INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 21, Issue: 1, Month: February, Year: 2026
Article Number: 7069, https://doi.org/10.15837/ijccc.2026.1.7069

CCC Publications
AGORA UNIVERSITY PRESS

FLSEST: CTR model based on important features and soft threshold

Jianlin Chen*, Qianying He, Yonglin Zhou

Jianlin Chen*
School of Intelligence Technology, Geely University of China
Chengdu, Sichuan 610000-China
*Corresponding author: chenjianlin@guc.edu.cn

Qianying He
School of Computer Science and Engineering, Guangzhou Institute of Science and Technology
Guangzhou, Guangdong 510000-China
20230725@gzist.edu.cn

Yonglin Zhou
School of Intelligence Technology, Geely University of China
Chengdu, Sichuan 610000-China
zhouyong5217@163.com

Abstract

Click-through rate (CTR) prediction plays a pivotal role in developing effective recommendation systems across industries. While existing models like DeepFM primarily focus on low-order and high-order feature interactions, they often fail to sufficiently account for the heterogeneous importance distribution among individual features. To address this limitation, we propose FLSEST, a novel architecture integrating a squeeze-excitation and soft threshold (SEST) mechanism that dynamically amplifies discriminative features while suppressing noise from less informative ones. Drawing inspiration from FLEN's design philosophy, we additionally introduce a feature-weighting bilinear interaction (FWBI) layer to resolve gradient coupling phenomena during feature interaction learning. Extensive experimental evaluations on multiple public datasets demonstrate that our FLSEST model achieves superior prediction performance compared to state-of-the-art shallow and deep recommendation models. Moreover, integrating our proposed SEST network with mainstream models such as FwFM and DeepFM further enhances their predictive capabilities, confirming the versatility and effectiveness of our approach.

Keywords: Recommendation algorithm, Click-through rate, Deep learning, Attention mechanism, Squeeze excitation and soft threshold network.

1 Introduction

Product recommendation systems were originally developed to address the growing demands of Internet companies. Foundational data indicate that such systems contribute significantly to revenue, accounting for 35% of Amazon's sales and generating over $100 billion for ByteDance. Furthermore, approximately 60% of video traffic on YouTube-a globally recognized video-sharing platform-originates from personalized recommendations. Therefore, enhancing the accuracy of personalized recommendation systems is of great importance to both Internet enterprises and individual users. The core task of a recommendation system is to predict the click-through rate (CTR). Various models have been proposed in the literature to address this task, including logistic regression (LR) [1], gradient boosting decision tree combined with logistic regression (GBDT + LR), polynomial prediction models[2], and factorization machines (FM) [3, 4] . Feature interactions significantly affect CTR prediction. Different features represent various aspects and dimensions, making their combinations crucial. Nevertheless, traditional feature interaction approaches face inherent limitations. First, generating high-quality features comes at a high computational cost, as effective feature combinations are often crafted for specific task scenarios. As a result, engineers must spend considerable time manually designing these combinations, relying heavily on their domain expertise. A promising alternative is the application of deep learning technologies, which offer an effective solution to the challenges of manual feature engineering. These methods significantly enhance the model's ability to learn complex feature interactions automatically, thereby improving the overall prediction accuracy.

Currently, deep learning-based models are the research frontier in CTR [5–9]. Several neural network-based models have achieved notable success in both online and offline scenarios. Representative approaches include the fuzzy neural network (FNN) [10], which integrates factorization machine principles; the deep neural factorization machine (DeepFM) [11], merging a factorization machine with a fully connected neural network; the attentional factorization machine (AFM) [12], enhanced through attention mechanisms for refined feature interaction; and the automatic feature interaction selection (AutoEIS) [13], employing automated processes for feature selection.

In CTR prediction tasks, different features have different significance to the prediction results. Existing methodologies systematically investigate feature importance and interaction dynamics across diverse application domains. Key developments encompass the field-aware factorization machine (FFM) [14], which captures field-specific feature correlations; the field-weighted factorization machine (FwFM) [15] optimizing feature weighting for click-through rate prediction; and the attention-enhanced factorization machine (AFM) [12] that dynamically adjusts interaction weights through neural attention mechanisms. The feature importance and bilinear feature interaction NETwork (FiBiNET) [16] dynamically learned the importance of the features via the squeeze-and-excitation network (SENET). Unlike current trends, this study develops a new model named FLSEST that exploits important features and feature domains. The main contributions of this work are as follows:

(1) Inspired by X.Jin et al.[17] and Gomez-Rodriguez et al.[18], we propose a squeeze-excitation and soft threshold (SEST) network to learn important features and remove the unimportant ones. The input of this layer is the input embedding, while the output is the SEST embedding.

(2) We solve the gradient coupling problem by employing the FLEN feature intersection method. Specifically, the feature intersections are divided into the intersection combination MF between the general feature domains and the intersection combination FM within general feature domains [19]. Moreover, the DiceFactor mechanism is used to couple the FM, and we adopt the naming method FwBI of the original FLEN model. Our trials highlight that combining SEST and FwBI affords superior results to current state-of-the-art deep recommendation models on multiple data sets.

(3) Combining the proposed SEST network with mainstream shallow models, e.g., FwFM, and deep models, e.g., DeepFM, further improves its performance.

2 Related Works

2.1 Shallow Models

The study of recommendation algorithms can be traced back to early collaborative filtering (CF) methods, which are primarily categorized into user-based and item-based approaches [20]. However, collaborative filtering techniques often suffer from the data sparsity problem, which significantly degrades recommendation performance. To mitigate this issue, matrix factorization (MF) methods were introduced and achieved remarkable success, most notably in the Netflix recommendation competition. Since then, MF[21] has become one of the mainstream technologies in recommendation systems.

To further address the challenges posed by high-dimensional sparse data, the Factorization Machines (FM) model was proposed by Rendle in 2010. The core idea of FM lies in modeling second-order feature interactions through cross terms, while reducing computational complexity via factorization. Compared to traditional matrix factorization methods, FM not only models user–item rating matrices but also effectively incorporates contextual information—such as user behaviors and item attributes—thus enhancing the overall performance of recommendation systems [3].

Many state-of-the-art shallow models are built as extensions of FM. For instance, Field-aware Factorization Machines (FFM) [14] were developed to capture different interaction strengths across various fields. In FFM, each feature is associated with distinct latent vectors depending on the field of its interacting feature. This field-aware mechanism allows the model to learn more precise feature interactions. For example, in ad click-through rate (CTR) prediction tasks, features such as user age, gender, and device type originate from different fields. FFM captures these cross-field interactions with fine granularity, offering richer representations than standard FM. However, due to its high memory consumption, FFM is currently deployed in production environments only by a limited number of Internet companies.

To reduce memory usage while maintaining modeling performance, the Field-weighted Factorization Machine (FwFM) model was proposed in [15] as an optimized variant of FFM. FwFM introduces a field-weighting mechanism to assign learnable weights to different field pairs, effectively capturing the varying importance of cross-field interactions. This approach improves the modeling of feature relationships while maintaining computational efficiency, offering a better trade-off between accuracy and resource usage.

Additionally, the successful application of attention mechanisms in computer vision has inspired researchers to integrate similar techniques into recommendation models. The Attentional Factorization Machine [22] is a shallow model that enhances FM by incorporating an attention mechanism to model feature interactions more effectively. AFM dynamically assigns attention scores to different feature interaction pairs, enabling the model to focus on the most informative relationships while suppressing less relevant ones. This not only improves interpretability but also further boosts the accuracy of prediction tasks.

2.2 Deep Models

Leveraging deep neural networks (DNNs) to learn complex and selective feature interactions has become a prominent research direction in recommendation systems. FNN utilizes FM-pretrained feature embeddings and subsequently trains a neural network to capture higher-order feature interactions. This approach addresses the limitation of traditional FM, which can only model second-order interactions, while also alleviating the inefficiency of training deep models from scratch.

Qu et al. [23] introduced the Product-based Neural Network (PNN), which incorporates a product layer between the embedding and DNN layers to capture interactive signals directly, without relying on FM-pretrained embeddings.

He et al. [24] proposed the Neural Factorization Machine (NFM), which introduces a bi-interaction pooling layer after embedding to model second-order feature interactions. This layer retains more informative representations of the input features and reduces the learning burden on the multilayer perceptron (MLP).

Further, Yi et al. [25] developed the Operation-aware Neural Network (ONN), which enhances the processing of embedding representations by employing different embedding strategies tailored to specific operations (e.g., replication or inner product). These models typically follow a framework that combines a shallow component with an MLP, allowing for both linear and non-linear modeling of feature interactions.

In addition, hybrid architectures such as DeepFM and xDeepFM [26] combine shallow and deep components to simultaneously learn low- and high-order feature interactions. These models are designed to integrate memorization and generalization capabilities within a unified framework.

More recently, the Attentive DenseNet-based Factorization Machine has extended the DeepFM framework by incorporating an attention mechanism after the DNN layer, enabling the model to focus on the most informative feature interactions. an attention-based recommendation framework that emphasizes topic-specific features while suppressing common, less informative terms, thereby mitigating the homogenization effect often observed in recommended systems[27].

Finally, a dual structure combining a deep neural network and a cross network to explicitly capture cross-feature relationships, enabling a more expressive representation of user interaction data[28].

3 The Proposed FLSEST Model

The proposed FLSEST model is illustrated in Figure 1, and the SESTLayer part is depicted in Figure 2. mainly including five layers: 1) Input layer, which is used to input original data and perform one-hot encoding. The input encompasses various major domains (e.g., user, item) along with their granular features, such as age and gender in the user domain. 2) The embedding layer transforms sparse one-hot encodings from the input layer into dense embedding vectors via a trainable projection matrix. These vectors subsequently bifurcate into dual data streams: the first directly feeds into the FwBI layer and MLP layer, while the second is routed to the SEST layer. 3) Squeeze-excitation and soft threshold layer, which is used to learn important features and remove the unimportant features. The output of the SEST layer is the SEST embedding layer, which is parallel with the embedding layer at the same level. 4) FwBI layer, which is employed to learn the interactions between different features. 5) MLP layer to learn high-order feature interaction. The model aggregates the original outputs from the FwBI and MLP layers with their SEST-enhanced counterparts, followed by joint processing through ReLU activation for non-linear feature fusion. These components will be described in detail in the following sections.

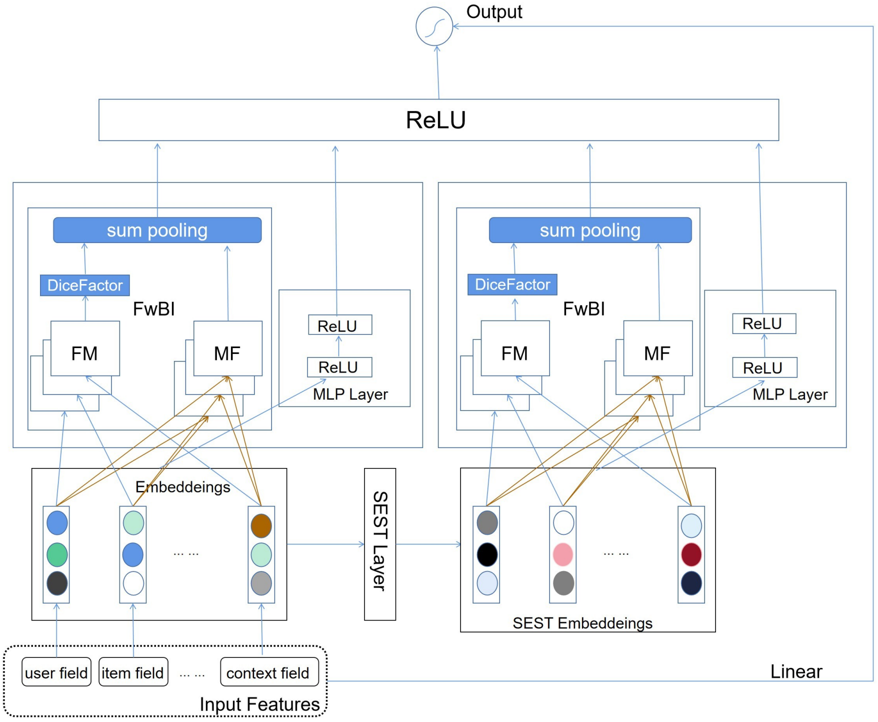


Figure 1: FLSEST Model Swales

3.1 Input Layer

In most CTR prediction tasks, the data sets have multiple feature domains, and the value of each feature domain is encoded into high dimensional sparse data through a one-hot encoding process in the

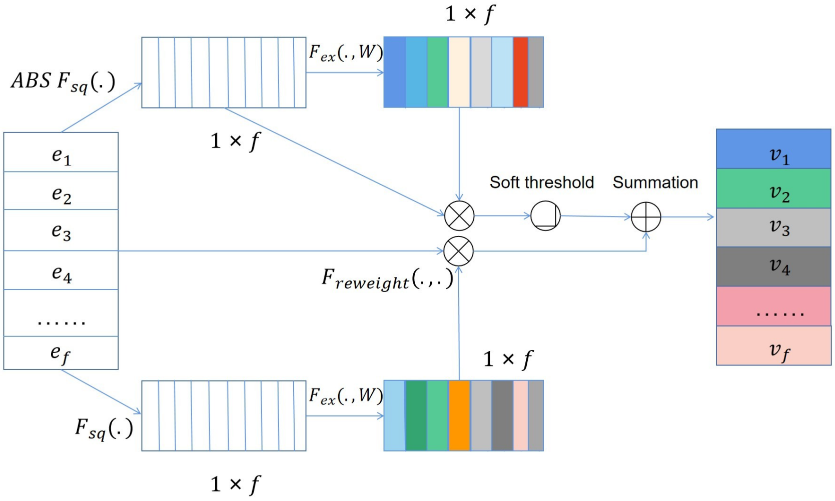


Figure 2: SEST Network

input layer. The common feature domains include “age”, “gender”, and “interest”, and the feature domain is also called a feature in several studies, but they are essentially the same. Moreover, the feature domain in this study has a hierarchical structure. For instance, the feature domains “age” and “gender” belong to the general feature domain “user”. If m feature domains exist under the general feature domain “user”, then the value of the general feature domain “user” is $x = concat(x_1, ..., x_m)$, where x_m is the feature mosaic after one-hot encoding in the math feature domain $x_m = concat(X_n|F(n) = m)$. Introducing the general feature domain aims to learn classification features utilizing the following FwBI model.

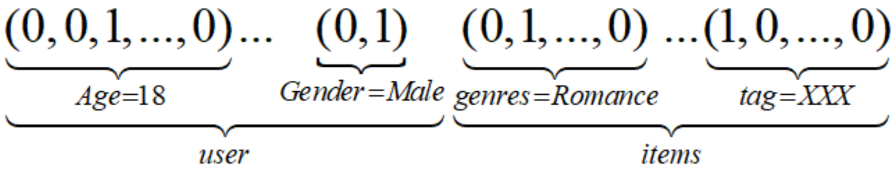


Figure 3: Feature domain

3.2 Embedding Layer

Before developing the embedding model, the one-hot encoding usually represents the words' information. However, the one-hot encoding method will lead to a high feature dimension and cannot represent the information between different words. Hence, the embedding layer transforms the high dimensional sparse discrete variables from the input layer into low dimensional dense vectors. For a single-value feature, e.g., gender=female, the vector after embedding is the embedding of this feature in the feature domain. If a feature has many values, e.g., hobby=exercise, reading, and listening to music, the vector after embedding is the sum of the embedding of all features in the feature domain. For the missing features, the embedding value is set to zero. The vector length of single-value and multi-value features after embedding is the same. As illustrated in Figure 4, the input length of different feature domains can be different, yet the embedding dimension k remains the same.

The sum of the embedding of all features in a particular general feature domain is defined as the embedding vector $e_m = \sum\limits_{n|F(n)=m} e_n$.

3.3 Squeeze-excitation and soft threshold layer

The SENET was first applied in computer vision applications and achieved great success. Furthermore, the FiBiNET model introduced SENET into the CTR field and achieved promising results.

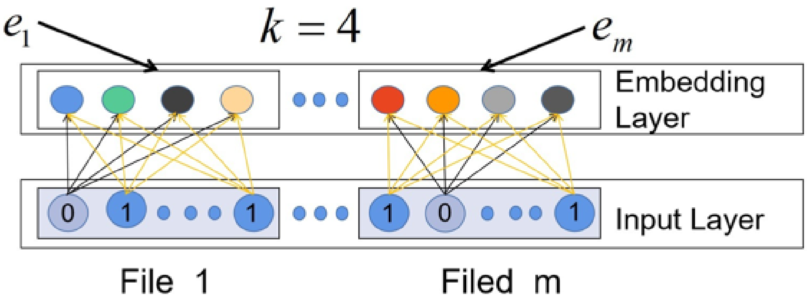


Figure 4: Structure of the embedding layer

Therefore, this study utilizes a SEST network to learn important features. The SEST architecture is illustrated in Figure 2, mainly comprising two structures, i.e., the absolute soft threshold structure and the squeeze-excitation weight structure, which are in parallel to finally obtain their sum (the squeeze mechanism in the structures is similar). The weighted squeeze excitation structure is similar to the computer vision's channel attention mechanism, which weights different features directly. The absolute soft threshold structure removes unimportant features by learning different soft thresholds for each different feature.

3.3.1 absolute soft threshold structure

(1) Absolute squeeze structure

Let the original embedding be $E = [e_1, ..., e_f]$, where $e_i \in R^k$ and k is the embedding dimension, and the squeezed vector is $Z = [z_1, ..., Z_i, ..., z_f]$, where $i \in [1, ..., f]$, and $z_i$is a scalar used to represent the global information of the ith feature. $z_i$is calculated as follows:

$$z_i = (abs)F_{sq}(e_i) = \frac{1}{k}\sum_{t=1}^{k}|e_i^{(t)}| \tag{1}$$

(2) Excitation structure

Two fully connected neural networks are used to learn the feature weights. The first network reduces the dimension ratio of r, and the activation function is σ_1. The second network increases the dimension, and the activation function is σ_2. The weight of the feature embedding is calculated as follows:

$$A = F_{ex}(Z) = \sigma_2(W_2\sigma_1(W_1Z)) \tag{2}$$

where $A \in R^f$, $W_1 \in R^{f\times\frac{f}{r}}$, and $W_2 \in R^{\frac{f}{r}\times f}$ are the learning parameters, r is the reduction ratio, and σ_1 and σ_2 are activation functions.

(3) Soft threshold structure

A soft threshold is used as the nonlinear transform layer to remove the unimportant features:

$$\eta_i(e_i, \lambda_i) = sgn(e_i)(|e_i| - \lambda_i)_+ \tag{3}$$

where λ_i is a non-negative threshold, $(|e_i| - \lambda_i)_+$ is equal to $|e_i - \lambda_i$ when $(|e_i| - \lambda_i) > 0$ and is equal to 0 when $(|e_i| - \lambda_i) < 0$, and is a sign function. In this study, the soft threshold λ_i is obtained using the automatic learning method:

$$\lambda_i = z_i \cdot a_i \tag{4}$$

Where z_i is derived as per Equation (1). And the matrix in Equation (2) can be explicitly expanded as $A = [a_1, ..., a_i, ..., a_f]$.

3.3.2 Squeeze-excitation weight structure

(1) Squeeze structure

The squeeze structure is a summary statistic used to record each feature domain, where the average pooling method is used to compress the original embedding. Let the original Embedding be $E = [e_1, ..., e_f]$, with $e_i \in R^k$, and k is the embedding dimension. The compressed vector is $Z^{'} = [z_1^{'}, ..., z_i^{'}, ..., z_f^{'}]$, where $i \in [1, ..., f]$and $z_i^{'}$ is a scalar used to represent the global information of the i-th feature is calculated as follows:

$$z_i^{'} = F_{sq}(e_i) = \frac{1}{k}\sum_{t=1}^{k} e_i^{(t)} \tag{5}$$

(2) Excitation structure

Similar to the excitation structure of Section 2.3.1, two fully connected networks are used to learn the feature weights. The first network reduces the dimension with a ratio of r and an activation function of σ_1. The second network increases the dimension with an activation function σ_2. The weight of the feature embedding is calculated as follows:

$$A = F_{ex}(Z^{'}) = \sigma_2(W_2\sigma_1(W_1Z^{'})) \tag{6}$$

(3) Weight structure

The weighting structure reweights the features. A new weight vector is obtained by multiplying the original embedding vector E and the weight vector A:

$$M = F_{ReWeight}(A, E) = [a_1 \cdot e_1, ..., a_f \cdot e_f] = [m_1, ..., m_f] \tag{7}$$

where $a_i \in R^k$, and $m_i \in R^k$.

3.3.3 Output

The output of the SEST layer is obtained by concatenating the output of the absolute soft threshold structure presented in Section 3.3.1 and the output of the squeeze excitation weight structure of Section 3.3.2, expressed as:

$$V = \eta + M = [\eta_1 + m_1, ..., \eta_f + m_f] = [v_1, ..., v_f] \tag{8}$$

3.3.4 Parameter analysis

This subsection first calculates the number of parameters of the traditional SENET structure applied to the CTR model. Let the original Embedding be $E = [e_1, ..., e_f]$, where $e_i \in R^k$, and k is the embedding dimension. Then can be expressed in more detail as $E = [e_{1,1}, ..., e_{1,k}, ..., e_{i,k}, ..., e_{f,1}, ..., e_{f,k}]$. $Z^{'}$ obtained through global average pooling in Eq. (5) can be expressed as $Z^{'} = [z_{1,1}^{'}, ..., z_{1,k}^{'}], ..., z_{i,k}^{'}, ..., z_{f,1}^{'}, ..., z_{f,k}^{'}]$. Then the weights of SENET are W_1 and W_2 as shown in equation (9).

$$\begin{cases} W_1 = [w_{1,1}, ..., w_{1,k}], ..., w_{i,k}, ..., w_{f,1}, ..., w_{f,k}] \\ W_2 = [w_{1,1}, ..., w_{1,k}], ..., w_{i,k}, ..., w_{f,1}, ..., w_{f,k}] \end{cases} \tag{9}$$

Then the total number of participants in SENET is $2f * k$. In this paper, the SEST model uses a parallel structure, one part of which is the traditional SENET structure, and the other part is based on the neural network adaptive soft thresholding of the attention mechanism named absolute soft threshold structure, and finally summing up the two parallel structures. Let the original Embedding be $E = [e_1, ..., e_f]$, with $e_i \in R^k$, and k is the embedding dimension. Then E can be expressed in more detail as $E = [e_{1,1}, ..., e_{1,k}], ..., e_{i,k}, ..., e_{f,1}, ..., e_{f,k}$. The Z obtained by absolute global average pooling through Eq. (1) can be expressed as $Z = [z_{1,1}, ..., z_{1,k}], ..., z_{i,k}, ..., z_{f,1}, ..., z_{f,k}]$. Then the weights of the absolute soft threshold structure are W_3 and W_4 as shown in equation (10).

$$\begin{cases} W_3 = [w_{1,1}, ..., w_{1,k}], ..., w_{i,k}, ..., w_{f,1}, ..., w_{f,k}] \\ W_4 = [w_{1,1}, ..., w_{1,k}], ..., w_{i,k}, ..., w_{f,1}, ..., w_{f,k}] \end{cases} \tag{10}$$

Then the number of parameters of absolute soft threshold structure is $2f * k$. The final number of parameters of the SEST model in this paper is the sum of the number of parameters of the two parallel structures as $4f * k$. It can be seen that the number of parameters in the SEST model of this paper is in the same order of magnitude compared to the traditional SENET model.

3.4 FwBI layer

The FwBI layer is used to learn the feature interactions and is divided into three parts: linear, FM, and MF. The linear part is similar to the linear part in the FMM model and the FwFM model, which is used to learn the data's overall bias and feature weight. The output is a linear combination of all features plus a bias term and is calculated as follows:

$$L = w_0 + \sum_{i=1}^{f} x_i w_i \tag{11}$$

where w_0 is the global bias and f is the total number of feature domains. This model has two parallel embedding layers: the original and the SEST embedding layers. Both layers introduce the FwBI layer to learn the feature interaction, and only one linear operation is performed.

The second part, FM is used to learn the feature interactions in a general feature domain, expressed as follows:

$$FM = \sum_{m}^{M} [(\sum_{n|F(n)=m} e_n) \odot (\sum_{n|F(n)=m} e_n) - \sum_{n|F(n)=m} (e_n \odot e_n)] r[m][m] \tag{12}$$

where $\odot$ denotes the Hadamard product, $r[m][m]$ captures inter-domain relative importance (e.g., user vs. item domains), implemented as a learnable parameter that is dynamically optimized through backpropagation and gradient descent during model training. And $\sum_{n|F(n)=m} e_n$ is introduced in Section 2.2. The third part is the MF part, which is used to learn the feature interactions between different general feature domains, expressed as follows:

$$MF = \sum_{i=1}^{M} \sum_{j=i+1}^{M} (e_{(i)} \odot e_{(j)}) r[i][j] \tag{13}$$

where M is the number of general feature domains, and $r[i][j]$ captures intra-domain feature interaction strength (e.g., between age and gender attributes within the user domain). Implemented as a trainable parameter, it is dynamically optimized via backpropagation and gradient descent during model training. The FwBI layer is expressed as follows:

$$FwBI = w_0 + \sum_{i=1}^{f} x_i w_i + \sum_{i=1}^{M} \sum_{j=i+1}^{M} (e_{(i)} \odot e_{(j)}) r[i][j] + \sum_{m}^{M} [(\sum_{n|F(n)=m} e_n) \odot (\sum_{n|F(n)=m} e_n) - \sum_{n|F(n)=m} (e_n \odot e_n)] r[m][m] \tag{14}$$

In several cases, FwBI fits the current partial shallow model, e.g., when $M = 1$ and $r[m][m] = 0.5$, The FwBI layer is expressed as follows:

$$FwBI = FM = w_0 + \sum_{i=1}^{f} x_i w_i + \frac{1}{2} \sum_{m}^{M} [(\sum_{n|F(n)=m} e_n) \odot (\sum_{n|F(n)=m} e_n) - \sum_{n|F(n)=m} (e_n \odot e_n)] \tag{15}$$

when f feature domains exist, i.e., $M = f$, then The FwBI layer is expressed as follows:

$$FwBI = FwFm = w_0 + \sum_{i=1}^{M} x_i w_i + \sum_{i=1}^{M} \sum_{j=i+1}^{M} (e_{(i)} \odot e_{(j)}) r[i][j] \tag{16}$$

We employ DiceFactor to decouple FM [12] internally. Similar to the DropOut mechanism, the DiceFactor's core is to randomly discard the feature interactions to prevent the adaptation of one feature to other features.

3.5 MLP layer

he MLP layer comprises several fully connected layers and is used to learn complex implicit higher-order feature interactions. Let the output vector of the embedding layer be $E^{(0)} = [e_1, ..., e^f]$, then the forward transfer of the MLP layer is:

$$E^{(l+1)} = \sigma(W^{(l)} E^{(l)} + b^{(l)}) \tag{17}$$

where l is the number of fully connected layers, σ is the activation function, $W^{(l)}$ is the weight of the l-th fully connected layer, and $b^{(l)}$ is the bias of the l-th layer. The final output of the MLP layer is:

$$y = \sigma(W^{(H+1)} E^{(H)} + B^{(H+1)}) \tag{18}$$

where H is the total number of layers in the fully connected neural network.

4 Experiment and Results

4.1 Datasets and Data Preprocessing

This work employs the Movilens-100k, Movilens-1m, and Avazu-first-one-million-records datasets. The GroupLens Research developed the Movilens datasets at the University of Minnesota containing users' ratings of movies ranging from 1-5 points from dislike to like. The datasets include basic user and movie information. The user information includes user id, gender, age, and occupation, while the movie information includes movie id, title, and category. In order to facilitate the binary classification training, the movies with a score less than four are defined as dislike, and the remaining ones as like. The Avazu-first-one-million-records dataset is from the Kaggle website , containing basic information such as user id, click or not, click time, website domain name, website category, app domain name, and app category.

4.2 Evaluation Indicators

This study utilizes the AUC and LogLoss evaluation indicators. AUC is an essential index for binary classification evaluation, which is the area under the ROC curve and above the x-axis. The larger the AUC, the better the classification performance. AUC is defined as follows:

$$AUC = \frac{\sum\limits_{ins_i \in positive} rank_{ins_i} - \frac{M \times (M+1)}{2}}{M \times N} \tag{19}$$

where M and N are the number of positive samples and the negative samples, respectively. $rank_{ins_i}$ is the rank of the i-th sample, i.e., the rank of the probability score of the i-th sample. $\sum\limits_{ins_i \in positive} rank_{ins_i}$ is the sum of the ranks of the positive samples.

LogLoss is a logarithmic loss function. The lower the LogLoss value, the better the classification performance. LogLoss is defined as follows:

$$LogLoss = -\frac{\sum\limits_{i=1}^{n} (label_{(i)} \times log(ectr_{(i)}) + (1 - label_{(i)} \times log(1 - ectr_{(i)}))}{n} \tag{20}$$

where n is the total number of samples, label represents the true label of the sample, and ectr is the estimated CTR value.

4.3 Experiment process and Results

4.3.1 Parameter settings

The dataset is split into training and testing sets in an 8:2 ratio, reserving 10% of the training data for validation. For training, a batch size of 64 is applied to the MovieLens-100k dataset, while 128 is used for both MovieLens-1M and Avazu's first one million records. We employ the Adam optimizer with a fixed random seed of 1024, training for 100 epochs. The model configuration includes an embedding dimension of 16, a learning rate of 0.00001, and the ReLU activation function.

4.3.2 Comparison

In order to verify the FLSEST model's effectiveness, we challenge it against the current state-of-the-art models:
(1) NFM: This is an early classic model in the deep recommendation domain, which is essentially a linear combination of FM and a DNN fully connected neural network.
(2) DeepFM: This model comprises an FM component and a deep component integrated into a parallel structure that learns the interaction between low-order and implicit high-order features.
(3) xDeepFM: xDeepFM is obtained by combining the CIN model and the MLP with a parallel structure, which can implicitly learn low- and high-order feature interactions.
(4) FiBiNet: It utilizes SENET to learn feature weights dynamically.
(5) FLEN: This model mainly aims at the gradient coupling problem.
(6) EDCN[29]: Enhancing Explicit and Implicit Feature Interactions via Information Sharing for Parallel Deep CTR Models.
(7) FinalMLP[30]: An Enhanced Two-Stream MLP Model for CTR Prediction.
(8) DisenCTR[31]: Dynamic Graph-based Disentangled Representation for Click-Through Rate Prediction.

The above partial model implementation code cite github's deepctr repository[32]. The corresponding evaluation results are reported in Table 1.

Table 1: Performance of different models on the datasets

	movielens-100k		**movielens-1m**		**avazu**	
Model	**AUC**	**LogLoss**	**AUC**	**LogLoss**	**AUC**	**LogLoss**
NFM	0.7816	0.5585	0.8096	0.5210	0.7617	0.3787
DeepFM	0.7825	0.5583	0.8123	0.5179	0.7608	0.3812
xDeepFM	0.7846	0.5562	0.8139	0.5174	0.7613	0.3815
FiBiNet	0.7818	0.5559	0.8127	0.5184	0.7624	0.3781
FLEN	0.7838	0.5557	0.8145	0.5156	0.7612	0.3840
EDCN	0.7843	0.5548	0.8153	0.5151	0.7628	0.3775
FinalMLP	0.7856	0.5546	0.8156	0.5143	**0.7636**	**0.3774**
DisenCTR	0.7849	0.5552	0.8150	0.5154	0.7622	0.3780
FLSEST	**0.7859**	**0.5532**	**0.8161**	**0.5132**	0.7629	0.3778

Table 1 highlights that the NFM model has a poor performance on the Movielens-100k dataset, and the proposed FLSEST affords the best performance. Compared with FiBiNet and FLEN, the AUC of FLSEST is 0.41% and 0.21% higher, respectively. Moreover, the LogLoss of FLSEST is 0.27% and 0.25% lower than FiBiNet and FLEN, respectively. This is because, in FLSEST, the feature intersection is divided into the intersection combination MF between the general feature domains and the intersection combination FM within general feature domains through the FwBI layer, reducing the gradient coupling. In addition, in the developed network, learning important features through the SEST network improves the recommendation results.

From the results on the Movielens-1m and Avazu-first-one-million-records datasets, the AUC and LogLoss of all models increase when the number of data samples increases. Moreover, all models achieve similar performance to the Movieens-100k dataset, and the proposed FLSEST model still outperforms the other compared methods on the Movielens-1m dataset, and only slightly underperforms FinalMLP on the Avazu-first-one-million-records dataset.

4.3.3 Effect of SEST on shallow models

This experiment involves two state-of-the-art shallow models, the FM and the FwFM, which are integrated with SEST. The experimental results on Movieens-100k, Movielens-1m and Avazu-first-one-million-records datasets are shown in Table 2, highlighting that SEST increases the score of both shallow models.

Table 2: SEST increases the score of shallow models

	movielens-100k		**movielens-1m**		**avazu**	
Model	**AUC**	**LogLoss**	**AUC**	**LogLoss**	**AUC**	**LogLoss**
FM	0.7620	0.5679	0.8033	0.5302	0.7570	0.3920
FM+SEST	**0.7792**	**0.5608**	**0.8080**	**0.5256**	**0.7604**	**0.3865**
FwFM	0.7795	0.5615	0.8068	0.5249	0.7595	0.3846
FwFM+SEST	**0.7811**	**0.5590**	**0.8111**	**0.5220**	**0.7611**	**0.3821**

4.3.4 Effect of SEST on deep models

In the subsequent trial, we consider two deep models, NFM and DeepFM, which were integrated with SEST and tested on the Movielens-100k, Movielens-1m and Avazu-first-one-million-records datasets. The corresponding results are reported in Table 3, revealing that SEST increases the score of both deep models. DeepFM with the addition of SEST works best on the Movielens-100k and Movielens-1m datasets, and NFM with the addition of SEST works best on the Avazu-first-one-million-records dataset. Moreover, compared with the results of Table 2, the effect of SEST on deep models is less evident than on shallow models.

Table 3: SEST increases the score of deep models

	movielens-100k		**movielens-1m**		**avazu**	
Model	**AUC**	**LogLoss**	**AUC**	**LogLoss**	**AUC**	**LogLoss**
NFM	0.7816	0.5585	0.8096	0.5210	0.7617	0.3787
NFM+SEST	**0.7832**	**0.5582**	**0.8112**	**0.5201**	**0.7622**	**0.3773**
DeepFM	0.7825	0.5583	0.8123	0.5179	0.7608	0.3812
DeepFM+SEST	**0.7836**	**0.5579**	**0.8134**	**0.5173**	**0.7618**	**0.3776**

4.3.5 Comparison between SEST and SENET

In this experiment, the developed SEST model replaces the SENET module in the FiBiNet model, and thus FiBiSEST is created. Then, FiBiSEST and FiBiNet are compared as deep models, and SENET and SEST are integrated with FwFM and compared as shallow models. In this subsection of the experimental metrics in addition to AUC and LogLoss we introduce LogSize, which is the size of the weights trained by the model. The corresponding results on the Movieens-100k, Movielens-1m and Avazu-first-one-million-records datasets are presented in Table 4.

From Table 4, it can be seen that the LogSize of the models FwFM+SEST and FiBiSEST are slightly higher than those of FwFM+SENET and FiBiNET, respectively, which is consistent with the analysis in the parameter analysis section of 3.3.4. In addition Table 4 shows that on all three datasets Movielens-100k, Movielens-1m and Avazu-first-one-million-records, the shallow model FwFM+SEST has higher AUC and lower LogLoss than FwFM+SENET. Replacing the SENET module in the FiBiNet model with the SEST model creates a new depth model, FiBiSEST, which performs better than

FiBiNet on all three datasets. Thus, the SEST proposed in this study is superior to SENET in learning important features.

Table 4: SESTvs. SENET

	movielens-100k			**movielens-1m**			**avazu**		
Model	**AUC**	**LogLoss**	**LogSize**	**AUC**	**LogLoss**	**LogSize**	**AUC**	**LogLoss**	**LogSize**
FwFM+SENET	0.7806	0.5598	**0.47MB**	0.8083	0.5255	**1.03MB**	0.7604	0.3838	**19.4MB**
FwFM+SEST	**0.7811**	**0.5590**	0.51MB	**0.8111**	**0.5220**	1.07MB	**0.7611**	**0.3821**	19.7MB
FiBiNET	0.7818	0.5559	**1.31MB**	0.8127	0.5184	**1.87MB**	0.7624	0.3781	**36.9MB**
FiBiSEST	**0.7839**	**0.5537**	1.51MB	**0.8145**	**0.5153**	2.07MB	**0.7632**	**0.3776**	39.4MB

4.3.6 The ablation study of the FLSEST model

In this subsection, we conduct an ablation study on all components of FLSEST, including the FwBI layer, the MLP layer, and the SEST layer.

Table 5: Ablation study of the FLSEST model

Dataset	**FwBI**	**MLP**	**SEST**	**AUC**	**LogLoss**
movielens-100k	✓	✓	✓	0.7797	0.5612
	✓	✓	✓	0.7806	0.5595
	✓	✓	✓	0.7809	0.5591
	✓	✓	✓	0.7831	0.5561
	✓	✓	✓	**0.7859**	**0.5532**
movielens-1m	✓	✓	✓	0.7859	0.5532
	✓	✓	✓	0.8078	0.5238
	✓	✓	✓	0.8125	0.5211
	✓	✓	✓	0.8135	0.5184
	✓	✓	✓	**0.8161**	**0.5132**
avazu	✓	✓	✓	0.7598	0.3839
	✓	✓	✓	0.7609	0.3814
	✓	✓	✓	0.7608	0.3815
	✓	✓	✓	0.7622	0.3792
	✓	✓	✓	**0.7629**	**0.3778**

As demonstrated in Table 5, the proposed SEST Layer delivers substantial performance gains to the model. Both the FwBI Layer and MLP Layer exhibit inferior results when used independently compared to their synergistic integration with the SEST Layer.

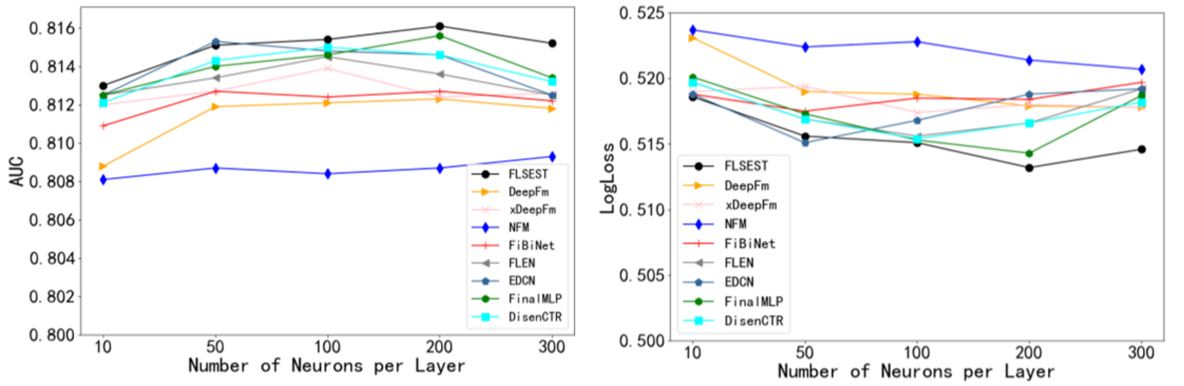
4.3.7 Influence of hyperparameters on the model

Finally, the influence of some hyperparameters on the model's performance is investigated on the Movielens-1m dataset. Considering the second DNN neuron layer as an example, the effect of the neuron cardinality is illustrated in Figure 5.

When other factors remain constant, increasing the number of neurons in each layer increases the model's complexity, which is not always beneficial (Figure 5). For example, for the FLSEST model, the growth rate slows down when the neurons exceed 50, and the model's performance is negatively affected when the neurons increase from 200 to 300. The latter performance degrades due to overfitting caused by the large neuron cardinality.

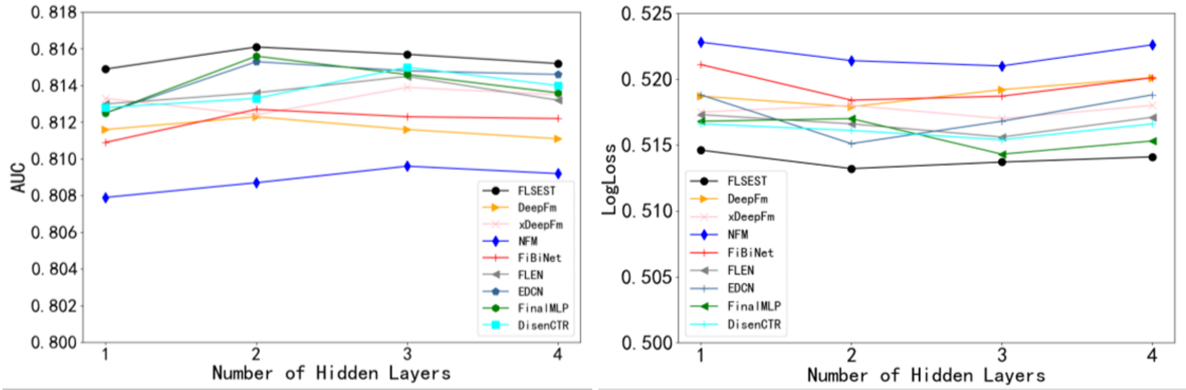
When the number of neurons per layer is 200, the interplay of the DNN layer cardinality on the model's performance is depicted in Figure 6.

When all factors but the number of DNN layers remain constant, increasing the number of DNN layers increases the model's complexity, which is not always beneficial (Figure 6). Specifically, the



(a) AUC comparison of the number of neurons (b) Logloss comparison of the number of neurons

Figure 5: Effect of the number of DNN neurons on the model

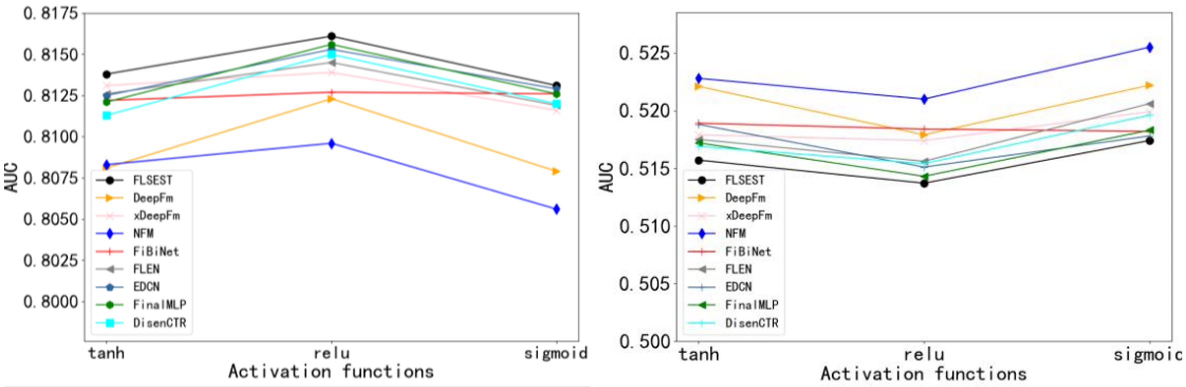


(a) AUC comparison of the number of layers(b) Logloss comparison of the number of layers

Figure 6: Effect of the number of DNN layers on the model

AUC and Logloss of most models increase first and then reduce as the number of layers increases due to overfitting.

Considering the optimal hyperparameters utilized in this trial, the effects of different activation functions on the model are illustrated in Figure 7. The latter figure demonstrates that ReLU attains the best results for almost all models, followed by Tanh, while Sigmoid presented the poorest performance.



(a) AUC comparison of activation functions(b) Logloss comparison of activation functions

Figure 7: Effect of the activation function on the model

Additionally, we investigated the effects of dropout rates and learning rates on the FLSEST model, with the corresponding experimental results illustrated in Figures 8 and 9, respectively.

The experimental results revealed that the effectiveness of the FLSEST model initially increased

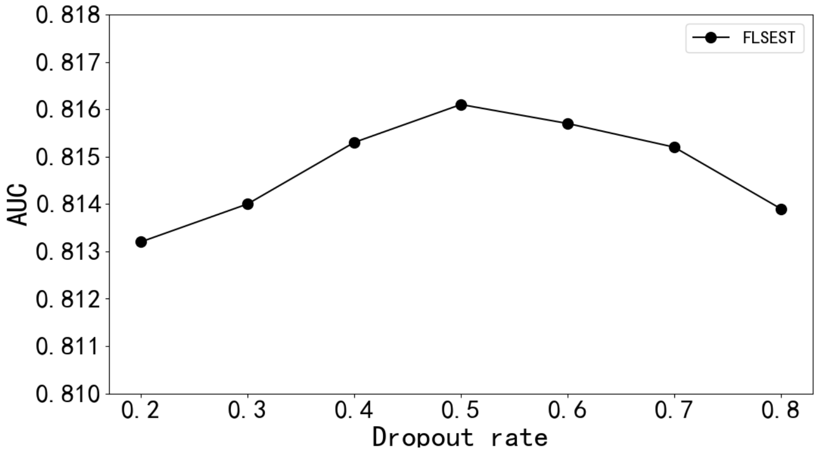


Figure 8: Effect of the dropout rate on the model

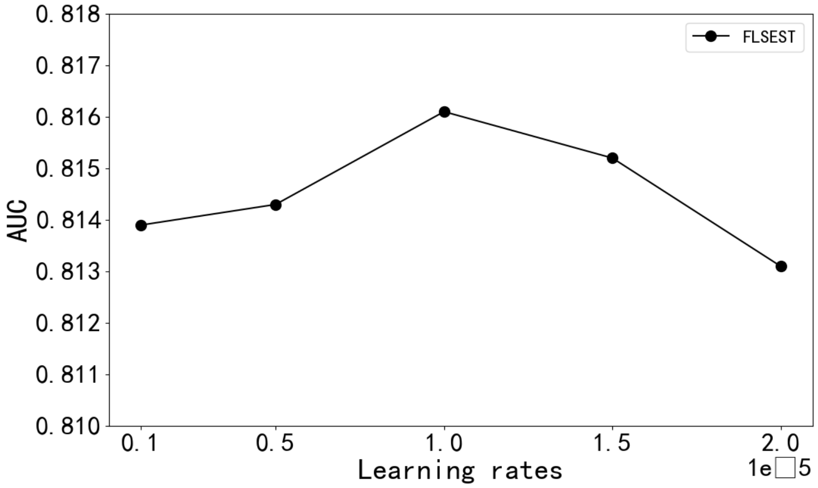


Figure 9: Effect of the learning rates on the model

and subsequently decreased as the dropout rate varied, peaking at a dropout rate of 0.5. Figure 9 shows that the FLSEST model achieves optimal performance at a learning rate of 0.00001.

5 Conclusions

This study proposes a novel CTR model named FLSEST, which learns critical features in the CTR domain through the proposed SEST network. In deep recommendation algorithms for CTR, different features inherently possess varying importance levels. Integrating the SEST network with existing shallow and deep models further enhances CTR prediction performance, demonstrating its versatility and effectiveness. In order to solve the gradient coupling problem, we adopt the idea of the FLEN model considering feature intersections, i.e., the feature intersections are divided into the intersection combination MF between general feature domains and the intersection combination FM within the general feature domain. All experimental results in Section 4 (Experiments and Results) substantiate our claims. In future work, we will investigate the long-term temporal dynamics of feature importance variations in CTR scenarios.

Author contributions

The authors contributed equally to this work.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Y. Yang and P. Zhai, "Click-through rate prediction in online advertising: A literature review," *Information Processing & Management*, vol. 59, no. 2, p. 102853, 2022.

[2] P. Morala, J. A. Cifuentes, R. E. Lillo, and I. Ucar, "Towards a mathematical framework to inform neural network modelling via polynomial regression," *Neural Networks*, vol. 142, pp. 57–72, 2021.

[3] M. Blondel, A. Fujino, N. Ueda, and M. Ishihata, "Higher-order factorization machines," *Advances in neural information processing systems*, vol. 29, 2016.

[4] S. Rendle, "Factorization machines with libfm," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 3, pp. 1–22, 2012.

[5] Z. Li, D. Jin, and K. Yuan, "Attentional factorization machine with review-based user–item interaction for recommendation," *Scientific Reports*, vol. 13, no. 1, p. 13454, 2023.

[6] Y. Yin, N. D. Ochieng, J. Sun, X. Bao, and Z. Wang, "Penet: A feature excitation learning approach to advertisement click-through rate prediction," *Neural Networks*, vol. 172, p. 106127, 2024.

[7] S. Guo, X. Liao, F. Meng, Q. Zhao, Y. Tang, H. Li, and Q. Zong, "Fsasa: Sequential recommendation based on fusing session-aware models and self-attention networks," *Computer Science and Information Systems*, vol. 21, no. 1, pp. 1–20, 2024.

[8] X. Zeng, S. Li, Z. Zhang, L. Jin, Z. Guo, and K. Wei, "Rain: Reconstructed-aware in-context enhancement with graph denoising for session-based recommendation," *Neural Networks*, vol. 184, p. 107056, 2025.

[9] S. Wang, J. Zhu, Y. Wang, C. Ma, X. Zhao, Y. Zhang, Z. Yuan, and S. Ruan, "Hierarchical gating network for cross-domain sequential recommendation," *ACM Transactions on Information Systems*, vol. 43, no. 4, pp. 1–32, 2025.

[10] P. V. de Campos Souza, "Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature," *Applied soft computing*, vol. 92, p. 106275, 2020.

[11] H. Wu, Z. Zhang, K. Yue, B. Zhang, J. He, and L. Sun, "Dual-regularized matrix factorization with deep neural networks for recommender systems," *Knowledge-Based Systems*, vol. 145, pp. 46–58, 2018.

[12] P. Wen, W. Yuan, Q. Qin, S. Sang, and Z. Zhang, "Neural attention model for recommendation based on factorization machines," *Applied Intelligence*, vol. 51, pp. 1829–1844, 2021.

[13] K. Xiao, X. Jiang, P. Hou, and H. Zhu, "Autoeis: Automatic feature embedding, interaction and selection on default prediction," *Information Processing & Management*, vol. 61, no. 1, p. 103526, 2024.

[14] L. Zhang, W. Shen, J. Huang, S. Li, and G. Pan, "Field-aware neural factorization machine for click-through rate prediction," *IEEE Access*, vol. 7, pp. 75 032–75 040, 2019.

[15] J. Pan, J. Xu, A. L. Ruiz, W. Zhao, S. Pan, Y. Sun, and Q. Lu, "Field-weighted factorization machines for click-through rate prediction in display advertising," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1349–1357.

[16] T. Huang, Z. Zhang, and J. Zhang, "Fibinet: combining feature importance and bilinear feature interaction for click-through rate prediction," in *Proceedings of the 13th ACM conference on recommender systems*, 2019, pp. 169–177.

[17] X. Jin, Y. Xie, X.-S. Wei, B.-R. Zhao, Z.-M. Chen, and X. Tan, "Delving deep into spatial pooling for squeeze-and-excitation networks," *Pattern Recognition*, vol. 121, p. 108159, 2022.

[18] M. Gomez-Rodriguez, L. Song, H. Daneshm, and B. Schölkopf, "Estimating diffusion networks: Recovery conditions, sample complexity and soft-thresholding algorithm," *Journal of Machine Learning Research*, vol. 17, no. 90, pp. 1–29, 2016.

[19] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Fedctr: Federated native ad ctr prediction with cross-platform user behavior data," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–19, 2022.

[20] M. Papagelis and D. Plexousakis, "Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents," *Engineering Applications of Artificial Intelligence*, vol. 18, no. 7, pp. 781–789, 2005.

[21] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[22] G. Kang, L. Ding, J. Liu, B. Cao, and Y. Xu, "Web api recommendation based on self-attentional neural factorization machines with domain interactions," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 6, pp. 3953–3963, 2023.

[23] Y. Qu, B. Fang, W. Zhang, R. Tang, M. Niu, H. Guo, Y. Yu, and X. He, "Product-based neural networks for user response prediction over multi-field categorical data," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 1, pp. 1–35, 2018.

[24] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017, pp. 355–364.

[25] Y. Yang, B. Xu, S. Shen, F. Shen, and J. Zhao, "Operation-aware neural networks for user response prediction," *Neural Networks*, vol. 121, pp. 161–168, 2020.

[26] Q. Lu, S. Li, T. Yang, and C. Xu, "An adaptive hybrid xdeepfm based deep interest network model for click-through rate prediction system," *PeerJ Computer Science*, vol. 7, p. e716, 2021.

[27] A. Kumar, D. K. Jain, A. Mallik, and S. Kumar, "Modified node2vec and attention based fusion framework for next poi recommendation," *Information Fusion*, vol. 101, p. 101998, 2024.

[28] Q. Zhang, W. Liao, G. Zhang, B. Yuan, and J. Lu, "A deep dual adversarial network for cross-domain recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3266–3278, 2021.

[29] B. Chen, Y. Wang, Z. Liu, R. Tang, W. Guo, H. Zheng, W. Yao, M. Zhang, and X. He, "Enhancing explicit and implicit feature interactions via information sharing for parallel deep ctr models," in *Proceedings of the 30th ACM international conference on information & knowledge management*, 2021, pp. 3757–3766.

[30] K. Mao, J. Zhu, L. Su, G. Cai, Y. Li, and Z. Dong, "Finalmlp: an enhanced two-stream mlp model for ctr prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 4, 2023, pp. 4552–4560.

[31] Y. Wang, Y. Qin, F. Sun, B. Zhang, X. Hou, K. Hu, J. Cheng, J. Lei, and M. Zhang, "Disenctr: Dynamic graph-based disentangled representation for click-through rate prediction," in *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, 2022, pp. 2314–2318.

[32] Z. Tao, X. Wang, X. He, X. Huang, and T.-S. Chua, "Hoafm: a high-order attentive factorization machine for ctr prediction," *Information Processing & Management*, vol. 57, no. 6, p. 102076, 2020.

Copyright ©2026 by the authors. Licensee Agora University, Oradea, Romania.
This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License.
Journal's webpage: http://univagora.ro/jour/index.php/ijccc/

C O P E
Member since 2012
JM08090

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:

Chen, J.; He, Q.; Zhou, Y. (2026). FLSEST: CTR model based on important features and soft threshold, *International Journal of Computers Communications & Control*, 21(1), 7069, 2026.
https://doi.org/10.15837/ijccc.2026.1.7069