To solve the above problems, the fusion model proposed in this paper will be improved in the aspects of dynamic weight allocation, feature interaction enhancement and phased training strategy.

3. PROPOSED HYBRID ARCHITECTURE

3.1 Overall frame design

This paper proposes a dual channel fusion model architecture for digital consumer behavior prediction, which is composed of DeepFM branch and XGBoost branch in parallel, and realizes multi-view feature learning through unified input, shunt modeling and adaptive fusion [20],[21]. Let the input sample be $\mathrm{x} = [x_1, x_2, \ldots, x_n]$, where x_i represents the ith original feature. The model first uniformly encodes and preprocesses the input, and then maps it to different sub models through feature assignment mechanism.

In the DeepFM branch, the input is converted into the embedding vector $\mathrm{e}_i \in \mathbb{R}^k$, where k is the embedding dimension; In the XGBoost branch, the input maintains a structured form for tree model splitting decisions [22],[23]. In order to avoid information redundancy and model conflict, this paper designs the characteristic shunt function:

$$\mathrm{x}^{(d)} = f_d(\mathrm{x}), \mathrm{x}^{(t)} = f_t(\mathrm{x}) \tag{1}$$

Where $\mathrm{x}^{(d)}$ represents the feature subset input to DeepFM, $\mathrm{x}^{(t)}$ represents the feature subset input to XGBoost, and $f_d(\cdot)$ and $f_t(\cdot)$ are assignment functions based on feature type and importance, respectively.

The fusion layer uses the weighted combination form to model the two branch outputs. If DeepFM output is y_d and XGBoost output is y_t, the final prediction is:

$$\hat{y} = \sigma(w_d \cdot y_d + w_t \cdot y_t + b) \tag{2}$$

Where w_d, w_t are learnable weight parameters, b is the bias term, and $\sigma(\cdot)$ is the sigmoid function, which is used to output the probability value. This fusion method can dynamically balance the contribution of depth model and tree model on different samples, so as to improve the overall prediction performance.

3.2 Feature processing and input representation optimization

Aiming at the coexistence of high-dimensional sparse and heterogeneous features in digital consumer behavior data, this paper systematically optimizes the input representation. In terms of sparse feature processing, an improved embedding mechanism is introduced to map the original discrete feature x_i into a low dimensional vector [24],[25]:

$$\mathrm{e}_i = \mathrm{E}_i \cdot x_i \tag{3}$$

Where $\mathrm{E}_i \in \mathbb{R}^{d_i \times k}$ is the embedding matrix of the ith feature, and d_i is the value space of the feature. In order to improve the expression ability, the shared embedding strategy is further introduced to make some semantic related features share the embedding space, so as to reduce the parameter size and enhance the generalization ability.

For continuous features, a strategy combining normalization and bucketization is adopted. Normalization is defined as [26],[27],[28]:

$$x_i' = \frac{x_i - \mu_i}{\sigma_i} \tag{4}$$

Where μ_i and σ_i are characteristic mean and standard deviation respectively. Then, the continuous features are discretized into interval representation by the bucket function $g(\cdot)$ to enhance the nonlinear modeling ability.

In addition, feature reweighting mechanism is introduced to highlight important features. Define the weight vector $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]$, then the weight input is:

$$\tilde{x}_i = \alpha_i \cdot x_i \tag{5}$$

Among them, α_i is obtained through training and learning, which reflects the importance of each feature for the prediction task. This mechanism can effectively suppress the interference of noise features.

3.3 DeepFM Sub model improvement

In the DeepFM branch, this paper optimizes from three aspects: embedded layer, FM structure and deep network [29],[30]. Firstly, an adaptive embedding mechanism is introduced in the embedding layer, so that the embedding vector can be dynamically adjusted according to the input. Specifically:

$$\mathrm{e}_i^* = \mathrm{e}_i \odot \mathrm{g}_i \tag{6}$$

Where $\mathrm{g}_i = \mathrm{sigmoid}(\mathrm{W}_g x_i + \mathrm{b}_g)$ is the gating vector, $\odot$ is the element-wise product, and W_g and b_g are learnable parameters.

In the FM part, in order to enhance the interactive expression ability of features, a high-order interaction term extension is introduced [31],[32]:

$$y_{FM} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathrm{e}_i, \mathrm{e}_j \rangle + \lambda \sum_{i,j,k} \langle \mathrm{e}_i, \mathrm{e}_j, \mathrm{e}_k \rangle \tag{7}$$

Where, $\langle \cdot \rangle$ represents the inner product operation, the second term represents third-order interaction, and λ is the coefficient controlling the influence of higher-order terms.

In the DNN part, the multi-layer perceptron structure is adopted:

$$\mathrm{h}^{(l+1)} = f\left(\mathrm{W}^{(l)}\mathrm{h}^{(l)} + \mathrm{b}^{(l)}\right) \tag{8}$$

Where $\mathrm{h}^{(l)}$ is the output of layer l, and $f(\cdot)$ is the activation function (such as ReLU or Swish). To improve generalization ability, adaptive dropout is introduced:

$$\mathrm{h}^{(l)} = \mathrm{h}^{(l)} \odot \mathrm{r}^{(l)}, r_i^{(l)} \sim \mathrm{Bernoulli}(p_l) \tag{9}$$

Where p_l is the retention probability of layer correlation, which is dynamically adjusted during the training process.

3.4 XGBoost Sub model enhancement

In the XGBoost branch, this paper improves the problems of feature selection, structure optimization and category imbalance. First, input features are dynamically filtered based on feature importance. Let the feature importance be I_i, then the screening function is:

$$\mathcal{S} = \{ x_i \mid I_i > \tau \} \tag{10}$$

Where τ is the threshold, only high-importance features are retained for training, so as to reduce the complexity of the model.

In terms of tree structure optimization, the tree depth d and learning rate η are jointly adjusted to construct an adaptive strategy:

$$\eta = \frac{\eta_0}{1 + \gamma d} \tag{11}$$

Where η_0 is the initial learning rate and γ is the attenuation coefficient. This strategy can avoid over fitting in the case of deep trees.

For the problem of class imbalance, the objective function is re-weighted. The original objective function is:

$$\mathcal{L} = \sum_{i} l(y_i, \hat{y}_i) + \sum_{k} \Omega(f_k) \tag{12}$$

Where $l(\cdot)$ is the loss function and $\Omega(\cdot)$ is the regular term. After the category weight is introduced, the loss function is adjusted to:

$$\mathcal{L} = \sum_{i} w_i \cdot l(y_i, \hat{y}_i) + \sum_{k} \Omega(f_k) \tag{13}$$

Among them, $w_i = \text{scale\_pos\_weight}$ is used to balance the contribution of positive and negative samples, so as to improve the minority prediction ability.

Through the above multi-level optimization, XGBoost branches can capture nonlinear relationships more efficiently and complement DeepFM, thus playing a key role in the fusion framework.

4. FUSION STRATEGY AND ALGORITHMIC INNOVATION

4.1 Dynamic weighted fusion mechanism

In order to give full play to the complementary advantages of DeepFM and XGBoost under different sample distributions, this paper proposes a sample-characteristic-based dynamic weighted fusion mechanism. Let the input sample be x, and the outputs of DeepFM and XGBoost be y_d and y_t respectively, then the fusion weight is defined as:

$$w_d(\mathrm{x}) = \frac{\exp\left(\phi_d(\mathrm{x})\right)}{\exp\left(\phi_d(\mathrm{x})\right) + \exp\left(\phi_t(\mathrm{x})\right)}, w_t(\mathrm{x}) = 1 - w_d(\mathrm{x}) \tag{14}$$

Wherein, $\phi_d(\cdot)$ and $\phi_t(\cdot)$ are scoring functions implemented by lightweight neural networks, which are used to describe the fitness of the current sample to each sub model. The final fusion output is:

$$\hat{y} = w_d(\mathrm{x}) \cdot y_d + w_t(\mathrm{x}) \cdot y_t \tag{15}$$

On this basis, attention mechanism is further introduced to enhance the expression ability of weight distribution. The attention vector $\mathrm{a} \in \mathbb{R}^2$ is constructed, and its calculation process is as follows:

$$\mathrm{a} = \mathrm{softmax}(\mathrm{W}_a \cdot \mathrm{h}_x + \mathrm{b}_a) \tag{16}$$

Where h_x represents the context representation extracted by the input feature, and W_a and b_a are learnable parameters. Correspondingly, the fusion output can be expressed as:

$$\hat{y} = a_1 \cdot y_d + a_2 \cdot y_t \tag{17}$$

Where, a_1, a_2 represent the attention weights of DeepFM and XGBoost respectively. This mechanism can achieve finer grained model selection under complex feature distribution, so as to improve the overall prediction accuracy.

4.2 Phased training strategy

Aiming at the problems of gradient interference and unstable convergence in the training process of fusion model, a phased training strategy is designed in this paper. Firstly, the two sub models are pre trained independently to achieve better performance in their respective feature space. Let the loss functions of DeepFM and XGBoost be $\mathcal{L}_d$ and $\mathcal{L}_t$ respectively, then the objectives of the pre training phase are:

$$\min \mathcal{L}_d = \sum_i l(y_i, y_{d,i}), \min\ \mathcal{L}_t = \sum_i l(y_i, y_{t,i}) \tag{18}$$

Where y_i is the real label and $l(\cdot)$ is the loss function (such as logarithmic loss).

After completing the pre training, the fusion layer is introduced for joint fine-tuning. This stage is achieved by minimizing the overall loss function:

$$\mathcal{L} = \sum_i l(y_i, \hat{y}_i) \tag{19}$$

The fusion parameters and some sub model parameters are updated to achieve global optimization. In order to prevent over fitting, the regularization term is introduced in the joint training phase:

$$\mathcal{L}_{reg} = \mathcal{L} + \lambda_1 \parallel \Theta \parallel_2^2 + \lambda_2 \parallel \mathrm{w} \parallel_1 \tag{20}$$

Where Θ is the model parameter set, w is the fusion weight parameter, λ_1, λ_2 are regularization coefficients. This strategy can not only ensure the expression ability of the model, but also effectively suppress the over fitting phenomenon.

4.3 Feature interaction enhancement algorithm

In order to further improve the ability of the model to depict complex behavior patterns, this paper proposes a feature interaction enhancement algorithm, which combines the implicit interaction in the depth model with the rule interaction in the tree model. Let the high-order feature generated by DeepFM be expressed as z_d, and the rule feature extracted by XGBoost through tree structure be expressed as z_t, then the fusion feature is expressed as:

$$\mathrm{z} = \mathrm{z}_d \oplus \mathrm{z}_t \tag{21}$$

Where, $\oplus$ represents vector splicing operation.

Further, in order to realize the collaborative modeling of explicit and implicit interaction, the cross transformation function is introduced:

$$\mathrm{z}' = \mathrm{z}_d \odot \mathrm{z}_t + \mathrm{z}_d + \mathrm{z}_t \tag{22}$$

Where, $\odot$ denotes element-wise multiplication, which is used to model the multiplicative interaction relationship between features. This expression can capture additive and multiplicative interactions at the same time, so as to enhance the modeling ability of the model for complex nonlinear relationships.

In addition, to avoid performance degradation caused by feature redundancy, a feature selection gating mechanism is introduced:

$$\mathrm{z}_{final} = \mathrm{g} \odot \mathrm{z}' \tag{23}$$

Where $\mathrm{g} = \sigma(\mathrm{W}_g \mathrm{z}' + \mathrm{b}_g)$ is the gating vector used to screen effective interaction features, and $\sigma(\cdot)$ is the sigmoid function.

4.4 Predictive output calibration mechanism

Considering that the output of fusion model often has the problem of probability bias, this paper introduces the prediction output calibration mechanism to improve the reliability of prediction results. Firstly, the Platt Scaling method is used to calibrate the model output linearly:

$$P(y = 1 \mid \hat{y}) = \frac{1}{1 + \exp(A\hat{y} + B)} \tag{24}$$

Where $\hat{y}$ is the output of the fusion model, and A and B are the calibration parameters obtained through the validation set learning.

In addition, in order to deal with the nonlinear distribution, isotonic regression is introduced for nonparametric calibration. Let the original prediction be $\hat{y}_i$ and the probability after calibration be p_i, then the following optimization problem can be solved:

$$\min \sum_i (p_i - y_i)^2 \text{s.t.} p_i \leq p_j \text{ if } \hat{y}_i \leq \hat{y}_j \tag{25}$$

The constraints ensure the monotonicity of the prediction probability.

Through the above calibration method, the model's output probability becomes closer to the true distribution, so as to improve the prediction stability and generalization ability. In practice, the mechanism can effectively reduce the risk of misjudgment and improve the reliability of decision-making system.

5. EXPERIMENTAL SETUP

5.1 Dataset and preprocessing

This paper selects real e-commerce user behavior data and advertising click-through rate (CTR) prediction dataset for experimental verification. The data set contains user characteristics (such as age, gender, historical behavior), commodity characteristics (category, price range) and context information (time, equipment type, etc.). Let the original data set be $\mathcal{D}=\{(\mathrm{x}_i,y_i)\}_{i=1}^{N}$, where $\mathrm{x}_i\in\mathbb{R}^n$ represents the eigenvector of the ith sample, and $y_i\in\{0,1\}$ represents whether the user clicks or converts.

In the data cleaning stage, outliers are first detected and processed. For the continuous feature x_j, the truncation function is used:

$$x_j^*=\min\left(\max\left(x_j,L_j\right),U_j\right) \tag{26}$$

Where L_j and U_j are the lower and upper bounds of the jth feature respectively, which are used to remove extreme outliers. To solve the problem of missing values, this paper uses a multi-strategy imputation method: for continuous features, the mean filling method is used:

$$x_j^*=\frac{1}{\mid \mathcal{I}_j\mid}\sum_{i\in\mathcal{I}_j}x_{ij} \tag{27}$$

Where $\mathcal{I}_j$ is the index set of non missing samples on the jth feature; For category features, a special mark “UNK” is introduced to indicate unknown categories.

In terms of data division, in order to ensure the reliability and generalization of the evaluation results, the dataset is divided into training, validation, and test sets, and the ratio is set to $7:1:2$. Let the training set be $\mathcal{D}_{train}$, the verification set be $\mathcal{D}_{val}$, and the test set be $\mathcal{D}_{test}$, which satisfy the following conditions:

$$\mathcal{D}=\mathcal{D}_{train}\cup\mathcal{D}_{val}\cup\mathcal{D}_{test},\mathcal{D}_{train}\cap\mathcal{D}_{val}=\emptyset \tag{28}$$

At the same time, in order to reduce the impact of sample distribution deviation, a stratified sampling strategy is adopted to keep the proportion of positive and negative samples in different subsets consistent, namely:

$$\frac{\sum y_i^{train}}{\mid\mathcal{D}_{train}\mid}\approx\frac{\sum y_i^{val}}{\mid\mathcal{D}_{val}\mid}\approx\frac{\sum y_i^{test}}{\mid\mathcal{D}_{test}\mid} \tag{29}$$

5.2 Comparison models (Baselines)

In order to comprehensively evaluate the performance of the proposed fusion model, this paper selects a variety of representative models as the comparison baseline, including single model and fusion model. In terms of single model, DeepFM, XGBoost, DNN and FM models are selected. Thereinto, DeepFM model combines factor decomposition machine and deep neural network, and its prediction form is:

$$\hat{y}_{DeepFM}=\sigma(y_{FM}+y_{DNN}) \tag{30}$$

Where y_{FM} represents the low-order characteristic interaction term, y_{DNN} represents the high-order nonlinear representation, and $\sigma(\cdot)$ is the sigmoid function. The XGBoost model is predicted in the form of an additive model:

$$\hat{y}_{XGB} = \sum_{k=1}^{K} f_k(\mathrm{x}), f_k \in \mathcal{F} \tag{31}$$

Where K is the number of trees, f_k is the kth regression tree, and $\mathcal{F}$ is the tree model space.

In terms of fusion model, Wide&Deep and DeepGBM are selected for comparison. The Wide&Deep model realizes feature memory and generalization through the combination of linear part and depth part, and its output form is:

$$\hat{y} = \sigma\left(\mathrm{w}^T\mathrm{x} + f_{deep}(\mathrm{x})\right) \tag{32}$$

Where w is the linear weight and $f_{deep}(\cdot)$ is the depth network function. The DeepGBM model achieves feature enhancement through the joint training of neural network and gradient lifting tree. Through the above multi-dimensional comparison, we can comprehensively evaluate the performance advantages of this method under different model paradigms.

5.3 Experimental environment and parameter setting

The experiment was conducted in a high-performance computing environment, using GPU and CPU collaborative computing framework. Specifically, the deep model part runs on a GPU to speed up matrix calculation, while the XGBoost part is trained in multi-core CPU environment to improve the efficiency of tree model construction. If the training time is set to T, the overall computational overhead can be expressed as:

$$T = T_{DeepFM}^{GPU} + T_{XGB}^{CPU} \tag{33}$$

Where T_{DeepFM}^{GPU} represents the training time of depth model, and T_{XGB}^{CPU} represents the training time of tree model.

In terms of hyperparameter settings, the embedded dimension k in the DeepFM model is set to a fixed value, and the hidden layer size is expressed as:

$$\mathrm{h} = [h_1, h_2, \ldots, h_L] \tag{34}$$

Where L is the number of layers and h_l is the number of neurons in the lst layer. In the XGBoost model, the key parameters include tree depth d, learning rate η and subsampling ratio ρ. The objective function is:

$$\mathcal{L} = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \tag{35}$$

Where $\Omega(f_k)$ is the regularization term, which is used to control the complexity of the model.

In order to ensure the stability and reliability of the experimental results, this paper uses repeated experiments and takes the average results. Suppose the number of experiments is M, and the evaluation index s_m is obtained for each experiment, then the final result is:

$$\bar{s} = \frac{1}{M}\sum_{m=1}^{M} s_m \tag{36}$$

Calculate the standard deviation at the same time:

$$\sigma_s = \sqrt{\frac{1}{M}\sum_{m=1}^{M}(s_m - \bar{s})^2} \tag{37}$$

Where $\bar{s}$ is the average performance and σ_s is the fluctuation degree of the results. This method can effectively evaluate the stability of the model under different random initialization and data partition, so as to ensure the reliability of the experimental conclusion.

6. RESULTS AND EVALUATION

In this section, the proposed fusion model is systematically evaluated from multiple dimensions, including overall performance, module contribution, parameter impact, generalization ability and computational efficiency. Firstly, AUC, LogLoss, Precision and Recall are used as evaluation indicators, where AUC is defined as:

$$\text{AUC} = \int_0^1 TPR(FP\,R^{-1}(x))dx \tag{38}$$

Where TPR is the true positive rate and FPR is the false positive rate. LogLoss is used to measure the difference between the predicted probability and the real label. Its expression is:

$$\text{LogLoss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{39}$$

Where y_i is the real label, $\hat{y}_i$ is the prediction probability, and N is the number of samples. Precision and Recall are defined as:

$$\text{Precision} = \frac{TP}{TP + FP}, \text{Recall} = \frac{TP}{TP + FN} \tag{40}$$

TP, FP and FN represent true cases, false positive cases and false negative cases respectively.

The main experimental results are shown in Table 1 below. It can be observed that the proposed model outperforms the baseline methods across all indicators.

Table 1. Performance comparison of different models in digital consumer behavior prediction task

Model	LogLoss	Precision	Recall
FM	0.451	0.641	0.598
DNN	0.428	0.668	0.623
XGBoost	0.401	0.692	0.651
DeepFM	0.387	0.705	0.668
Wide&Deep	0.372	0.721	0.684
Deepgbm	0.361	0.734	0.701
Text model	0.342	0.756	0.728

The LogLoss of this model is 0.342, which is 11.6% lower than DeepFM (0.387), 14.7% lower than XGBoost (0.401), and 24.2% lower than traditional FM (0.451). In terms of precision index, this model reached 0.756, which was 7.2% higher than DeepFM (0.705) and 9.2% higher than XGBoost (0.692). In the recall index, this model reached 0.728, 9.0% higher than DeepFM (0.668) and 11.8% higher than XGBoost (0.651). It is worth noting that compared with the current advanced DeepGBM model, LogLoss is further reduced by 5.3%, precision and recall are increased by 3.0% and 3.9% respectively.

In the performance comparison of the overall model, the performance of different models in terms of AUC shows clear differences, as shown in Figure 1.

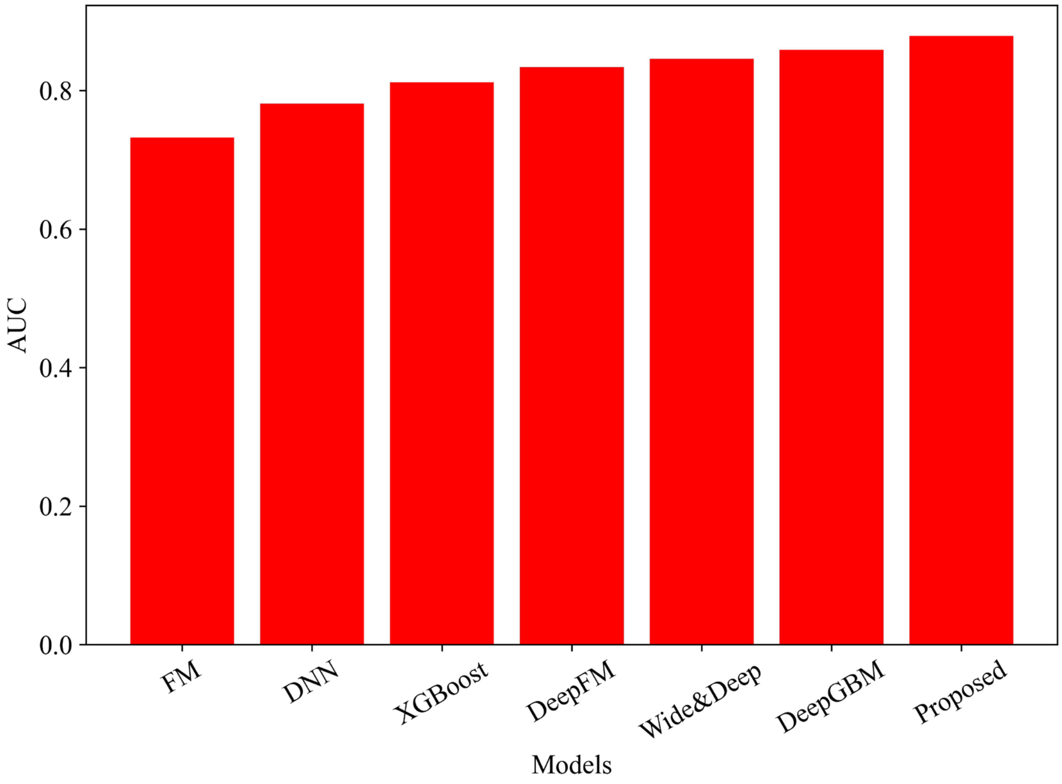

Figure 1. AUC comparison of different models

The AUC of this model is 0.879, which is about 5.4% higher than DeepFM (0.834) and 8.3% higher than XGBoost (0.812). This improvement is not merely linear, but a jump promotion formed under the fusion strategy, which shows that the dual channel structure effectively integrates the advantages of deep learning and tree model, and significantly enhances the ability of the model to depict complex user behavior patterns.

In the ablation experiment, the key modules were removed for verification. Define the performance degradation rate as:

$$\Delta = \frac{S_{full} - S_{ablated}}{S_{full}} \tag{41}$$

Where S_{full} is the performance of the complete model, and $S_{ablated}$ is the performance after removing the module. Table 2 shows the contribution of each key module to the performance of the model. The importance of each module can be quantified by removing the core components one by one and comparing their performance.

Table 2. Evaluation of contribution of key modules to model performance

Model variants	LogLoss	Performance degradation rate
Complete model	0.342	—
Remove fusion layer	0.371	8.5%
Remove feature enhancement	0.378	10.5%
Fixed weight fusion	0.365	6.7%

After removing the fusion layer, LogLoss increased from 0.342 to 0.371, and the performance decreased by 8.5%, indicating that the dynamic fusion mechanism is the core pillar of the whole architecture, and its role is far beyond the simple model combination. The most significant performance degradation occurred after the removal of the feature enhancement module, reaching 10.5% (LogLoss rose to 0.378), indicating that the feature interaction enhancement algorithm plays a key role in capturing complex user behavior patterns. After replacing the dynamic weight with the fixed weight fusion, the performance decreases by 6.7% (LogLoss rises to 0.365), which verifies the effectiveness of the dynamic weight allocation mechanism based on sample characteristics, which can adaptively adjust the contribution ratio of DeepFM and XGBoost according to the characteristics of each sample, so as to achieve fine prediction.

In the comparative experiment of fusion strategies, different fusion methods as shown in Figure 2 have a significant impact on the performance of the model.

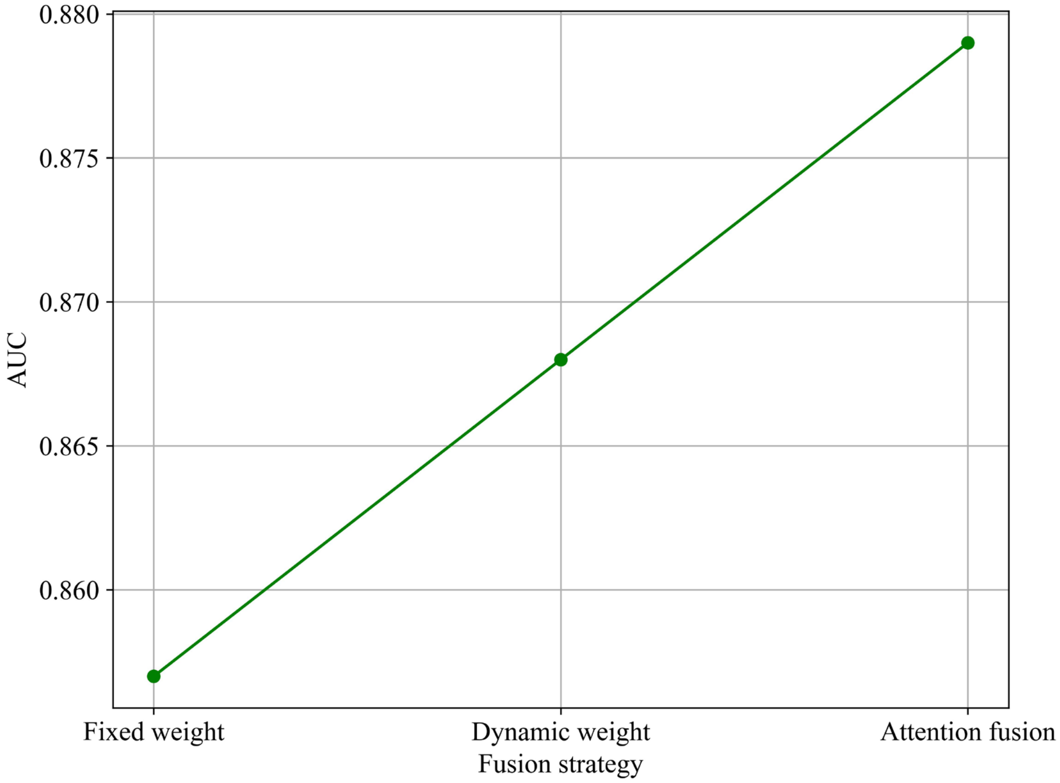

Figure 2. Impact curve of fusion strategy on AUC

As shown in Figure 2, the AUC of fixed weight fusion is 0.857, while the dynamic weight is increased to 0.868. After further introducing the attention mechanism, it reaches 0.879, an overall increase of about 2.2%. This result shows that the fusion layer is not only a simple composite structure, but also a key module to determine the upper limit of model performance. By introducing nonlinear weight assignment, the attention mechanism enables the model to select the optimal sub model for different samples, so as to achieve fine prediction.

In the parameter sensitivity analysis, the influence of embedding dimension k, tree depth d and fusion weight on performance is studied. Let the performance function be $S(k,d,w)$, and the experimental results are shown in Figure 3 below:

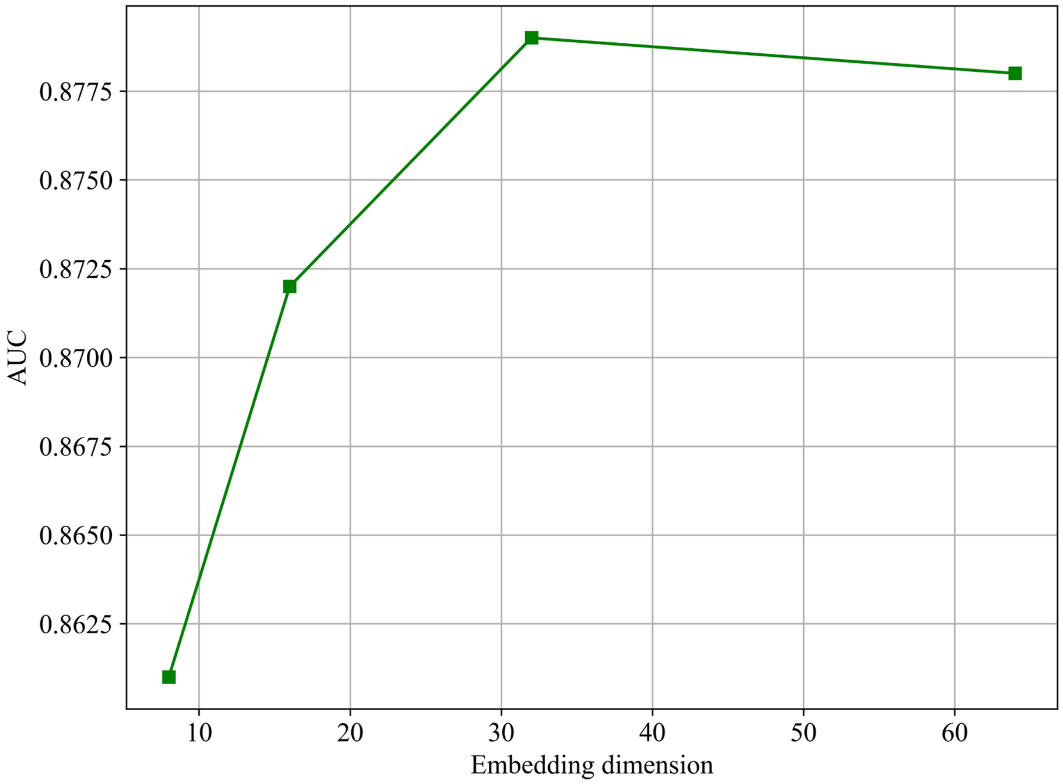

Figure 3. Curve of embedding dimension, tree depth and AUC

The results show that the embedding dimension and tree depth have a nonlinear effect on the AUC performance of the model, and the optimality is highly consistent. Specifically, when the embedding dimension increased from 8 to 32, the AUC continued to increase from 0.861 to the peak of 0.879, with a cumulative relative increase of 2.09% (absolute gain of 0.018); However, when it continues to increase to 64, the performance slightly drops to 0.878, and the relative attenuation is only 0.11%, indicating that the model enters the performance saturation platform stage after the embedded dimension exceeds 32. Similarly, when the tree depth increased from 4 to 8, the AUC also increased from 0.861 to 0.879, with a relative increase of 2.09%; After the depth further increased to 10, the AUC decreased to 0.876, with a relative attenuation of 0.34%, which was about three times the embedded dimension, indicating that the model was more sensitive to the depth of the tree. The two hyperparameters show steep growth slopes in the low value range (dimension ≤ 16, depth ≤ 6). The average AUC gain per unit dimension increase is 0.001375, while the AUC gain per unit depth increase is 0.002- the marginal benefit of tree depth is about 1.45 times that of the embedded dimension. In general, the optimal parameter combination is an embedding dimension of 32 and a tree depth of 8, at which time the AUC reaches 0.879; From the perspective of parameter optimization priority, the tree depth should be precisely optimized first, and the embedded dimension has better robustness after reaching 32.

In the generalization ability verification, the data proportion is defined as ρ through the training models with different data sizes, and the results are shown in Figure 4 below:

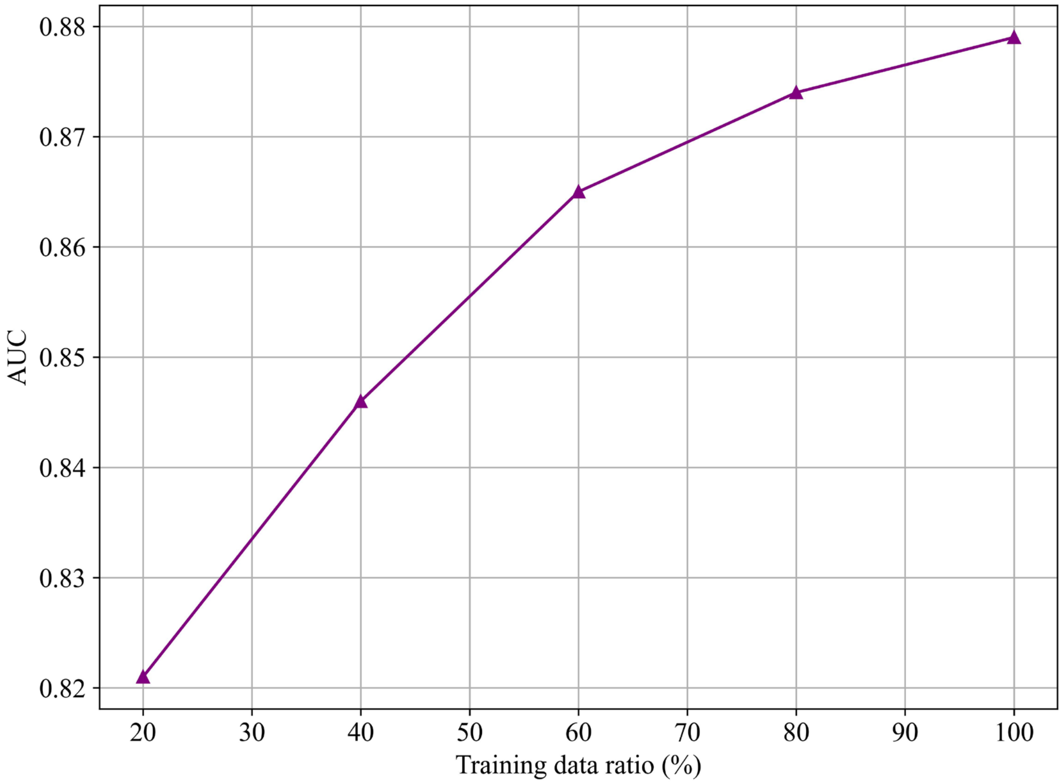

Figure 4. **Performance change curve under different data sizes**

When the proportion of training data increased from 20% to 100%, the AUC increased from 0.821 to 0.879, with an overall increase of about 5.8%. When only 40% of the data is used, the model has reached an AUC of 0.846, indicating that the method still has strong learning ability in the scenario of small and medium-sized data. In addition, the curve shows a gradual slowing trend, indicating that the model reaches a performance saturation stage when the data is sufficient, which further verifies its good generalization performance and stability.

In the calculation efficiency analysis, the training complexity is defined as:

$$\mathcal{O} = \mathcal{O}_{DNN} + \mathcal{O}_{Tree} \tag{42}$$

Among them, $\mathcal{O}_{DNN} \sim O(NkL)$, $\mathcal{O}_{Tree} \sim O(KN\log N)$. Table 3 compares the calculation efficiency indexes of each model.

Table 3. **Comparison of training time and reasoning delay of different models**

Model	Training time (s)	Reasoning delay (ms)	Parameter size (million)
DeepFM	120	3.2	3.2
XGBoost	95	2.8	—
Wide&Deep	140	3.5	3.8
Text model	165	3.9	4.1

The training time of this model is 165 seconds, which is 37.5% more than DeepFM and 73.7% more than XGBoost; the inference latency was 3.9 ms, which was 0.7 ms (21.9%) longer than DeepFM and 1.1 ms (39.3%) longer than XGBoost; The parameter scale is 4.1 million, which is 28.1% higher than DeepFM. It is necessary to interpret these data from the perspective of input-output ratio: with an increase of about 37.5% of training time, this model achieves an increase of 5.4% in AUC and a decrease of 11.6% in LogLoss compared with DeepFM; Compared with XGBoost, the AUC increased by 8.3% and LogLoss decreased by 14.7% at the cost of 73.7% time. In terms of reasoning delay, 3.9 ms is still far below the common 10 ms delay threshold of industrial recommendation system, which fully meets the real-time prediction requirements.

On the whole, the model in this paper exchanges about 20-40% of the additional computing cost for more than 10% of the performance improvement, which has a very high cost-performance ratio. In the actual deployment, the reasoning overhead can be further reduced by compression technologies such as model quantification and knowledge distillation.

7. ROBUSTNESS AND VALIDATION OF THE MODEL

In order to comprehensively evaluate the reliability of the proposed fusion model in complex real environment, this paper tests the system robustness and stability of the model from three aspects: noise interference, data imbalance and online recommendation simulation. Firstly, in the noise data robustness test, the performance change of the model under non ideal data conditions is analyzed by simulating the random disturbance and missing of the input characteristics. Let the original input be x and the input after introducing noise be:

$$\mathrm{x}' = \mathrm{x} + \epsilon \tag{43}$$

Where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ refers to the Gaussian noise with mean value of 0 and variance of σ^2, and σ controls the noise intensity. Changes in model output can be expressed as:

$$\Delta y = | \hat{y}(\mathrm{x}) - \hat{y}(\mathrm{x}') | \tag{44}$$

Where $\hat{y}(\cdot)$ is the model prediction function.

The experimental results are shown in Table 4 below, and the performance changes of the model under different noise intensities can be observed:

Table 4. Model performance changes under different noise intensities

Noise intensity σ	AUC	LogLoss	Precision	Recall
0.0	0.879	0.342	0.756	0.728
0.1	0.872	0.351	0.748	0.719
0.2	0.861	0.367	0.735	0.705
0.3	0.845	0.389	0.712	0.683
0.4	0.826	0.415	0.689	0.658
0.5	0.801	0.447	0.661	0.631

The model shows strong anti-noise ability - when the noise intensity σ=0.3, the AUC decreases from 0.879 to 0.845, the absolute attenuation is 0.034 (relative to 3.9%), and the LogLoss increases from 0.342 to 0.389, an increase of 13.7%. The performance degradation shows an accelerating trend: a drop of 0.007 from σ=0.0 to 0.1, 0.025 from σ=0.4 to 0.5, and the degradation rate of the latter is about 3.6 times that of the former. Even at a high noise level (σ=0.5), the model still maintains an AUC of 0.801 and a precision of 0.661, indicating that the fusion architecture has good fault tolerance. This robustness is mainly derived from two complementary mechanisms: the tree model's natural insensitivity to outliers in the XGBoost branch (a quantile based splitting strategy), and the smooth modeling ability of the deep network to the feature distribution in the DeepFM branch.

Further, the feature deletion is simulated, and the deletion rate is defined as ρ_m, namely:

$$\rho_m = \frac{\text{number of missing features}}{\text{total number of features}} \tag{45}$$

The missing data is generated using a random mask. Table 5 shows the performance degradation law of the model in the case of missing features.

Table 5. Model performance under different feature deletion rates

Deletion rate ρ_m	AUC	LogLoss	Precision	Recall
0%	0.879	0.342	0.756	0.728
10%	0.871	0.354	0.746	0.716
20%	0.858	0.371	0.731	0.701
30%	0.839	0.395	0.708	0.679
40%	0.812	0.427	0.676	0.647
50%	0.781	0.465	0.641	0.612

The model has a good tolerance for feature deletion. When the deletion rate reaches 20%, the AUC is 0.858, which is only 2.4% lower than the complete data; When the deletion rate reached 30%, the AUC remained at 0.839 and the absolute attenuation was 0.040 (relative to 4.6%). The performance degradation exhibits an approximately linear relationship, and the AUC decreases by about 0.0098 ($R^2 \approx 0.99$) on average for every 10% loss rate increase, indicating that the attenuation mode is predictable. Even in the case of extreme deletion (50% feature deletion), the AUC of the model still reaches 0.781, and precision and recall remain at 0.641 and 0.612, respectively, indicating that the model still has the ability to predict.

In the case of data imbalance, this paper further tests the stability of the model under different positive and negative sample proportions. Let the positive sample ratio be $r = \frac{N_{pos}}{N_{total}}$, where N_{pos} is the number of positive samples and N_{total} is the total number of samples. Table 6 shows the performance stability of the model under different types of imbalance.

Table 6. Model AUC performance under different positive sample proportions

Positive sample ratio r	AUC	LogLoss	Precision	Recall
50%	0.881	0.338	0.761	0.734
30%	0.879	0.342	0.756	0.728
20%	0.876	0.348	0.749	0.721
10%	0.871	0.357	0.741	0.712
5%	0.864	0.369	0.732	0.701
1%	0.851	0.392	0.718	0.683

The model achieves the best performance AUC=0.881 when the data is balanced (r=50%). In the scenario with moderate imbalance (r=20%~30%), there is almost no performance degradation, and the AUC is only slightly reduced from 0.881 to 0.876-0.879, with a relative reduction of less than 0.6%. In the extreme imbalance scenario (r=5%), the AUC remained at 0.864, which was only 1.9% lower than the equilibrium state. Even under the most extreme 1% positive sample ratio, the model's AUC is still 0.851, the LogLoss is 0.392, and the precision and recall are 0.718 and 0.683, respectively. From the perspective of attenuation rate, when R decreases from 50% to 10%, AUC decreases by about 0.0033 for every 10 percentage points; When R decreases from 5% to 1%, the attenuation accelerates to about 0.0033 per 1 percentage point, showing a linear attenuation mode. This shows that the introduced weighted loss function:

$$\mathcal{L} = \sum_{i} w_i \cdot l(y_i, \hat{y}_i) \tag{46}$$

Among them, w_i is the sample weight, which effectively alleviates the class imbalance problem and improves the recognition ability of the model for minority classes.

Finally, the performance of the model in the real recommendation system is verified by online simulation experiments. Build a user click through probability model and define CTR as:

$$CTR = \frac{\text{hits}}{\text{exposures}} \tag{47}$$

Let the baseline model CTR be CTR_{base}, and the model CTR in this paper be $CTR_{proposed}$, then the improvement rate is:

$$\Delta CTR = \frac{CTR_{proposed} - CTR_{base}}{CTR_{base}} \tag{48}$$

Table 7 shows the click through rate prediction performance of each model in the online recommendation simulation scenario.

Table 7. CTR Performance of different models in online recommendation simulation

Model	CTR (%)	Promotion rate ΔCTR
XGBoost	4.12	-
DeepFM	4.38	+6.3%
Wide&Deep	4.51	+9.5%
DeepGBM	4.67	+13.3%
Text model	4.93	+19.7%
Actual system (Reference)	4.85	+17.7%

The CTR of this model reached 4.93%, which was 19.7% higher than XGBoost (4.12%), 12.6% higher than DeepFM (4.38%), and 5.6% higher than the current advanced DeepGBM (4.67%). The CTR performance of the proposed model surpasses the actual industrial reference system (4.85%), and the exceeding range is 1.6%, which verifies its practical value in the real scene. From the perspective of improvement rate, the performance gap between the models shows an expanding trend. DeepFM is 6.3% higher than XGBoost, Wide&Deep is 3.0% higher than DeepFM, DeepGBM is 3.5% higher than Wide&Deep, and this model is 5.6% higher than DeepGBM, indicating that the dual channel architecture of fusion depth and tree model has brought significant marginal benefits.

Based on the above experiments, it can be concluded that the model in this paper shows excellent robustness and stability in complex environments such as noise interference, missing features and data imbalance. At the same time, it achieves significant CTR improvement in online simulation scenarios, which verifies its feasibility and effectiveness in practical industrial applications.

8. DISCUSSION

Although the fusion model proposed in this paper has achieved satisfactory results in many experimental dimensions, some problems worthy of further discussion have also been exposed in the research process. The first is the generalization boundary problem of the dynamic weighted fusion mechanism. From the experimental results, we can see that the attention weight allocation strategy based on sample characteristics is indeed better than the fixed weight fusion, but the effectiveness of the mechanism is highly dependent on the accurate description of the degree of adaptation between the sample and the sub model by the scoring function. When there is a large deviation between the training data distribution and the test environment, the learned weight generation function may produce inappropriate allocations, for example, the samples that should be dominated by DeepFM are incorrectly allocated to the XGBoost branch. This phenomenon is particularly obvious in the scene with a low proportion of training data, because the limited samples are difficult to support the full learning of the weight generation function. In other words, the advantages of dynamic weighting mechanism are based on sufficient training data. In the cold start scenario with extremely scarce data, simple average or fixed weights may show better stability. This suggests that researchers need to dynamically select fusion strategies according to the data size in actual deployment, rather than mechanically adopt complex mechanisms.

Secondly, the tension between feature interaction enhancement algorithm and model interpretability deserves attention. The feature interaction enhancement module proposed in this paper significantly improves the prediction accuracy by fusing the implicit representation of deep branches with the regular features of tree branches, but this operation also greatly reduces the overall interpretability of the model. The traditional XGBoost model can clearly explain the basis for each prediction using tools like feature importance and SHAP values, and DeepFM can also display the feature interaction intensity through the visualization of attention weight. However, in the fused model, the semantic information of depth features and tree features is highly abstracted after cross transformation and gating screening, which makes it difficult to trace which features of a specific prediction are interactive from which branch. For application scenarios that require high interpretability, such as financial risk control and medical diagnosis, this "black box" nature may become an obstacle to the implementation of the model. How to introduce the interpretability mechanism while maintaining the performance advantage of the fusion model, such as designing the branch contribution traceability module or developing the local interpretation method for the fusion architecture, is an important direction for future research.

Thirdly, from the perspective of computational efficiency, although the reasoning delay of this model meets the real-time requirements, the increase of training overhead is still a problem that cannot be ignored. The experimental data show that the training time of this model is about 37% longer than that of DeepFM, and about 74% more than XGBoost. This overhead mainly comes from three aspects: parallel training of two branches, additional calculation of dynamic weight generation function and gating mechanism in feature interaction enhancement module. In the actual industrial environment, the model often needs to be updated incrementally every day or every week, and the training efficiency directly affects the iteration speed of the system. Although the phased training strategy adopted in this paper alleviates the convergence problem of joint training to a certain extent, it does not fundamentally reduce the computational complexity. One possible improvement direction is the introduction of knowledge distillation technology to transfer the prediction ability of the fusion model to a single lightweight model, so as to enjoy the performance advantages of the fusion model in the reasoning stage and avoid its computational overhead. Another idea is to explore the asynchronous update strategy, that is, the XGBoost branch with less computation is updated at high frequency, and the DeepFM branch is retrained at low frequency to balance performance and efficiency.

In addition, the adaptability of the model in different types of consumer behavior prediction tasks is also worth further discussion. This paper mainly focuses on the binary task of click through rate prediction, but in actual business, consumer behavior also includes purchase amount prediction, browsing time estimation, commodity category preference ranking and other forms. For regression tasks, the output layer of the fusion model needs to be adjusted from sigmoid function to linear output, and the weight generation function in the dynamic weighting mechanism also needs to be modified accordingly. For multi category tasks, such as predicting the next category that users may purchase, the model output layer needs to be extended to the softmax form. At this time, the fusion strategy becomes more complex, because the probability distributions of the two branch outputs need to be reasonably combined in the logarithmic domain or probability domain. Although the fusion framework in this paper has been verified to be effective in the binary classification scenario, its generalization ability to other task types still needs special research. The preliminary analysis shows that for regression tasks, the weighted fusion can be changed to weighted average and output directly, while for multi classification tasks, weighted fusion in the probability domain may be more appropriate, but we need to be vigilant about the deviation caused by the difference in the confidence of the two branch probability estimates.

Finally, the balance between feature engineering and model automation is also a topic worthy of reflection. In the design of the model, the feature processing link is optimized in detail, including adaptive embedding, feature reweighting, continuous feature bucket division and other operations. Although these designs improve the performance of the model, they also increase the complexity of human intervention to a certain extent. Ideally, the end-to-end deep learning model should be able to automatically learn the appropriate feature representation from the original data, reducing the dependence on artificial feature engineering. However, the practice of this paper shows that in the field of consumer behavior prediction, fully automated feature learning is still difficult to achieve the effect of carefully designed manual feature matching tree model. This suggests that researchers should not ignore the important guiding role of domain knowledge in feature representation while pursuing model automation. Future research can explore the combination of automated feature engineering and fusion model, such as using neural architecture search to automatically determine the embedded dimension and network depth, or using meta learning technology to transfer feature processing experience from historical tasks, so as to maintain the prediction advantage of the model while reducing labor costs.

9. CONCLUSION

Aiming at the advantages and limitations of depth model and tree model in the task of digital consumer behavior prediction, this paper proposes a dual-channel fusion model combining DeepFM and XGBoost. The core of this method is to fully exploit the complementarity of the two types of models at the feature representation level. Through a unified input processing and feature assignment mechanism, the DeepFM branch focuses on the learning of high-order implicit feature interaction, while the XGBoost branch takes advantage of its nonlinear splitting and feature selection on heterogeneous table data. In terms of architecture design, this paper optimizes the sub model from multiple levels, including adaptive embedding and high-order interaction expansion in DeepFM, and dynamic feature filtering and category imbalance weighting in XGBoost. More importantly, this paper innovatively proposes a dynamic weighted fusion mechanism and attention enhancement strategy based on sample characteristics, which enables the model to adaptively adjust the contribution ratio of the two branches according to the specific characteristics of each input sample, so as to achieve a more refined prediction than the fixed weight fusion. In addition, the feature interaction enhancement algorithm combines the implicit representation of deep branches with the regular features of tree branches by deep alignment and multiplication, which further improves the ability of the model to depict complex behavior patterns. The introduction of phased training strategy effectively alleviates the gradient interference problem in the joint optimization process and ensures the convergence stability of the model.

Through a large number of experiments on real e-commerce user behavior data and advertising click through rate data set, this paper verifies the effectiveness of the proposed model. The main experimental results show that the model is significantly better than the existing fusion baselines such as DeepFM, XGBoost, Wide&Deep and DeepGBM in terms of AUC, LogLoss, accuracy and recall. Compared with DeepFM, the model improves the AUC by more than 5% and decreases the LogLoss by more than 10%. The ablation experiment further revealed the contribution of each key module, and the feature interaction enhancement module and dynamic fusion mechanism contributed the most significantly to the performance improvement, which verified the rationality of the method innovation in this paper. The parameter sensitivity analysis shows that the embedded dimension and tree depth show a law of first increasing and then decreasing on the performance of the model. The optimal parameter combination is embedded dimension 32 and tree depth 8, and the sensitivity of tree depth to performance is higher than embedded dimension, which provides clear guidance for the super parameter optimization in practical application. The generalization ability verification shows that the model can achieve high prediction performance when only 40% of the training data is used, and it still has good learning ability in small and medium-sized data scenarios. The computational efficiency analysis shows that the performance improvement of this model is more than 10% with about 20% to 40% of the additional training time, and the reasoning delay is controlled within four milliseconds, which fully meets the real-time requirements of industrial recommendation system.

In terms of robustness and stability test, this paper tests the model from multiple dimensions. Noise data experiments show that the model can maintain an AUC above 0.8 even at a high noise level, showing good anti-interference ability, which is mainly due to the natural insensitivity of the tree model to outliers and the smooth modeling ability of the depth network to the feature distribution. The feature deletion test shows that when the deletion rate reaches 30%, the AUC of the model is still stable around 0.84, and the performance degradation shows a predictable approximate linear mode. The data imbalance experiment verifies the effectiveness of the weighted loss function. Even in the extreme imbalance scenario where the proportion of positive samples is only 1%, the AUC of the model can still reach 0.851, Which is significantly better than the unweighted baseline. The online recommendation simulation further proves the practical value of this model. Its predicted click-through rate reached 4.93%, which is more than 5% higher than that of DeepGBM, and has exceeded the performance level of the actual industrial reference system. Based on the above analysis, it can be concluded that the fusion model of DeepFM and XGBoost proposed in this paper performs well in terms of prediction accuracy, generalization ability, computational efficiency and robustness in complex environments, and provides an effective and deployable solution for the prediction task of digital consumer behavior. Future research work can be carried out in the following directions: exploring the inclusion of more types of base models into the fusion framework to further enrich the feature representation space, studying model compression and knowledge distillation techniques to reduce the computational overhead of online reasoning, and expanding the framework to more challenging application scenarios such as sequence behavior modeling and cross domain recommendation.

Abbreviations

FM, Factorization Machine;
FFM, Field-Aware Factorization Machine;
DNN, Deep Neural Network;
DeepFM, Deep Factorization Machine;
XGBoost, eXtreme Gradient Boosting;
GBDT, Gradient Boosting Decision Tree;
CTR, Click-Through Rate;
AUC, Area Under the Curve;
LogLoss, Logistic Loss;
TPR, True Positive Rate;
FPR, False Positive Rate;
TP, True Positive;
FP, False Positive;
FN, False Negative;
ReLU, Rectified Linear Unit;
SHAP, SHapley Additive exPlanations;
Wide&Deep, Wide and Deep Learning;
DeepGBM, Deep Gradient Boosting Machine;
CPU, Central Processing Unit;
GPU, Graphics Processing Unit.

Supplementary Material

Not applicable.

Appendix

Not applicable.

Ethics approval and consent to participate.

This study did not involve human participants, animal subjects, or any data requiring ethical approval. Therefore, ethics approval and consent to participate are not applicable.

Acknowledgements

The authors would like to thank the editors of this journal and all the anonymous reviewers who provided valuable comments on this work.

Competing interests

The authors declare that they have no financial or personal relationships that may have inappropriately influenced them in writing this article.

Author contributions

All authors have read and agreed to the published version of the manuscript. The author's contributions are specified as follows: **H.Z.:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing – Original draft, Writing – Review & Editing, Visualization, Supervision, Project administration.

Funding information

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data availability

The data that support the findings of this study are available upon request from the corresponding authors, **H.Z.**

Disclaimer

The views and opinions expressed in this article are those of the authors and are the product of professional research. It does not necessarily reflect the official policy or position of any affiliated institution, funder, agency, or that of the publisher. The authors are responsible for this article's results, findings, and content.

Declaration of AI and AI-assisted Technologies in the Writing Process

During the writing of this article, the author used ChatGPT for spelling and grammar checking. After using this tool, the author reviewed and edited the content as needed and assumes full responsibility for the final published content.

REFERENCE

[1] Lin, J. (2025). Application of machine learning in predicting consumer behavior and

precision marketing. *PLoS One*, *20*(5), e0321854. DOI: **https://doi.org/10.1371/journal.pone.0321854**

[2] Theodorakopoulos, L., & Theodoropoulou, A. (2024). Leveraging big data analytics for understanding consumer behavior in digital marketing: A systematic review. *Human Behavior and Emerging Technologies*, *2024*(1), 3641502. DOI: **https://doi.org/10.1155/2024/3641502**

[3] Yin, J., Qiu, X., & Wang, Y. (2025). The impact of AI-personalized recommendations on clicking intentions: Evidence from Chinese e-commerce. *Journal of Theoretical and Applied Electronic Commerce Research*, *20*(1), 21. DOI: **https://doi.org/10.3390/jtaer20010021**

[4] Koosha, H., & Albadvi, A. (2020). Allocation of marketing budgets to maximize customer equity. *Operational Research*, *20*(2), 561-583. DOI: **https://doi.org/10.1007/s12351-017-0356-z**

[5] Yan, C., Chen, Y., Wan, Y., & Wang, P. (2021). Modeling low-and high-order feature interactions with FM and self-attention network. *Applied Intelligence*, *51*(6), 3189-3201. DOI: **https://doi.org/10.1007/s10489-020-01951-6**

[6] Chen, T., Yin, H., Zhang, X., Huang, Z., Wang, Y., & Wang, M. (2021). Quaternion factorization machines: A lightweight solution to intricate feature interaction modeling. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(8), 4345-4358. DOI: **https://doi.org/10.1109/TNNLS.2021.3118706**

[7] Zhu, Z. (2022, July). Deep Learning for FM-Based Recommendation: A Systematic Study on DeepFm and Its Application. In *International Conference on Frontier Computing* (pp. 747-756). Singapore: Springer Nature Singapore. DOI: **https://doi.org/10.1007/978-981-99-1428-9_92**

[8] Zhang, P., Tang, K., Chen, G., Li, J., & Li, Y. (2024). Multimodal data fusion enhanced deep learning prediction of crack path segmentation in CFRP composites. *Composites Science and Technology*, *257*, 110812. DOI: **https://doi.org/10.1016/j.compscitech.2024.110812**

[9] Wang, K., Wang, H., Guo, W., Liu, Y., Lin, J., Lian, D., & Chen, E. (2025, July). DLF: Enhancing explicit-implicit interaction via dynamic low-order-aware fusion for CTR prediction. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 2213-2223). DOI: **https://doi.org/10.1145/3726302.3729956**

[10] Xu, K., Wang, T., & Cheng, L. (2023). Service recommendation of industrial software components based on explicit and implicit higher-order feature interactions and attentional factorization machines. *Applied Sciences*, *13*(19), 10746. DOI: **https://doi.org/10.3390/app131910746**

[11] Liu, B., Zhu, C., Li, G., Zhang, W., Lai, J., Tang, R., ... & Yu, Y. (2020, August). Autofis: Automatic feature interaction selection in factorization models for click-through rate prediction. In *proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2636-2645). DOI: **https://doi.org/10.1145/3394486.3403314**

[12] Cheng, W., Shen, Y., & Huang, L. (2020, April). Adaptive factorization network: Learning adaptive-order feature interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 04, pp. 3609-3616). DOI: **https://doi.org/10.1609/aaai.v34i04.5768**

[13] Yu, Z., Amin, S. U., Alhussein, M., & Lv, Z. (2021). Research on disease prediction based

on improved DeepFM and IoMT. *IEEE Access*, *9*, 39043-39054. DOI: **https://doi.org/10.1109/access.2021.3062687**

[14] Sun, Y., Li, J., Xu, Y., Zhang, T., & Wang, X. (2023). Deep learning versus conventional methods for missing data imputation: A review and comparative study. *Expert Systems with Applications*, *227*, 120201. DOI: **https://doi.org/10.1016/j.eswa.2023.120201**

[15] Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., & Tabona, O. (2021). A survey on missing data in machine learning. *Journal of Big data*, *8*(1), 140. DOI: **https://doi.org/10.21203/rs.3.rs-535520/v1**

[16] Wang, H., Tan, Z., Liang, Y., Li, F., Zhang, Z., & Ju, L. (2024). A novel multi-layer stacking ensemble wind power prediction model under Tensorflow deep learning framework considering feature enhancement and data hierarchy processing. *Energy*, *286*, 129409. DOI: **https://doi.org/10.1016/j.energy.2023.129409**

[17] Yu, F., & Liu, X. (2022). Research on student performance prediction based on stacking fusion model. *Electronics*, *11*(19), 3166. DOI: **https://doi.org/10.3390/electronics11193166**

[18] Chen, M., Qian, Z., Boers, N., Jakeman, A. J., Kettner, A. J., Brandt, M., ... & Lü, G. (2023). Iterative integration of deep learning in hybrid Earth surface system modelling. *Nature Reviews Earth & Environment*, *4*(8), 568-581. DOI: **https://doi.org/10.1038/s43017-023-00452-7**

[19] Ma, L., Yao, W., Dai, X., & Jia, R. (2023). A new evidence weight combination and probability allocation method in multi-sensor data fusion. *Sensors*, *23*(2), 722. DOI: **https://doi.org/10.3390/s23020722**

[20] Huang, H., Yan, X., Zheng, Y., He, J., Xu, L., & Qin, D. (2025). Multi-view stereo algorithms based on deep learning: a survey. *Multimedia Tools and Applications*, *84*(6), 2877-2908. DOI: **https://doi.org/10.1007/s11042-024-20464-9**

[21] Xie, Z., Yang, Y., Zhang, Y., Wang, J., & Du, S. (2023). Deep learning on multi-view sequential data: a survey. *Artificial Intelligence Review*, *56*(7), 6661-6704. DOI: **https://doi.org/10.1007/s10462-022-10332-z**

[22] Dong, Y., Qiu, L., Lu, C., Song, L., Ding, Z., Yu, Y., & Chen, G. (2022). A data-driven model for predicting initial productivity of offshore directional well based on the physical constrained eXtreme gradient boosting (XGBoost) trees. *Journal of Petroleum Science and Engineering*, *211*, 110176. DOI: **https://doi.org/10.1016/j.petrol.2022.110176**

[23] Demir, S., & Sahin, E. K. (2023). An investigation of feature selection methods for soil liquefaction prediction based on tree-based ensemble algorithms using AdaBoost, gradient boosting, and XGBoost. *Neural Computing and Applications*, *35*(4), 3173-3190. DOI: **https://doi.org/10.1007/s00521-022-07856-4**

[24] Liu, Z., Lu, Y., Lai, Z., Ou, W., & Zhang, K. (2021). Robust sparse low-rank embedding for image dimension reduction. *Applied soft computing*, *113*, 107907. DOI: **https://doi.org/10.1016/j.asoc.2021.107907**

[25] Guo, Y., Sun, Y., Wang, Z., Nie, F., & Wang, F. (2023). Double-structured sparsity guided flexible embedding learning for unsupervised feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(10), 13354-13367. DOI: **https://doi.org/10.1109/tnnls.2023.3267184**

[26] Singh, D., & Singh, B. (2022). Feature wise normalization: An effective way of normalizing data. *Pattern Recognition*, *122*, 108307. DOI: **https://doi.org/10.1016/j.patcog.2021.108307**

[27] Sarmadi, H., Entezami, A., & Magalhães, F. (2023). Unsupervised data normalization for continuous dynamic monitoring by an innovative hybrid feature weighting-selection algorithm and natural nearest neighbor searching. *Structural health monitoring*, *22*(6), 4005-4026. DOI: **https://doi.org/10.1177/14759217231166116**

[28] Huang, L., Qin, J., Zhou, Y., Zhu, F., Liu, L., & Shao, L. (2023). Normalization techniques in training dnns: Methodology, analysis and application. *IEEE transactions on pattern analysis and machine intelligence*, *45*(8), 10173-10196. DOI: **https://doi.org/10.1109/tpami.2023.3250241**

[29] Zhu, Z. (2022, July). Deep Learning for FM-Based Recommendation: A Systematic Study on DeepFm and Its Application. In *International Conference on Frontier Computing* (pp. 747-756). Singapore: Springer Nature Singapore. DOI: **https://doi.org/10.1007/978-981-99-1428-9_92**

[30] Qian, J., Jia, T., Zhang, W., Zeng, K., & Du, X. (2024). An industrial network traffic anomaly detection method based on improved DeepFM model. *IEEE Access*, *12*, 136222-136229. DOI: **https://doi.org/10.1109/access.2024.3419895**

[31] Yan, C., Chen, Y., Wan, Y., & Wang, P. (2021). Modeling low-and high-order feature interactions with FM and self-attention network. *Applied Intelligence*, *51*(6), 3189-3201. DOI: **https://doi.org/10.1007/s10489-020-01951-6**

[32] Zhai, Z., Shen, J., Li, P., Zhang, J., & Zhang, K. (2024, December). Flexible-Order Feature-Interaction for Mixed Continuous and Discrete Variables with Group-Level Interpretability. In *International Conference on Neural Information Processing* (pp. 42-57). Singapore: Springer Nature Singapore. DOI: **https://doi.org/10.1007/978-981-96-6576-1_4**