Multi-source Feature Fusion and Information Flow Advertising Effectiveness Prediction Model Based on TabTransformer

Xia Yan[1, 2], Anita Binti Rosli[2], Aryaty Binti Alwie[2]
[1]College of Journalism and Communication, Shanghai Jian Qiao University, Shanghai 201306, China
[2]Department of Social Science and Management, Faculty of Humanities, Management and Science, Universiti Putra Malaysia Sarawak, 97008, Bintulu, Sarawak, Malaysia
Corresponding author: Anita Binti Rosli (e-mail: anitarosli@upm.edu.my)

ABSTRACT Click-through rate (CTR) prediction for feed advertisements relies on effective modeling of multi-source heterogeneous features. However, the inherent difficulty in capturing contextual semantic relationships among different feature types significantly limits model generalization. To address this issue, this paper proposes a multi-source feature fusion framework based on TabTransformer by encoding heterogeneous categorical information into a unified feature table. The self-attention mechanism is employed to model high-order semantic interactions, while multi-head attention dynamically captures contextual dependencies among features. Meanwhile, a parallel residual pathway integrates numerical information with Transformer representations to enhance feature complementarity. The fused representations are processed by a fully connected layer with Sigmoid activation, and model optimization is performed using cross-entropy loss. Considering that multi-source information fusion and contextual dependency modeling are also essential for intelligent electromagnetic information processing and adaptive communication systems, the proposed framework provides a data-driven reference for complex heterogeneous signal representation and decision optimization . Experimental results demonstrate that the proposed method achieves an AUC of 0.945 and a LogLoss of 0.352 in CTR prediction. In cold-start scenarios, the model maintains a LogLoss of only 0.402 for new users, while outperforming DeepFM and DCN in top-level recommendation accuracy (P@1 = 0.582). Furthermore, its probability calibration error of only 0.008 provides reliable support for advertising bidding and effectively alleviates the generalization degradation caused by insufficient user or advertisement embeddings.

INDEX TERMS click-through rate prediction, multi-source feature fusion, tabtransformer, self-attention mechanism, electromagnetic information processing

I. INTRODUCTION

AS digital transformation sweeps across all sectors, the industry is increasingly adopting online strategies for both sourcing and sales. Correspondingly, with the rapid development of internet advertising, information flow advertising has simultaneously become the most important and pervasive form of advertising across the entire industry, crucially influencing the market presence of both raw materials and finished products. Information flow advertising integrates advertising content into the user's browsing flow in a natural and non-intrusive manner, making ads more attractive and increasing their exposure and clickthrough rate (CTR) [1,2]. CTR prediction plays a crucial role in information flow advertising systems. Its core task is to predict the click probability of an ad based on historical user behavior, ad features, and other contextual information, thereby achieving personalized recommendations and maximizing advertising effectiveness. However, with the rapid growth and increasing complexity of information flow advertising data, traditional CTR prediction methods have been unable to meet the needs of efficient and accurate prediction, and marketing strategy research has lagged behind [3,4]. This surge in digital complexity, particularly in fields like e-commerce where product characteristics are highly varied and sourcing data is complex, highlights a persistent issue: how to effectively model different types of features in complex, multi-source, heterogeneous data environments remains a major research challenge in advertising systems [5,6]. Current research on ad CTR prediction largely relies on modeling multiple categorical features, such as users, ads, and context. The most common approach in these studies is to use an embedding + MLP architecture to learn the representation of categorical features. This approach maps categorical features into a fixed lowdimensional space and

then applies a nonlinear transformation using a multilayer perceptron (MLP) to capture the underlying correlations between categorical features. The core challenge currently facing the field of CTR prediction is how to effectively handle the complex interactive relationships of multi-source heterogeneous data. When dealing with high-dimensional sparse category features, the traditional embedding+MLP architecture often uses fixed-dimensional embedding vector representations, which makes it difficult to capture the dynamic semantic associations between features, resulting in a significant decrease in prediction performance in data-sparse areas. In recent years, deep learning technology has been widely used in tasks such as advertising content recognition, user behavior prediction, and CTR estimation in the research of information flow advertising and user behavior modeling. Wang et al. proposed a product placement video advertising saliency evaluation model based on deep learning, verifying the key role of visual attention in advertising effectiveness [7]. In the research on system security related to advertising recommendation, Wang et al. and Kalidindi et al. respectively proposed network intrusion detection systems based on TabTransformer, which effectively addressed the problem of data imbalance attack identification [8,9]. In addition, Häglund et al. explored contextual advertising push methods based on AI (Artificial Intelligence) without relying on user personal data, providing a new solution path for protecting privacy and improving relevance [10]. These studies provide methodological and technical support for the TabTransformer-based information flow advertising CTR prediction model proposed in this study, especially in terms of feature modeling, user behavior understanding, and contextual semantic capture.

To overcome the limitations of traditional CTR prediction models in feature modeling, this paper proposes a multi-source feature fusion model based on the TabTransformer. This model uniformly encodes categorical features such as users, ads, and context. It uses a self-attention mechanism to deeply explore the contextual dependencies and semantic interactions between them. Multi-head attention is employed to dynamically assign feature weights, improving the modeling of high-order relationships. Furthermore, to effectively handle numerical features, a parallel residual pathway is designed, combined with normalization to enhance model robustness. The fused features are fed into a fully connected layer with a Sigmoid activation to output the CTR prediction value. Finally, training is performed using a cross-entropy loss, which improves prediction accuracy and generalization. In terms of model architecture, this paper leverages the self-attention mechanism to explore high-order semantic relationships between categorical features. A parallel residual pathway for numerical features is designed, which is seamlessly integrated with the Transformer output through LayerNorm normalization and residual connections, effectively addressing the scale inconsistency issue in modeling mixed data types. Experimental results demonstrate that the proposed model achieves significant performance improvements over traditional methods on multiple datasets. The method demonstrates superior generalization capabilities, particularly for cold-start and long-tail category problems, demonstrating the vast potential of TabTransformer for multi-source feature fusion. From a commercial perspective, this model offers benefits that extend to specialized industries. It not only improves the overall prediction accuracy of the advertising system for all products but also provides a reliable basis for ad bidding decisions—especially critical for high-value machinery or B2B material sourcing—through more accurate probability calibration. This directly contributes to the platform's advertising revenue growth while simultaneously enhancing the user experience by delivering highly relevant product promotions.

II. MULTI-SOURCE FEATURE INTERACTION MODELING MECHANISM

UNIFIED REPRESENTATION OF MULTI-SOURCE CATEGORICAL FEATURES

In the context of CTR prediction for information flow advertising, model input features typically come from multiple data sources, including user profiles, ad content, exposure environment, and behavioral context, with categorical data types being the primary focus [11,12]. To achieve unified modeling of features from different sources, this section formats the raw features, performs index mapping, and initializes embedding vectors to ensure that the subsequent Transformer architecture can receive structured and semantically aligned input.

CATEGORICAL FEATURE PREPROCESSING AND INDEX MAPPING

The categorical features in the original data are represented in the form of strings or IDs and must first be standardized, preprocessed, and mapped. For user features such as gender, age group, region, and interest tags, and advertising features such as advertising ID, advertiser ID, and advertising industry category, as well as context features such as display location, device type, and timestamp, a unified numerical indexing method is used for encoding [13,14]. Independent vocabulary is constructed for all categorical fields, with each field individually numbered to avoid feature space conflicts. For example, user gender "male" and device type "mobile", despite being semantically unrelated, may be mapped to the same integer. To prevent semantic confusion, a separate mapping table is required.

EMBEDDING REPRESENTATION GENERATION AND FEATURE CONCATENATION

To facilitate subsequent modeling within the TabTransformer architecture, all encoded categorical features must be converted to dense vectors of uniform dimension. To this end, this paper assigns a separate embedding matrix to each categorical field, initialized with trainable parameters, and maps each category ID to a real vector of length d. In practice, this paper sets the embedding dimension d=32,

balancing expressiveness and computational efficiency. For long-tail categories, regularization constraints or shared embedding strategies can be employed to prevent overfitting. After the embedding conversion, the embedding vectors from different fields are concatenated in field order to form a two-dimensional matrix input $X \in Rn \times d$, where n is the number of fields, and d is the embedding dimension. If the input features contain 20 fields totaling three categories: user, ad, and context, the final input dimension is 20×32. This matrix serves as the input to the TabTransformer, subsequently capturing the contextual relationships between fields.

Figure 1 shows the representation features of the six categories of features (gender, age, region, interest, ad ID, and device) learned after training with the full TabTransformer in the embedding space. Figure 1(a) uses t-Distributed Stochastic Neighbor Embedding (t-SNE) to perform two-dimensional dimensionality reduction on the embedding vectors of each feature category, visualizing their six clustering structures in the space. The horizontal and vertical axes represent the directions of the two principal components after embedding, and colors distinguish different fields. It can be observed that each feature category forms relatively close clusters in the embedding space, indicating that the model preserves the semantic structure between fields in its feature representation and that different categories have clear distinctions. Figure 1(b) shows the L2 norm distribution of the same batch of embedding vectors. The vertical axis represents the modulus of the embedding vectors. The numerical scales of different fields vary, indicating that these two types of features have higher expressive power or dynamic weight during training.

Specifically, for high cardinality category features (such as User ID and Ad ID), this paper does not employ label-based statistical methods such as target encoding or frequency encoding to avoid data leakage and failure issues in cold-start scenarios. Instead, it assigns an independent learnable embedding matrix to each high cardinality segment and uses a Xavier initialization strategy. For long-tail IDs with extremely low occurrence frequency (such as users or ads appearing only 1–2 times), a shared embedding bucket mechanism is introduced: mapping low-frequency IDs to a unified "UNK" embedding vector and constraining its update magnitude through L2 regularization, thereby effectively mitigating overfitting while preserving expressive power.

TABTRANSFORMER ENCODING STRUCTURE CONSTRUCTION

After achieving a unified representation of multi-source categorical features, this paper uses the TabTransformer structure as the core encoding module to perform deep modeling on the embedded feature matrix to explore high-order semantic relationships between fields [15,16]. The TabTransformer leverages a self-attention mechanism to establish dependencies between input fields, capturing nonlinear interaction structures that are difficult to explicitly model with traditional fully connected networks. To adapt to the feature scale and sparsity features of information flow advertising CTR prediction, this paper adjusts the structure and optimizes parameters based on the standard TabTransformer and embeds it into the entire end-to-end modeling process.

TRANSFORMER ENCODING MODULE CONSTRUCTION AND INPUT DEFINITION

The input matrix $H \in Rn \times d$ is generated by the embedding module in the previous stage, where n represents the number of categorical fields, and d is the embedding vector dimension. To preserve field-level structural information, this paper incorporates field position encoding into the embedding input. Unlike the absolute position encoding used in the original Transformer, this paper uses a learnable vector of field numbers as the position encoding and adds it element-wise to the embedded features. This operation preserves the input dimensionality while explicitly identifying the structural order of each field.

The Transformer encoding module consists of multiple stacked Transformer blocks, each of which contains a multi-head self-attention mechanism and a fully connected feedforward network. This paper adopts a twolayer stacked Transformer block structure to balance model expressiveness and computational efficiency. The number of attention heads in each layer is set to h=4; the hidden dimension is kept consistent with the embedding dimension, that is, d=32; the feedforward layer width is set to 4d=128. Residual connections and LayerNorm are used to ensure training stability. All Transformer parameters are trainable and initialized using the Xavier initialization strategy.

In the attention mechanism, the input vector is first mapped into query (Q), key (K), and value (V) vectors. The correlation weight matrix $A \in Rn \times n$ between fields is calculated using scaled dot-product attention, and then weighted summed with the value vector to update the representation of each field. The multi-head attention structure projects the input into multiple subspaces and processes them in parallel. The concatenated output is then linearly transformed. This mechanism allows the model to learn different types of feature dependencies from multiple representation subspaces, enhancing its ability to express nonlinear interactions.

ENCODING OUTPUT FUSION AND TRAINING ADAPTATION

After processing through multiple layers of Transformer blocks, the output is still a tensor $H \in Rn \times d$ with the same dimensions as the input, where each row corresponds to the contextual encoding representation of a field [17,18]. Since the subsequent CTR prediction task requires a vector representation of the semantic features of the entire sample, this paper uses field concatenation to concatenate all field representations in H into a one-dimensional vector $z \in Rn \cdot d$ for the downstream classification task. Compared to average

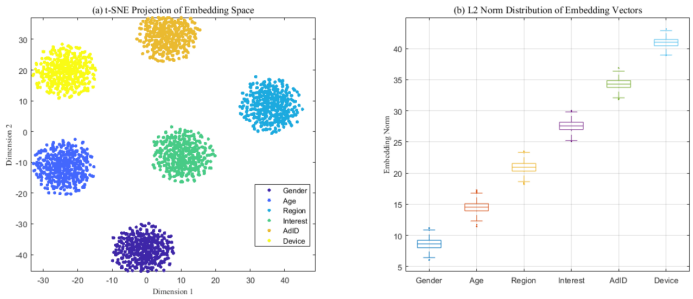

FIGURE 1. Visual analysis of categorical feature representation; Figure 1(a). t-SNE dimensionality reduction distribution of the embedding space; Figure 1(b). L2 norm distribution of embedding vectors for each category

pooling and max pooling, concatenation preserves more structural and positional information between fields and is suitable for medium-sized input scenarios with controllable feature dimensionality.

Figure 2 shows the encoding structure design of the TabTransformer, which is used for CTR prediction of information flow advertising. First, a unified embedding representation is created for multi-source categorical features, and field structure information is enhanced through a learnable positional encoding. This is then processed through two stacked Transformer blocks, each of which includes a 4-head selfattention mechanism (key dimension dk=8), scaled dot-product attention computation, residual connections, layer normalization, a 128-dimensional feedforward network (ReLU activation), and Dropout regularization (p=0.2). The output layer uses a field concatenation strategy to preserve structural information, generating an n·d-dimensional feature vector. Training utilizes gradient clipping and the AdamW optimizer to ensure stable model convergence. This architecture captures high-level semantic dependencies between fields through a dynamic attention mechanism.

MODELING FEATURE WEIGHTS WITH MULTI-HEAD ATTENTION MECHANISMS

After completing the feature input embedding and initializing the Transformer encoding structure, the model's modeling of semantic interactions between categorical features relies critically on the construction and optimization of the self-attention mechanism. This section focuses on the specific implementation of the multi-head attention mechanism in TabTransformer. By dynamically learning the attention weights between fields, a stable and generalizable feature interaction expression is constructed to improve the model's adaptability to complex contextual behavior structures.

SELF-ATTENTION WEIGHT CALCULATION MECHANISM

For each layer in the Transformer encoding structure, the input matrix $X \in Rn \times d$ is first projected into the query matrix $Q \in Rn \times dk$, the key matrix $K \in Rn \times dk$, and the value matrix $V \in Rn \times dv$ through linear mapping. Among them, dk=dv= 32/h, and h represents the number of attention heads. This paper sets h=4, that is, the dimension of each head is 8. The weight matrix is initialized with independent parameters to ensure that each head can capture independent semantic relationships in different representation subspaces. The attention weight matrix $A \in Rn \times n$ is calculated by scaling the dot product, as shown in Formula 1:

$$A = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right) \tag{1}$$

This weight matrix represents the importance distribution of each feature field to other fields. Softmax ensures that the sum of the weights of each row is 1, forming a standard probability distribution. Subsequently, the output matrix is obtained by weighted combination of the weights and V, as shown in Formula 2:

$$Z = A \cdot V \tag{2}$$

This operation completes the dynamic aggregation of semantics between fields and forms a contextenhanced feature representation.

MULTI-HEAD MECHANISM PARALLEL MODELING AND SEMANTIC FUSION

To improve the representation capability, the multi-head attention mechanism adopts a parallel structural design. Each attention head performs an independent linear transformation and attention calculation on the input matrix, learning semantic associations between fields from different subspaces. This paper uses h=4 heads, each with a dimension of 8. The outputs of the four heads are concatenated along the channel dimension and then mapped back to the original dimension of 32 through a linear transformation to maintain consistency with the embedding layer.

Figure 3 demonstrates the effectiveness of the model's multi-head attention mechanism in learning relationships between categorical features. Figure 3(a) is a heatmap of the average attention matrix of the first attention head across all samples, with the horizontal axis representing the focused key fields and the vertical axis representing the query fields. The visualization shows a clear focus on specific pairs of fields, demonstrating that the model is able to automatically capture strong dependencies between certain fields, such as the clear correlation between ad regions, user interests, and ad keywords. Figure 3(b) shows the distribution of attention

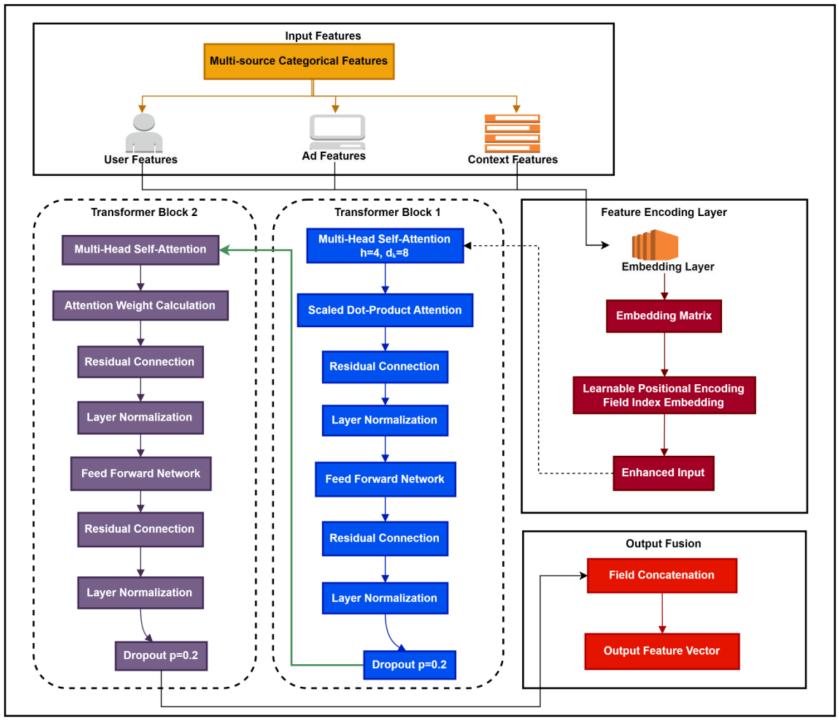

FIGURE 2. TabTransformer encoding structure design

sparsity across the four attention heads, with the horizontal axis representing the head number and the vertical axis representing the proportion of fields with a weight less than 0.05, indicating that the attention head assigns low weight to most fields. This demonstrates that different attention heads exhibit different sparsity, verifying the model's diverse attention allocation strategies, which enhance its representation and contextual awareness of multi-source categorical features.

To visually validate the model's ability to mine cross-source high-order semantics, the distribution of attention weights on specific feature pairs is further analyzed. For example, in multiple samples, the model's first attention head significantly enhances the association weight between "user interest tag = sustainable fashion" and "advertisement material = organic cotton" (the average attention score reaches 0.62, far higher than the random baseline of 0.05), while assigning extremely low weights (<0.08) to unrelated combinations (such as "interest = e-sports" and "material = organic cotton"). This phenomenon indicates that TabTransformer can automatically identify and strengthen the deep semantic alignment between user values and textile material properties, which is precisely the cross-domain high-order interaction that traditional embedding + MLP models struggle to explicitly model.

To systematically evaluate the robustness of the residual fusion mechanism under perturbation, this paper injects zero-mean Gaussian noise into the numerical features, scaling the standard deviation by a fixed proportion (0% to 20%) of the original standard deviation of each feature to maintain consistent relative perturbation intensity. Each noise level is independently repeated 10 times, and the mean of the LogLoss relative improvement rate is taken as the "fusion performance gain". The "feature importance" shown by the color bars in Figure 4 originates from the Gradient $\times$ Input method and is used to measure the local sensitivity of each numerical feature to the final prediction; it has been normalized using Z-score.

PARALLEL PATH FUSION OF NUMERICAL FEATURES

In the CTR prediction task of information flow advertising, in addition to a large number of categorical features, numerical features such as user behavior frequency, ad dwell time, and context position weight also carry key quantifiable information. Since the TabTransformer structure is mainly used to model discrete features, directly incorporating numerical features into the Transformer backbone may apply problems such as scale inconsistency and gradient perturbation, thereby affecting the stability and fusion effect of the entire model. Therefore, in addition to the TabTransformer main structure, this paper designs a parallel modeling path for numerical features and applies normalization, nonlinear mapping, and residual fusion mechanisms to effectively supplement and enhance the output of categorical features.

NUMERICAL FEATURE PREPROCESSING AND STANDARD NORMALIZATION

Numerical features must first undergo a unified scale transformation to avoid interference from distribution differences on model training. This paper uses standard normalization

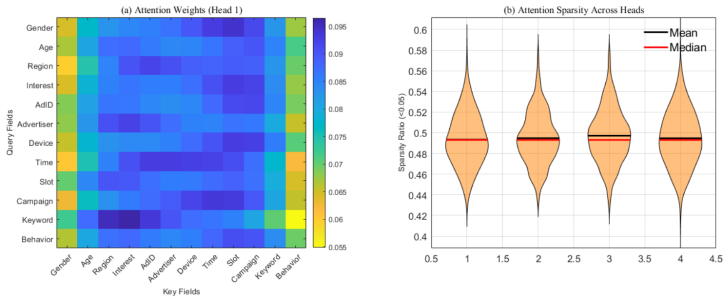

FIGURE 3. Multi-head attention weight distribution analysis; Figure 3(a). Single-head attention weight heatmap; Figure 3(b). Multihead attention sparsity distribution

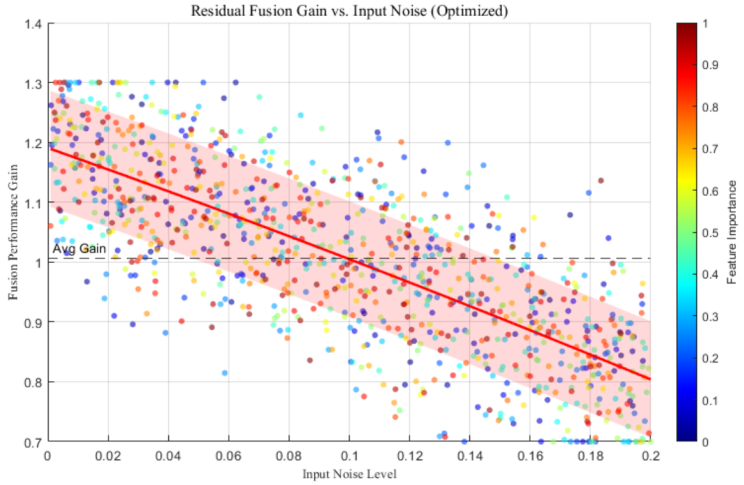

FIGURE 4. Relationship between residual fusion gain and input noise

(Z-score Normalization) for all numerical features, and the specific calculation is Formula 3:

$$x_i' = \frac{x_i - \mu}{\sigma} \tag{3}$$

xi represents the original feature value, and μ and σ are the mean and standard deviation of the feature in the training set, respectively. This processing can effectively eliminate the influence of dimension, making the eigenvalues stably distributed on the scale of zero mean and unit variance, and avoiding unstable gradient updates caused by eigenvalues that are too large or too small. In addition, for fields with many missing or outliers, a quantile truncation strategy (such as 1% and 99%) is used to smooth out the outliers to further improve the robustness of the model. The normalized numerical feature set constitutes the matrix Xnum∈ Rn×m , where n is the number of samples, and m is the number of numerical feature fields. In the experimental setting of this paper, the numerical features include a total of 12 items, covering the number of historical user exposures, number of clicks, context location information density, advertising bids, etc.

NUMERICAL FEATURE PATHWAY MODELING AND RESIDUAL FUSION

This paper chooses residual connection instead of simple concatenation or independent MLP fusion, mainly based on three considerations. First, numerical features and category embeddings are asymmetrical at the semantic level—the former are mostly behavioral statistics (such as historical click rates), while the latter encode structural identity information (such as user interest tags). Direct concatenation can easily lead to the attention mechanism of Transformer being dominated by numerical scale. Second, residual structure allows numerical paths to provide "incremental correction" rather than completely reconstructing representations, so that the fused features retain both the high-order interactive semantics captured by Transformer and the continuity prior of numerical features. Under input perturbation, residual fusion has higher robustness than concatenation fusion. Especially when category features are sparse or noise is high, the performance gain is more significant.

After the numerical features are normalized, they are input into an independent feedforward neural network module, which consists of two layers of fully connected structures with dimensions of [12, 64] and [64, 32], respectively. Each layer is followed by a RelU activation function and a Dropout operation (dropout rate of 0.2) to enhance the

nonlinear expression ability of the model and prevent overfitting. The network finally outputs Hnum∈Rn×d, whose dimension is consistent with the TabTransformer output and is set to d=32 to facilitate subsequent feature fusion operations. To ensure the stability of information from different sources in the gradient propagation during the fusion process, this paper uses a residual connection structure to integrate the numerical pathway and the Transformer backbone output. Assuming that the Transformer output is Hcat∈Rn×d, the fusion result is expressed by the following Formula 4:

$$H_{\text{fused}} = \text{LayerNorm}(H_{\text{cat}} + H_{\text{num}}) \tag{4}$$

This structure allows the numerical feature to retain its own representation ability while enhancing the compensatory expression between the categorical feature. In addition, LayerNorm normalization further stabilizes the activation distribution between different pathways, avoiding problems such as gradient unevenness and oscillation during the network training process. The fused representation Hfused is directly sent to the output layer for subsequent prediction. During training, parameters in both the categorical and numerical pathways participate in gradient updates, aligning the optimization objectives and ensuring collaborative learning of both types of information under a unified objective function. To address potential gradient explosion issues in the numerical pathway, this paper incorporates a gradient clipping mechanism (with a maximum norm set to 3.0) and sets a separate parameter set in the AdamW optimizer, giving the numerical pathway a small learning rate of 5×10^{-5} to maintain smooth learning.

To examine the contribution of the numerical pathway to overall performance, control groups with and without the numerical pathway are designed. In scenarios with high noise and long-tail categorical fields, the numerical path provides stable and robust supplementary information, effectively enhancing the prediction accuracy of the model under cold-start and data sparse conditions.

Figure 4 shows the relationship between fusion performance gain and input noise level, exploring the robustness of the residual fusion mechanism under feature perturbations. The horizontal axis represents the input feature noise level, ranging from 0 to 0.2, while the vertical axis represents the performance improvement achieved by the fusion model, with values roughly distributed between 0.7 and 1.3. The scatter plot shows a clear negative correlation, indicating that the performance gain from fusion decreases with increasing noise level. The red curve clearly captures this trend, while the red shaded area represents the confidence interval, revealing the uncertainty of the fitted model under different noise conditions. When the noise level approaches 0, the fusion gain is concentrated above 1.2, while when the noise level approaches 0.2, the gain drops to around 0.8, highlighting the residual fusion mechanism's greater adaptability to lownoise features. Furthermore, the dashed line indicates the global average gain of approximately 1.01, providing a benchmark for the model's performance improvement in real-world applications.

OUTPUT PREDICTION LAYER AND LOSS FUNCTION CONSTRUCTION

After completing the multi-source fusion modeling of category features and numerical features, the model needs to map the final feature representation to the CTR probability prediction value to complete the supervised learning task of click-through rate [19]. To this end, this paper designs an output prediction network composed of fully connected layers, combines the Sigmoid activation function to output the binary classification probability, and applies cross-entropy as the loss function to guide the optimization and update of model parameters.

FULLY CONNECTED OUTPUT STRUCTURE DESIGN AND PROBABILITY MAPPING

The fused feature representation is denoted as $H_{\text{fused}} \in R^{n\times d}$, where n represents the number of samples, and d=32 represents the fused feature dimension. This feature representation is used as input and fed into a single-layer fully connected network with weight parameter $W_{\text{out}} \in R^{d\times 1}$ and bias term $b_{\text{out}} \in R$. The intermediate representation is obtained through linear transformation, as shown in Formula 5:

$$y_{\text{logit}} = H_{\text{fused}} \cdot W_{\text{out}} + b_{\text{out}} \tag{5}$$

Then, the Sigmoid function is applied to perform nonlinear transformation on the logit value, and the click probability $\hat{y} \in R^n$ falling between [0,1] is output. The calculation formula is as shown in Formula 6:

$$\begin{aligned} \hat{y}_i &= \sigma\left(y_{\text{logit}}^{(i)}\right) \\ &= \frac{1}{1+e^{-y_{\text{logit}}^{(i)}}} \end{aligned} \tag{6}$$

This mapping ensures that the prediction result has interpretable probabilistic meaning and is suitable for the binary classification modeling requirements in the click rate prediction scenario. The Sigmoid function has numerical stability and can effectively avoid the gradient explosion problem of extreme values. At the same time, it ensures that the output is continuous and differentiable, providing conditions for the subsequent back propagation of the loss function. To improve the network's ability to distinguish different samples, no activation function is used before the output layer, and only the linear mapping structure is retained, thereby retaining the maximum expressive ability of the original feature to the output. During model training, the Dropout strategy is used to regularize the output layer input to weaken the strong dependency between neurons and prevent overfitting.

DEFINITION AND OPTIMIZATION MECHANISM OF LOSS FUNCTION

Given the severe imbalance between positive and negative samples in the CTR dataset (the click-through rate in this dataset is approximately 1.8%), using only standard cross-entropy can easily lead to a bias towards the negative class. Therefore, this paper introduces a class-weighted strategy into the loss function: setting the loss weight for positive samples (clicks) to α = 0.75, and the weight for negative samples to 1 - α = 0.25 (see Formula 8). This setting is determined through validation set tuning and can significantly improve the model's sensitivity to rare positive samples without excessively sacrificing the ability to distinguish negative samples. This strategy results in a faster decrease in loss during the early stages of training and a lower final LogLoss. This paper adopts the binary cross-entropy loss function as the supervisory signal for CTR prediction. Assuming that the true label is $y_i \in \{0, 1\}$; the predicted probability is $\hat{y}_i$; the total number of samples is n, then the total loss is Formula 7:

$$L_{\mathrm{CE}} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)\right] \quad (7)$$

This loss function still has a good gradient direction when the label distribution is unbalanced, and can amplify the penalty of the model on wrong predictions, thereby promoting the optimization effect of the model on click samples. To alleviate the model bias problem caused by the imbalance of the positive and negative sample ratio in the training set, this paper adopts a category weighting strategy in the training stage, applies a weighting factor α, and adjusts the loss perception intensity of positive and negative samples. The optimized loss function is shown in Formula 8:

$$\begin{aligned} L_{\mathrm{weighted}} = & -\frac{1}{n}\sum_{i=1}^{n}\Big[\alpha \cdot y_i \cdot \log(\hat{y}_i) \\ & + (1 - \alpha) \cdot (1 - y_i) \cdot \log(1 - \hat{y}_i)\Big] \end{aligned} \quad (8)$$

The value of α is set to 0.75 to increase the model's sensitivity to click samples. This strategy can significantly improve the model's ability to fit rare positive samples in extremely unbalanced CTR datasets. In addition, to improve the generalization performance of the model and suppress parameter overfitting, this paper applies the L2 regularization term as an auxiliary constraint in the loss function to prevent the model from overlearning the training data. The final loss function is Formula 9:

$$L_{\mathrm{final}} = L_{\mathrm{weighted}} + \lambda \cdot \|\Theta\|_2^2 \quad (9)$$

Θ represents the set of all trainable parameters of the model, and λ is the L2 regularization coefficient, which is set to 1×10^{-5}. During the experiment, this regularization term plays a key role in maintaining the convergence stability of the model and improving the generalization ability of unseen samples.

III. PERFORMANCE EVALUATION OF INFORMATION FLOW ADVERTISING CLICK PREDICTION

This study uses real-world business data from a large information flow advertising platform as an experimental dataset, encompassing multiple sources of features, including user click behavior, ad creatives, and context. The data spans a continuous month, with a sample size of over 100 million records, encompassing millions of unique users and hundreds of thousands of ad items. To ensure the fairness of model training and evaluation, the data is chronologically divided into a training set (70%), a validation set (15%), and a test set (15%) to avoid data leakage and sample overlap.

Category features include multi-dimensional sparse fields such as user ID, ad ID, interest tags, and device type, while numerical features include continuous variables such as ad impression frequency and historical click-through rate. All features undergo unified preprocessing: categorical features are encoded into fixedlength vectors, and numerical features are normalized to ensure compatibility with subsequent model inputs. The specific features used in this study are as follows: there are 20 categorical features, including: user-side features such as user ID, gender, age group (5 levels), region (province level), interest tags (Top-50 categories), and device type; ad-side features such as ad ID, advertiser ID, industry category (e.g., "textile raw materials," "garment"), and material tags (e.g., "cotton," "polyester," "recycled fiber"); context-side features such as display position (positions 1-10), time period (hourly granularity), page type, and network type. There are 12 numerical features, covering: user exposure count, click count, average CTR over the past 7 days, historical ad CTR, bid, average dwell time, as well as context position weight and time period popularity index. The data is strictly divided into training set (first 21 days), validation set (days 22–24), and test set (days 25–

30) in chronological order to prevent future information leakage. All time-series statistical features (such as "average click-through rate over the past 7 days" and "historical ad click-through rate") are constructed online using a sliding time window: for any sample timestamp t, its statistical features only aggregate historical behavior within the interval [t-7, t), and feature engineering during the training/validation/testing phases is completely isolated to ensure that the evaluation protocol conforms to real-world online deployment scenarios.

LOSS FUNCTION OPTIMIZATION AND MODEL CALIBRATION RESULTS

Figure 5(a) shows a comparison of the training convergence of three loss functions (standard cross-entropy, weighted cross-entropy, and one with L2 regularization). The horizontal axis represents the number of training epochs, and the vertical axis represents the loss value on a logarithmic

scale. It can be observed that the three curves converge to around 0.1 after around 50 epochs of training. However, after adding class weights (α = 0.75) and regularization, the overall loss level is lower, especially in the first 20 epochs, which demonstrates their effectiveness in alleviating sample imbalance and overfitting. Figure 5 (b) shows the expected calibration error (ECE) curve calculated based on 10 equal-frequency bins, with the horizontal axis representing the model's predicted probability and the vertical axis representing the actual click-through rate (i.e., true positive rate) within each bin. The dashed line y=x represents ideal calibration. The histogram overlay shows the sample distribution for each confidence interval. The overall ECE = 0.008, indicating that the model output is highly reliable.

To quantify the reliability of the model's output probabilities, this paper uses Expected Calibration Error (ECE) as an evaluation metric. ECE is calculated by dividing the predicted probabilities into M equal-frequency bins (M=10 in this paper) and calculating the weighted absolute deviation between the prediction accuracy and the average confidence level within each bin. Experimental results show that TabTransformer's ECE is only 0.008, significantly lower than DeepFM (0.021) and DCN (0.019), indicating that its predicted probabilities are highly close to the actual click frequency. This characteristic is crucial for advertising bidding on highvalue textiles—the platform can directly set bids based on the model's output probabilities without the need for an additional calibration module, thereby improving campaign efficiency while ensuring ROI.

AUC (AREA UNDER THE ROC CURVE)

AUC measures a model's ability to distinguish between positive and negative samples. Values closer to 1 indicate better ranking performance. During the specific evaluation, all samples are first sorted according to the CTR probability predicted by the model, and then the probability that the model ranks positive samples before negative samples is calculated. AUC does not depend on a specific threshold, so it can reflect the model's ranking ability at the overall level and is suitable for click data scenarios with unbalanced samples. In actual testing, batch evaluation is performed on the test set, and the prediction results of all batches are accumulated to calculate the global AUC score as the final metric. This method is suitable for recommendation systems that require high accuracy in ad ranking. The compared models include: TabTransformer, DeepFM (Deep Factorization Machine), DCN (Deep & Cross Network), xDeepFM (eXtreme Deep Factorization Machine), and Wide&Deep (Wide and Deep Learning)." Regular users" refers to non-new users who appear $\geq$5 times in the training set; "new users" refers to users who appear for the first time in the test set and have no behavioral records in the training set; "high-value users" is defined as active users whose cumulative clicks rank in the top 10% of all users; "mobile users" refers to users whose device type is a smartphone or tablet. All groups are determined independently based on their respective sets after time division to avoid cross-stage contamination.

Figure 6 illustrates the classification performance evaluation of recommendation models in the CTR prediction task. Figure 6(a) compares the ROC (Receiver Operating Characteristic) curves of different models across all users. The horizontal axis represents the false positive rate (FPR), and the vertical axis represents the true positive rate (TPR), measuring the model's overall discriminative ability. Figure 6(b) shows the AUC scores for five key user groups (such as high-value users and mobile users). The horizontal axis represents user groups, and the vertical axis represents the AUC value, reflecting the model's accuracy across different populations. As shown in Figure 6(a), the TabTransformer model significantly outperforms the other comparison models across the entire FPR range, achieving an AUC of 0.945, significantly higher than xDeepFM and Wide&Deep, demonstrating its stronger global classification capabilities. Figure 6(b) further demonstrates that TabTransformer achieves a high AUC of 0.917 in the high value user group, the highest among all models, demonstrating its strong ability to identify key demographics with commercial value. Furthermore, TabTransformer maintains its lead in relatively difficult-to-predict groups, such as new and mobile users, achieving an AUC of 0.785 for new users, significantly exceeding DeepFM's 0.743 and xDeepFM's 0.719, demonstrating excellent generalization and cold-start adaptability. In summary, TabTransformer not only achieves the best overall performance but also demonstrates a clear advantage in segmenting key users, demonstrating its robustness and practicality as a fusion model for ad recommendation tasks.

In addition to traditional CTR models (DeepFM, DCN, xDeepFM, Wide&Deep), this paper introduces two recent state-of-the-art CTR models based on self-attention as strong baselines: (1) AutoInt+, which uses interactive attention to explicitly model feature crossover; (2) FiBiNET, which combines bilinear interaction with SE attention mechanism. TabTransformer (AUC=0.945) still significantly outperforms AutoInt+ (0.921) and FiBiNET (0.928), validating the effectiveness of the multi-source unified representation and residual fusion architecture proposed in this paper.

LOGLOSS

LogLoss evaluates the degree of discrepancy between the model's predicted probabilities and the actual labels. The closer the CTR probability output by the model is to the actual label, the lower the loss. In the process, all samples in the test set are traversed, and the loss is accumulated based on their predicted values and actual click rates, then normalized by the total number of samples. This metric imposes a heavy penalty on mispredictions, particularly when the model predicts a click sample with an extremely low probability, resulting in a significant increase in loss. Therefore, LogLoss is often used to assess model fit quality and overfitting risk. In the context of ad click prediction, LogLoss helps refine the model's performance in estimating user click probability. "High CTR samples" refer to user-ad

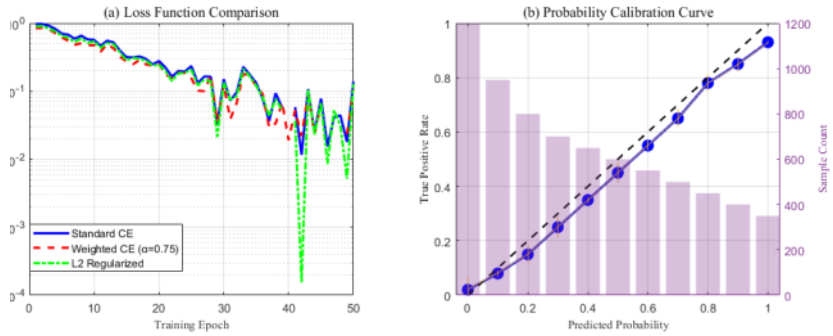

FIGURE 5. Loss function optimization and model calibration results; Figure 5(a). Comparison of training loss changes for different loss functions; Figure 5(b). Calibration curve of model prediction probability and sample distribution

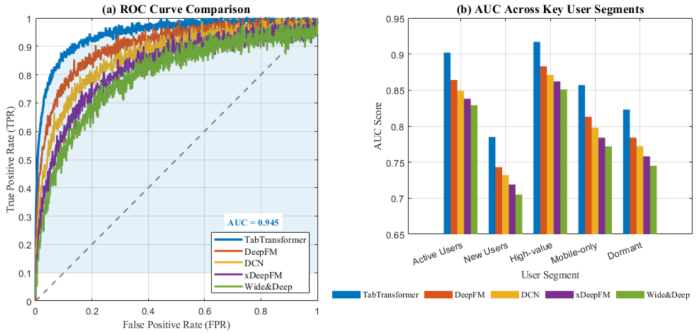

FIGURE 6. AUC performance comparison; Figure 6(a). Comparison of ROC curves of different models; Figure 6(b). Histogram of AUC scores for key user groups

pairs with a historical CTR $\geq$ 5%; "Low CTR samples" refer to long-tail interactions with a CTR $\leq$ 0.1%; "New users" are strictly limited to user IDs not appearing in the training set; "Long-tail ads" refer to ads ranking in the bottom 20% of impressions. All coldstart evaluations ensure that the corresponding IDs are completely invisible in the training set to realistically simulate the online scenario.

Table 1 shows the logarithmic loss (LogLoss) performance of five models in the ad click-through rate prediction task under different scenarios. This measure measures the error between the model's predicted probability and actual click behavior. Overall, the TabTransformer model achieves the lowest LogLoss across all evaluation dimensions (0.352 overall), significantly outperforming traditional architectures such as DeepFM (0.381) and DCN (0.372), demonstrating its greater stability and more accurate fit in probability estimation. In high-click samples, TabTransformer's loss is 0.412, significantly lower than Wide&Deep's 0.462, indicating that it more precisely captures the behavior of highly interested users. In low-click samples, the model's loss is only 0.298, the only model below 0.3, demonstrating its good adaptability to sparse click signals. In cold-start scenarios, TabTransformer achieves a LogLoss of 0.402 for new users, surpassing DeepFM's 0.438 and xDeepFM's 0.419, demonstrating its stable output capabilities even during cold-start scenarios. Furthermore, TabTransformer achieves the lowest loss (0.368) for long-tail ad prediction, further demonstrating its robustness and generalization capabilities when dealing with data with low exposure and uneven distribution.

PRECISION@K (TOP-K CLICK ACCURACY)

Precision@K primarily measures the model's click prediction accuracy in the Top-K recommendation list. The evaluation process involves first generating a list of candidate ads for each user sample, sorting them by model prediction score, and selecting the top K ranked ads. These ads are then compared with actual click data to calculate the proportion of clicked ads among the top K ranked ads. This metric emphasizes the toporder accuracy of the results and is suitable for scenarios with limited ad placement and requiring precise recommendations. By setting different K values (such as 5 and 10), a multi-level evaluation is performed to reflect the model's effectiveness at varying recommendation strengths.All Precision@K and Recall@K evaluations are based on the entire candidate set (i.e., all ads displayed to each user in the test set), without negative sampling. For each display event, the model scores and sorts all ads visible to the user that day, and the Top-K results are compared with actual click behavior to calculate the metric. This protocol faithfully reflects the actual operating logic of online recommendation systems, avoiding bias introduced

TABLE 1. LogLoss performance comparison

Model	Overall LogLoss	High-Click LogLoss	Low-Click LogLoss	New User LogLoss	Long-Tail Ad LogLoss
TabTransformer	0.352	0.412	0.298	0.402	0.368
DeepFM	0.381	0.452	0.325	0.438	0.398
DCN	0.372	0.441	0.318	0.427	0.388
xDeepFM	0.365	0.432	0.312	0.419	0.378
Wide&Deep	0.389	0.462	0.332	0.447	0.405

by sampling. Figure 7 analyzes the accuracy performance of each model at different recommendation list lengths (K values) and the model's recommendation performance for various ad types. Figure 7(a) shows the Precision@K curves of five recommendation models for K=1 to K=100, with the horizontal axis showing the K value (logarithmic scale) and the vertical axis showing the corresponding accuracy. It can be seen that the proposed TabTransformer model outperforms the comparison models at all K values, especially when K is small (K=1, P@1=0.582), surpassing DeepFM (0.543) and DCN (0.538), indicating that this model achieves higher accuracy in top recommendations. As K increases, the accuracy of each model gradually decreases, but TabTransformer maintains a consistent lead. Figure 7(b) shows the recommendation accuracy for different ad types when K is fixed at 10. The results show that the model performs well in categories such as education (P@10 = 0.487), finance (0.463), and food (0.456), while performance is relatively poor in travel and sports. Overall, this model demonstrates strong stability and adaptability across various K values and domains for ad recommendation tasks.

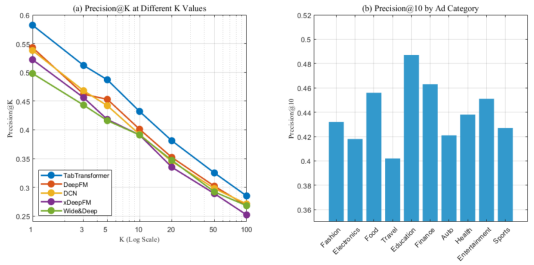

FIGURE 7. Evaluation results of model recommendation accuracy; Figure 7(a). Precision@K comparison of models at different K values; Figure 7(b). Precision@10 distribution for different ad types

RECALL@K (TOP-K CLICK RECALL)

Recall@K measures the proportion of all actual clicked ads that are accurately identified and ranked in the model's Top-K recommendations. The evaluation process first determines each user's set of actual clicked ads and then calculates how many of these ads appear in the model's Top-K recommendations. This metric focuses on the extent to which users' actual interests are covered, reflecting the model's recall ability. Compared to Precision@K, Recall@K focuses more on the overall covered click volume and is suitable for evaluating the comprehensiveness of the model's capture of user behavior. It can effectively reflect the model's promotion breadth, especially in long-tail advertising and cold-start users. Table 2 shows the Recall@K performance of different models across multiple user groups and at different K values. The recommendation accuracy of the five models is compared across four groups: the general population, new users, high-value users, and mobile users. Among the general population, TabTransformer achieves the highest Recall@K value of 0.521 when K=10, significantly outperforming DeepFM (0.488) and DCN (0.498), demonstrating its stronger ranking capabilities. For the new user group, the model maintains its lead at K=10, with a Recall@K of 0.398, a significant improvement over the traditional Wide&Deep (0.358), demonstrating its strong adaptability to cold-start problems. Among high-value users, TabTransformer performs particularly well at K=10, achieving a Recall@K of 0.568, significantly ahead of DCN (0.545) and xDeepFM (0.538). This demonstrates its superior accuracy in identifying commercially valuable users. The model also maintains its lead among mobile users, with a Recall@K of 0.538.

TABLE 2. Recall@K performance comparison (K = 5, 10)

User Group	K	TabTransformer	DeepFM	DCN	xDeepFM	Wide&Deep
Overall	5	0.412	0.382	0.392	0.385	0.378
	10	0.521	0.488	0.498	0.491	0.481
New Users	5	0.285	0.252	0.263	0.256	0.248
	10	0.398	0.362	0.374	0.367	0.358
High-value Users	5	0.452	0.421	0.432	0.425	0.418
	10	0.568	0.532	0.545	0.538	0.528
Mobile-only Users	5	0.428	0.396	0.407	0.4	0.392
	10	0.538	0.502	0.515	0.508	0.498

IV. CONCLUSION

To address the persistent problem of insufficient contextual relevance among category features in CTR prediction for information flow advertising—a challenge especially acute when marketing diverse items like textile materials and finished goods—this paper focuses on a multi-source feature fusion method based on TabTransformer. Specifically, it designs a feature interaction modeling framework combined with a multihead attention mechanism to enhance the semantic coherence of features related to the specialized characteristics of the textile industry's products. By uniformly representing multi-source category features and applying a self-attention mechanism to dynamically capture high-order semantic dependencies between fields, the model's adaptability to complex user interests and context changes is effectively improved. Furthermore, a residual path fusion strategy combining numerical features enhances the model's ability to model mixed data types. Finally, the

fused features are fed into a fully connected layer for CTR probability prediction, and the cross-entropy loss function is used to optimize parameters, achieving end-to-end performance improvement. Experimental results demonstrate that the proposed TabTransformer model performs exceptionally well in CTR prediction, achieving an AUC of 0.945 and the lowest overall LogLoss value of 0.352. Furthermore, when K=10, the Recall@K reaches 0.521 across the entire population, demonstrating its real-world effectiveness in recalling relevant advertisements, a crucial factor for ensuring niche products like specialized textile equipment are seen. This high recall is coupled with an excellent performance in reliability, evidenced by a probability calibration error of only 0.008, further demonstrating the model's overall strength in both coverage and the trustworthiness of its predictive scores for advertising decisions within the textile sector.

AUTHOR CONTRIBUTIONS

Conceptualization – Xia Yan, Anita Binti Rosli and Aryaty Binti Alwie; methodology – Xia Yan, Anita Binti Rosli and Aryaty Binti Alwie; investigation – Xia Yan, Anita Binti Rosli and Aryaty Binti Alwie; writing-original draft preparation – Xia Yan, Anita Binti Rosli and Aryaty Binti Alwie. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This research received no external funding.

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] W. M. Lim, S. Gupta, and A. Aggarwal, "How do digital natives perceive and react toward online advertising? Implications for SMEs," Journal of Strategic Marketing, vol. 32, no. 8, pp. 1071-1105, 2024, doi: 10.1080/0965254X.2021.1941204.

[2] L. F. Lina and L. Ahluwalia, "Customers' impulse buying in social commerce: The role of flow experience in personalized advertising," Jurnal Manajemen Maranatha, vol. 21, no. 1, pp. 1-8, 2021, doi: 10.28932/jmm.v21i1.3837.

[3] J. F. Hair Jr and M. Sarstedt, "Data, measurement, and causal inferences in machine learning: opportunities and challenges for marketing," Journal of Marketing Theory and Practice, vol. 29, no. 1, pp. 65-77, 2021, doi: 10.1080/10696679.2020.1860683.

[4] K. Alshaketheep, A. M. Mansour, M. M. Al-Ma'aitah, M. N. Dabaghia, and Y. M. Dabaghie, "Leveraging AI predictive analytics for marketing strategy: The mediating role of management awareness," Journal of System and Management Sciences, vol. 14, no. 2, pp. 71-89, 2024, doi: 10.33168/JSMS.2024.0205.

[5] A. W. Luangrath, J. Peck, W. Hedgcock, and Y. Xu, "Observing product touch: The vicarious haptic effect in digital marketing and virtual reality," Journal of Marketing Research, vol. 59, no. 2, pp. 306-326, 2022, doi: 10.1177/00222437211059540.

[6] A. Rosário and R. Raimundo, "Consumer marketing strategy and E-commerce in the last decade: a literature review," Journal of theoretical and applied electronic commerce research, vol. 16, no. 7, pp. 3003-3024, 2021, doi: 10.3390/jtaer16070164.

[7] S. Wang, C. Hu, and G. Jia, "Deep Learning-Based Saliency Assessment Model for Product Placement in Video Advertisements," Journal of Advanced Computing Systems, vol. 4, no. 5, pp. 27-41, 2024, doi: 10.69987/JACS.2024.40503.

[8] X. Wang, Y. Qiao, J. Xiong, Z. Zhao, N. Zhang, and M. Feng, "Advanced network intrusion detection with tabtransformer," Journal of Theory and Practice of Engineering Science, vol. 4, no. 03, pp. 191-198, 2024, doi: 10.53469/jtpes.2024.04(03).18.

[9] A. Kalidindi and M. B. Arrama, "A tabtransformer based model for detecting botnet-attacks on internet of things using deep learning," Journal of Theoretical and Applied Information Technology, vol. 101, no. 13, pp. 5206-5218, 2023.

[10] E. Häglund and J. Björklund, "AI-driven contextual advertising: Toward relevant messaging without personal data," Journal of Current Issues & Research in Advertising, vol. 45, no. 3, pp. 301-319, 2024, doi: 10.1080/10641734.2024.2334939.

[11] P. Lorenz-Spreen, L. Oswald, S. Lewandowsky, and R. Hertwig, "A systematic review of worldwide causal and correlational evidence on digital media and democracy," Nature human behaviour, vol. 7, no. 1, pp. 74-101, 2023, doi: 10.1038/s41562-022-01460-1.

[12] N. Ihzaturrahma and N. Kusumawati, "Influence of integrated marketing communication to brand awareness and brand image toward purchase intention of local fashion product," International Journal of Entrepreneurship and Management Practices, vol. 4, no. 15, pp. 23-41, 2021, doi: 10.35631/IJEMP.415002.

[13] E. Afful-Dadzie, A. Afful-Dadzie, and S. B. Egala, "Social media in health communication: A literature review of information quality," Health Information Management Journal, vol. 52, no. 1, pp. 3-17, 2023, doi: 10.1177/1833358321992683.

[14] R. Andriani, D. P. E. Pratiwi, and I. D. A. D. M. Santika, "Verbal and Non-Verbal Signs in Facial Wash Advertisements: A Semiotic Analysis," Yavana Bhasha: Journal of English Language Education, vol. 4, no. 2, pp. 24-29, 2021, doi: 10.25078/yb.v4i2.2768.

[15] H. Hu, Z. Lin, Q. Hu, Y. Zhang, W. Wei, and W. Wang, "Multi-source information fusion based DLaaS for traffic flow prediction," IEEE Transactions on Computers, vol. 73, no. 4, pp. 994-1003, 2023, doi: 10.1109/TC.2023.3236902.

[16] J. Liu, T. Li, S. Ji, P. Xie, S. Du, and F. Teng, "Urban flow pattern mining based on multi-source heterogeneous data fusion and knowledge graph embedding," IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 2, pp. 2133-2146, 2021, doi: 10.1109/TKDE.2021.3098612.

[17] M. Li, F. Wang, X. Jia, W. Li, T. Li, and G. Rui, "Multi-source data fusion for economic data analysis," Neural Computing and Applications, vol. 33, no. 10, pp. 4729-4739, 2021, doi: 10.1007/s00521-020-05531-0.

[18] F. Yang and P. Zhang, "MSIF: Multi-source information fusion based on information sets," Journal of Intelligent & Fuzzy Systems, vol. 44, no. 3, pp. 4103-4112, 2023, doi: 10.3233/JIFS-222210.

[19] G. Tong, Z. Li, H. Peng, and Y. Wang, "Multi-source features fusion single stage 3D object detection with transformer," IEEE robotics and automation letters, vol. 8, no. 4, pp. 2062-2069, 2023, doi: 10.1109/LRA.2023.3244124.