TBformer: Multi-scale Transformer with Time-Behavior Attention for Multi-modal Customer Churn Prediction

Author1, Author2

1. MIT
2. CMU

**Abstract:**

In highly competitive market of Internet service platforms, identifying and retaining potential churners through customer churn prediction techniques is crucial for maintaining platform vitality. The sequences of interaction behaviors between customers and platforms are closely related to churn prediction results. However, existing methods focus only on capturing the temporal dependencies in dynamic behavior sequences while ignoring the correlations between different behaviors. Moreover, classical methods apply only to static data, while deep learning-based methods focus on dynamic data, neither leveraging the complementary information between static and dynamic data. To address these issues, we propose a multi-modal customer churn prediction model based on Transformer with multi-scale Time-Behavior attention, TBformer, which adaptively fuses static and dynamic data. Time-Behavior module can capture multi-scale temporal dependencies and behavioral correlations in behavioral time series across time and behavior dimensions. We perform behavior-independent multi-scale dynamic feature fusion through bidirectional connection paths. Furthermore, the multi-modal fusion module based on the attention mechanism adaptively controls the fusion weights of static and dynamic features to improve performance. Extensive experiments on two publicly available datasets, KKBox and KDD, and a private dataset, HOF, demonstrate that our TBformer achieves an average AUC of 91.2% (+2.47%), outperforming the state-of-the-art customer churn prediction methods.

Keywords: Customer Churn Prediction, Time Series, Multi-modal, Multi-scale, Transformer

1.Introduction

With the rapid development of mobile Internet, consumers have an increasing number of service platforms to choose from. All Internet service platforms compete for more active users, which represent a predictable revenue stream. However, the cost of acquiring a new user is typically five to six times higher than retaining an existing one [20]. Obviously, platforms prefer to maintain a stable user base with lower operational costs. Customer churn prediction techniques leverage historical data to accurately identify high-risk users and detect churn trends. Consequently, platforms can strategically allocate marketing resources and proactively offer incentives in advance to users with a higher likelihood of churn, thereby enhancing the overall number of active users. Customer churn prediction has already attracted significant research attention and has been applied in various service industries, such as e-commerce[2], social networks [4], and online games [5]. Taking e-commerce platforms as an example, they face unique challenges with their large, dispersed customer base and lack of face-to-face interactions between businesses and customers [34]. Utilizing churn prediction to identify potential churners helps adjust subsequent marketing strategies and provides more precise personalized services and recommendations. Therefore, customer churn prediction has become a research hotspot in the consumer technology domain.

Customer churn prediction is a binary classification task that predicts the likelihood of churn based on customers’ personal information and historical behaviors [1]. Customer data can be categorized into two types based on data sources [6]: static data such as age, gender, and historical statistics, as well as time series dynamic data generated from customers’ interactive behaviors with entities. Classical methods struggle to extract complex high-dimensional features from dynamic data. With the recent advancements in deep learning for time series analysis [11][12][13][14][15][16], researchers utilize sequence models like Recurrent Neural Networks (RNN) [17] and Transformers [18] to extract features from dynamic data for analyzing customer behavioral patterns. Due to the frequent interactions between users and entities on service platforms, users’ different behaviors at the same time have logical intentions, and the same behaviors across different times exhibit trends and inertia. However, deep learning methods [19], while focus on capturing the variation trends of the same behavior across different times, fail to consider the mutual influences between different behaviors.

Moreover, current mainstream methods mostly focus on dynamic data with rich information and high potential value, while overlooking static data. Static data contains customers' background information, while dynamic data reflects trends of customer interest and real-time states of their behaviors. Due to the inherent complementarity between static and dynamic data, integrated analysis allows for a more comprehensive understanding of customers within limited data. However, to our knowledge, only Wu et al. [20] explicitly perform multi-modal feature fusion for customer churn prediction. Nevertheless, they directly fuse static and dynamic features through vector concatenation without considering the cross-modal interactions between different modalities. In short, existing methods have two main limitations [21]: failure to mine behavioral correlations within dynamic data, and lack of effective integration between static and dynamic data. These limitations severely restrict the accuracy of customer churn prediction.

To address the challenges above, we propose TBformer, a novel multi-modal Transformer network with Time-Behavior attention for customer churn prediction. TBformer not only models temporal dependencies and behavioral correlations in dynamic data but also effectively integrates static and dynamic data. The dynamic data changes over time, and fusing sequences at different scales can accurately reflect both global and local information. Therefore, we design behavior-independent patch embedding and time-behavior attention to extract multi-scale dynamic features. We also utilize DeepFM to extract static features. The multi-scale fusion module and multi-modal fusion module utilize the attention mechanism to calculate the cross-influence among different features. They dynamically control the weights of each feature in the fused results. Inspired by the Feature Pyramid Network (FPN), we conduct behavior-independent multi-scale feature fusion through a bidirectional connection that combines bottom-up and top-down paths. Finally, a linear classification head maps the multi-modal features to a churn probability output. The main contributions of this work can be summarized as follows:

1、TBformer is proposed for multi-modal customer churn prediction. It simultaneously extracts features from static and dynamic data and performs comprehensive multi-modal feature fusion.

2、Time-behavior attention is proposed to extract multi-scale features from dynamic data. It captures both the temporal dependencies and behavioral correlations along the time and behavior dimensions, respectively.

3、A behavior-independent multi-scale fusion module is proposed for dynamic multi-scale feature fusion through bidirectional connection paths, and a multi-modal fusion module is proposed for static-dynamic feature fusion. These modules adaptively calculate the feature interactions based on the attention mechanism.

4、The proposed TBformer is compared with several advanced models on two publicly available real-world datasets and a private dataset to demonstrate its superiority.

1. Related work

In 1995, Keaveney [22] first proposed a clear definition of customer churn: the behavior transition of a customer who no longer purchases or uses the relevant product or terminates the original service. Current customer churn prediction methods can be categorized into classical and deep learning-based methods depending on the feature extraction techniques.

2.1 Classical methods

Classical methods rely on feature engineering to extract features from static data such as personal information and consumption records. They then employ machine learning models like logistic regression [7], decision trees [8], and support vector machines (SVM) [9] to model and analyze the extracted features. Xie et al. [23] propose an Improved Balanced Random Forest (IBRF) by altering the class distribution and imposing higher penalties for misclassifying the minority class to learn the optimal features iteratively. But random forests are susceptible to input noise. To solve this problem, Stripling et al. [24] construct a classifier based on the Expected Maximum Profit (EMP) metric and use a Genetic Algorithm (GA) to train a logistic regression model. Subsequently, Höppner et al. [25] incorporate the EMP metric into the decision tree node splitting criterion and achieve a significant profit improvement compared to the tree method driven by classification accuracy. However, compared to individual models, researchers have gradually discovered that applying ensemble algorithms to comprehensively analyze results from multiple classification models can also effectively enhance performance. Therefore, Ran et al. [10] use SVM, Adaboost, Random Forest, and XGBoost models to identify churning airline customers. Lalwani et al. [9] consider feature selection using the Gravitational Search Algorithm and apply boosting and ensemble techniques to observe the performance of their models.

With the development of the Internet, customer data now contains a large number of behavioral interaction records. However, classical methods are only appropriate for non-numerical input. For continuous dynamic data, they ignore the temporal dependencies of customer behaviors and thus struggle to extract rich high-dimensional features. Therefore, researches based on deep learning have been promoted in recent years.

2.2 Deep Learning-based methods

The predictive performance of deep learning methods largely depends on the underlying architecture and depth of the sequence models. Kwon et al. [30] design a model based on RNN architecture, using log data and message data to predict customer churn in digital medical applications. Since RNN can't handle long sequences, Yang et al. [26] construct an Long Short-Term Memory (LSTM) network and combine the clustering results to predict customer churn in social applications. Zheng et al. [27] utilize LSTM to extract the temporal dependencies within the login activity sequence and employ a time-aware filtering component to extract user behavior patterns from in-game behaviors. Then, they combine and map them to the prediction results. Considering the gradient vanishing and exploding problems in LSTMs, Yin et al. [28] utilize a Transformer instead of LSTM to predict customer churn in Massive Open Online Courses (MOOC). Panimalarf et al. [31] enhance deep feature extraction using a deep artificial neural network (ANN) for feature extraction and a Weight Updated Tuned Naïve Bayes classifier to predict customer churn in a cloud environment. These methods only analyze dynamic data without static data. Xu et al. [29] utilize static data and dynamic data of merchants' daily transaction flows as the Gated Recurrent Unit (GRU) input to predict merchant churn. Wang et al. [37] and Liu et al. [38] transform static data into an embedding format consistent with dynamic data through one-hot encoding, and combine LSTM and Convolutional Neural Network (CNN) to enhance feature extraction capabilities. However, they employ only simple data fusion, leading to information redundancy. In constract, Wu [20] utilize two Transformer networks to extract correlations between different times and behaviors. Then, they perform vector concatenation of the extracted dynamic features and static features to predict the churn of browser customers.

Although research on customer churn prediction has progressed, current researches have focused on analyzing dynamic data, ignoring static data. Moreover, the self-attention mechanism of Transformer calculates global attention scores. However, the significance of other behaviors at the same time or the same behavior at different times is greater than that of the others.

3.Methodology

Static data contains long-term inherent attributes, while dynamic data exhibits short-term behavioral trends. Therefore, we employ different modules to extract features based on the characteristics of each data type to analyze the impacts of long-term and short-term factors on customer churn. Figure 1 shows the overall architecture of the proposed TBformer. DeepFM extracts static features . Meanwhile, the time-behavior module extracts dynamic multi-scale features from both the time and behavior dimensions while preserving global and local information. The multi-scale fusion module utilizes the attention mechanism to bidirectionally fuse the multi-scale features, obtaining the final dynamic features . The multi-modal fusion module overcomes the limitations of existing methods by efficiently integrating static and dynamic features. Finally, a linear classifier is utilized to obtain the prediction results.

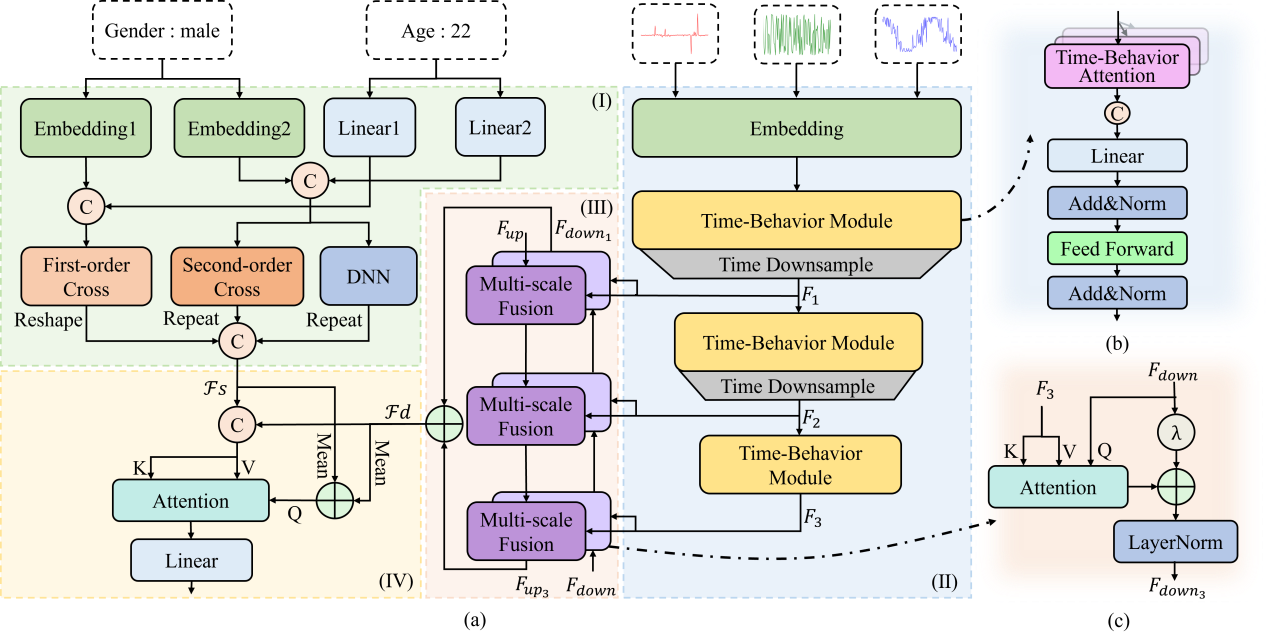


Figure 1. (a)The architecture of TBformer: Static data is input into DeepFM ((I)green shading) to obtain static features , while dynamic data is input into the time-behavior module ((II)blue shading) to generate multi-scale dynamic features . The multi-scale fusion module ((III)pink shading) bidirectionally fuses to obtain dynamic features . Subsequently, the multi-modal fusion module ((IV)yellow shading) integrates static features and dynamic features , and finally predicts the probability of customer churn through a linear layer. (b)The architecture of time-behavior module. (c)The architecture of bottom-up multi-scale fusion module.

3.1 DeepFM on Static Data

Static data represents customers' inherent attributes, characterized by low dimensionality and infrequent temporal variations. Categorical data within it (such as occupation, interests, etc.) becomes sparse after embedding. DeepFM is excellent for processing sparse data and explicitly capturing the interaction feature, demonstrating superior feature extraction capabilities in click-through rate prediction tasks for recommendation systems [32]. Therefore, we utilize DeepFM to extract static features. and represent the categorical and numerical data in the static input, respectively. As shown in Figure 1 (I), and are mapped to different continuous spaces through two embedding layers and linear layers:

where and map the discrete values of each category to and spaces, respectively. ,. is the embedding dimension. ,,, are the learnable weights and biases. ,. The vectors in the same dimensional space are concatenated. Subsequently, it performs first-order cross to learn the linear relationships between features in a one-dimensional space, and performs second-order cross and DNN mapping to learn complex nonlinear patterns in a two-dimensional space:

where, , represents vector concatenation. is a first-order cross feature. is a second-order cross feature. represents to flatten the matrix. represents two fully connected layers. is a deep feature. , and obtained through DeepFM, are concatenated to obtain the static features :

where, , . can measure the direct impact of individual features, such as how a user's age may directly influence churn probability. captures explicit feature interactions, for instance, the combination of age and subscription plan may reveal churn risks for specific user groups. learns implicit deep-level patterns. By concatenating these features, TBformer balances the interpretability of static data with the deep modeling capability of complex features, making it more suitable for customer churn prediction task.

3.2 Time-Behavior Attention on Dynamic Data

In dynamic data, different behaviors at the same time exhibit logical intentions, and same behaviors across different times contain trends and periods. To simultaneously consider both temporal and behavioral dimensions in time series, we design a dynamic feature extraction module. It can mine the deep features in the behavior dimension and obtain multi-scale information. As shown in Figure 1 (II), it consists of three submodules: behavior-independent embedding, time-behavior module, and time downsampling.

**Behavior-independent embedding**. Different behavioral data have distinct semantics, while sharing the same feature space can lead to information confusion and overlooking the important features. Therefore, we design behavior-independent embedding to enhance feature learning capabilities in Figure 2. Assuming the total number of days in the input data is , and there are different behaviors. The sequential behavior data can be denoted as , where represents the statistical results of the behavior on the day. By dividing the single behavior sequence along time dimension, we obtain patches, where the patch of the behavior is . We use a set of linear transformations to map } to embedding vectors:

where, and are the weights and biases used to embed the patch of the behavior. Stacking the embeddings in the original order of time and behavior dimensions, we obtain the embedded coding }. Since the self-attention mechanism lacks the ability to capture sequential information, positional encoding is also necessary to capture location information in both time and behavior dimensions. The final embedding results are as follows:

where, are the learnable parameters.

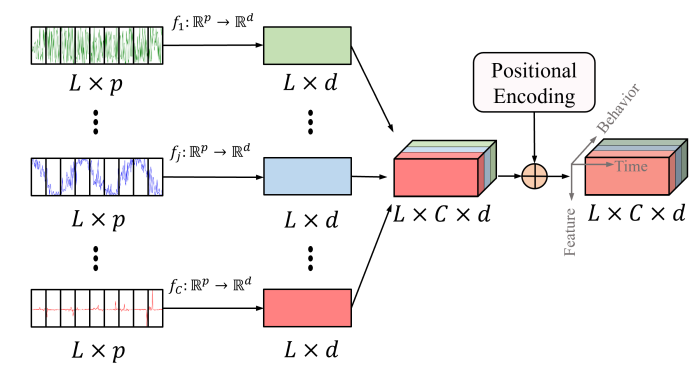


Figure 2 Behavior-independent Embedding. Each behavior time series is embedded independently.

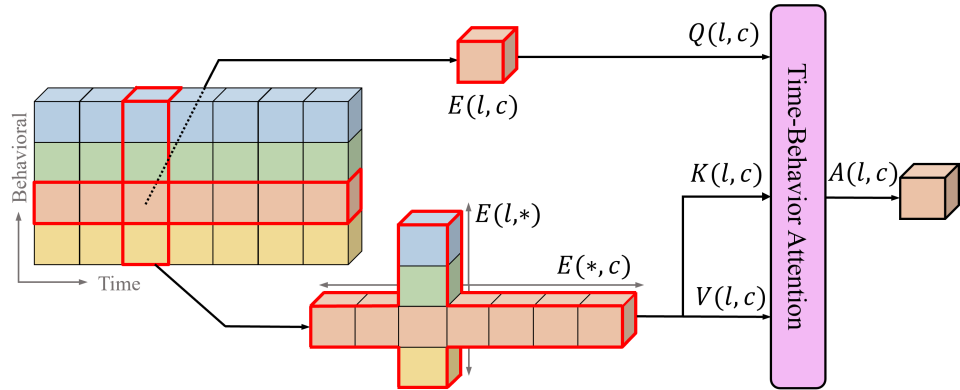


Figure 3 Time-Behavior Attention. It simultaneously extracts temporal relationships and behavioral relationships and performs Softmax independently along the time and behavior dimensions.

**Time-behavior module.** The time-behavior module based on Transformer is shown in Figure 1 (b). Since customers’ behaviors exhibit stable logic, the patches within the same time or behavioral dimension tend to have stronger associations compared to others. Therefore, we design the time-behavior attention mechanism in Figure 3. It focuses on the patches with the same time dimension or behavior dimension as the current patch to improve the effectiveness. represents the patch embeddings of the behavior at different times and represents the patch embeddings of different behaviors at the time point. The attention calculation is as follows:

where are the shared weights. and is the number of attention heads. is obtained from . .

For different patches and , there are numerous repeated calculations that occur when computing different attention results  and in Equations（11-14）. Therefore, the computation process can be adjusted to achieve parallel computation acceleration. First, the patch embeddings are mapped as follows:

where, . Then the attention scores are calculated along the time dimension and the behavior dimension respectively:

where represents the attention scores of for . represents the attention scores of for .

According to Equation（14）, the  operation, like in the global attention mechanism, should be performed on all dimensions. However, the temporal dimension and behavior dimension usually have different distributions of attention scores. Therefore, our designed  operation is performed independently on the time and behavior dimensions, respectively.

where, and are the attention scores after the operation.

Finally, we adjust Equation（14） to get the time-behavior attention outputs:

**Time downsampling:** Dense sampling captures customers' short-term interests, such as a sudden decrease in recent visits. Sparse sampling observes long-term behavioral trends, such as a gradual decline in activity. Therefore, we design a stackable time downsampling module to extract short-term changes and long-term trends:

where, , ,. are the learnable shared weights.

When is not a multiple of 2, we expand along the time dimension, fill with , and then conduct time downsampling:

In this case, we obtain double observation scales in the time dimension. Stacking time downsampling module multiple times can obtain features on multiple different time scales for dynamic input data.

3.3 Multi-scale Fusion

Inspired by the FPN network, we design behavior-independent multi-scale feature fusion module based on bidirectional connection paths. Bottom-up modeling preserves fine-grained information of short-term behaviors and infers long-term trends through hierarchical aggregation, making it suitable for cold-start users. Top-down modeling captures short-term anomalies from long-term trends, ensuring robust prediction. The bidirectional connection path combines bottom-up and top-down paths, reducing noise interference and improving the accuracy of customer churn prediction.

The multi-scale dynamic features extracted by the time-behavior module in Figure 1 (III) from top to bottom are denoted as . As shown in Figure 1 (c), we define the initial fusion feature . Then, in the order of , we sequentially integrate each scale feature into the fusion feature of different scales。Finally, is the bottom-up fusion output. The multi-scale fusion module leverages an attention mechanism to dynamically assign weights across different time scales, focusing on the most significant features.

where is the feature at the layer and is initialized as a zero vector. are the fusion features before and after . represents the scaled dot-product attention. is the weight of the attention mechanism, which is shared across the fusion path. represents flattening the dimension from to . The hyperparameter is used to control the proportion of scale features.

Since the different scales of multi-scale temporal features are along the time dimension, it is not necessary to repeat the attention calculation in the behavioral dimension. Therefore, we design corresponding behavior-independent paths for bidirectional connections. As shown in Figure 5, multi-scale fusion is performed independently on each behavioral sequence in the order . First, features at each scale are separated along the behavioral dimension into . Then, multi-scale features corresponding to each behavior are fused through multi-scale fusion to obtain . Finally, the concatenated is the bottom-up behavior-independent multi-scale fusion result.

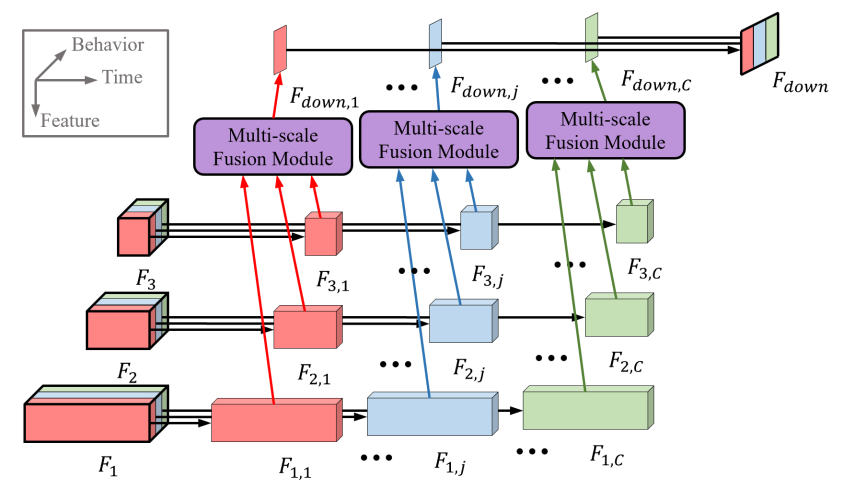


Figure 5 The process of behavior-independent multi-scale feature fusion. Following the bottom-up order of , multi-scale fusion is performed independently according to different behaviors.

The computation process of top-down multi-scale feature fusion is the same as the bottom-up fusion, except that the feature input order is reversed. Similarly, we can obtain the top-down multi-scale fusion result through the same computation process. The multi-scale fusion result is the sum of the outputs from the bottom-up and top-down connections:

where is the final dynamic feature obtained after bidirectional multi-scale feature fusion.

3.4 Multi-modal Fusion

Static and dynamic features exhibit complex nonlinear interactions. For example, a combination of membership level and a recent increase in complaints may serve as a churn warning signal. Classical vector concatenation methods fail to model the multi-modal interactions, whereas the attention mechanism can effectively capture the deep relationships between static and dynamic features. Therefore, we design a multi-modal fusion module based on the attention mechanism, as shown in Figure 1 (IV). It dynamically adjusts the weights of different features based on the current behavior patterns, reducing the interference of irrelevant data.

where, and represent the dynamic features and static features, respectively. is the fused multi-modal feature. represents the average calculation along the first dimension of the feature matrix. represents the scaled dot-product attention. is used as the query vector to control the proportion of and in the fused feature.

The linear classifier maps to the output:

where, and  are the weights and bias. and respectively represent predicted probability of customer not churning and churning.

4.Experiments

4.1 Experiment Setup

**Dataset.** We utilize two public datasets, KKBox (<https://www.kaggle.com/competitions/kkbox-churn-prediction-challenge>) , KDD (<https://www.biendata.xyz/competition/kddcup2015/>), and a private dataset, HOF. Each customer’s information is organized as a sample. We randomly divide them into training, validation, and test sets with a ratio of .

(1) KKBox: The music streaming service platform KKBox provides customer data consisting of seven static and eight dynamic data. The total number of samples is 970,960, of which 87,330 (8.99%) are marked as churn.

(2) KDD: The online course platform XuetangX provides customer data consisting of three static and seven dynamic data. The total number of samples is 120,542, of which 95,581 (79.29%) are marked as churn.

(3) HOF: A mobile game platform provides customer data consisting of three static and eight dynamic data. There are 185,261 samples, of which 102,775 (55.48%) are churn samples.

**Baselines**. To evaluate the performance of the proposed TBFormer, we conduct comparative experiments with eight representative advanced models.

(1) DeepFM [32] has excellent feature extraction capability for categorical data but can only handle static data.

(2) XGBoost [33] (eXtreme Gradient Boosting) is an efficient gradient boosting tree model and is a representative model for classical customer churn prediction.

(3)LightGBM[35] (Light Gradient Boosting Machine) is an efficient gradient boosting tree model that accelerates training speed and is suitable for high-dimensional data.

(4)CatBoost [36] (Categorical Boosting) is a gradient boosting tree model designed for handling categorical features, making it suitable for data with rich categorical attributes and exhibiting strong generalization ability.

(5) MfGRU [29] (Modality-fusion GRU) is a representative model based on the RNN framework. When only dynamic data is input, it is just a vanilla GRU.

(6) MBST [20] (Multivariate Behavior Sequence Transformer) is a representative model based on the Transformer framework. It extracts features along the time and behavior dimensions using two Transformers.

(7)FCLCNN-LSTM [37] (Fully Connected Layer Convolutional Neural Network - Long Short-Term Memory) is a model based on the LSTM framework. Static data is transformed into the same embedding vectors as dynamic data through one-hot encoding before being fed into the model together.

(8)CCP-Net [38] (Customer Churn Prediction Networks) integrates BiLSTM and CNN to enhance feature extraction capabilities. It processes static data using one-hot encoding and then inputs it into the model along with dynamic data.

**Evaluation Metrics**. Customer churn prediction is a typical binary classification problem. Therefore, we use Area Under Curve (AUC) and accuracy (ACC) as the evaluation metrics to measure the prediction performance of the model. In addition, we use Floating Point Operations (FLOPs) to measure the computational complexity of the model. ACC measures the percentage of correctly predicted results out of the total samples. AUC comprehensively evaluates the model's performance across all possible classification thresholds. The closer the values approach to 1, the better the model's performance. Additionally, we introduce Top Decile Lift (TDL) to measure the proportion of positive samples within the top 10%. TDL evaluates the model's effectiveness in identifying high-risk churn customers.

**Implementation Details**. All models are trained and tested on an Intel Core I9-12009K CPU with GeForce RTX 3090 Ti GPUs. The learning rate for all deep models is set to . The patch length is set to 3, and the time-behavior module has 3 layers. The dimension is set to 32. The sequence length for KKBox dataset is 31, while for KDD and HOF datasets, it is 30. The training and testing process is repeated five times, and the average values with the confidence interval are calculated as the final evaluation results to ensure reliability. Each training epoch employs K-fold cross-validation with K=5. Hyperparameters are determined through grid search in Section 4.4.

4.2 Performance Comparison

Table 1 shows the experimental results of the proposed model and the comparison models on three datasets. Based on the input data, the models can be divided into three types: static models, dynamic models, and multi-modal models.

Table 1. Performance comparison with SOTA methods(%). TBFormer achieves the best performance on all datasets. “w/o-dynamic” denotes only static data is input, while “w/o-static” denotes only dynamic data is input. “static+dynamic” denotes that both static and dynamic data are input simultaneously.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | KKBox | | | KDD | | | HOF | | |
| AUC | ACC | TDL | AUC | ACC | TDL | AUC | ACC | TDL |
| **w/o-dynamic** | | | | | | | | | |
| DeepFM | 90.93±0.55 | 80.84±0.27 | 6.254±0.213 | 64.55±0.10 | 60.47±0.10 | 1.159±0.004 | 52.37±0.18 | 51.80±0.18 | 1.083±0.030 |
| XGBoost | 90.73±0.06 | 82.18±0.35 | 6.893±0.019 | 60.01±0.45 | 58.39±0.14 | 1.080±0.011 | 53.60±0.08 | 52.32±0.14 | 1.182±0.021 |
| LightGBM | 91.05±0.10 | 82.21±0.21 | 6.877±0.032 | 62.34±0.29 | 58.85±0.20 | 1.113±0.005 | 53.76±0.13 | 52.20±0.19 | 1.176±0.017 |
| CatBoost | 91.59±0.04 | 82.38±0.44 | 6.887±0.027 | 62.96±0.30 | 59.39±0.12 | 1.093±0.008 | 53.88±0.10 | 52.54±0.11 | 1.186±0.020 |
| MfGRU | 92.45±0.39 | 81.90±1.02 | 6.840±0.093 | 65.00±0.08 | 60.44±0.53 | 1.159±0.005 | 53.22±0.04 | 51.96±0.37 | 1.195±0.002 |
| **w/o-static** | | | | | | | | | |
| XGBoost | 57.68±0.18 | 55.68±0.25 | 1.439±0.095 | 83.54±0.16 | 76.68±0.14 | 1.201±0.003 | 86.39±0.18 | 79.09±0.04 | 1.736±0.007 |
| LightGBM | 58.33±0.24 | 56.72±0.35 | 1.595±0.063 | 83.89±0.25 | 76.86±0.08 | 1.209±0.002 | 86.85±0.10 | 79.34±0.07 | 1.715±0.005 |
| CatBoost | 59.65±0.27 | 58.12±0.29 | 1.643±0.088 | 84.58±0.21 | 77.31±0.06 | 1.212±0.005 | 87.11±0.15 | 79.62±0.12 | 1.738±0.004 |
| MfGRU | 63.56±0.31 | 60.13±0.42 | 2.840±0.061 | 86.64±0.25 | 79.46±0.20 | 1.223±0.006 | 61.90±0.29 | 58.64±0.16 | 1.523±0.010 |
| MBST | 64.88±0.61 | 60.71±0.49 | 2.935±0.081 | 86.16±0.18 | 79.37±0.20 | 1.218±0.002 | 87.40±0.39 | 80.12±1.04 | 1.712±0.028 |
| TBFormer | 67.14±0.24 | 0.6092±0.39 | 3.209±0.060 | 87.02±0.02 | 80.05±0.14 | 1.228±0.001 | 92.81±0.14 | 85.87±0.20 | 1.750±0.009 |
| **static+dynamic** | | | | | | | | | |
| XGBoost | 91.19±0.02 | 82.40±0.10 | 6.900±0.027 | 83.76±0.08 | 76.83±0.16 | 1.207±0.006 | 86.51±0.08 | 79.11±0.08 | 1.737±0.007 |
| LightGBM | 91.46±0.04 | 82.56±0.07 | 6.803±0.018 | 84.07±0.11 | 77.04±0.21 | 1.211±0.003 | 86.91±0.13 | 79.51±0.10 | 1.727±0.003 |
| CatBoost | 91.91±0.03 | 82.67±0.16 | 6.815±0.020 | 84.82±0.06 | 77.73±0.27 | 1.217±0.004 | 87.23±0.06 | 79.85±0.05 | 1.742±0.006 |
| MfGRU | 91.79±0.14 | 81.31±0.27 | 6.686±0.065 | 87.09±0.04 | 79.74±0.10 | 1.228±0.002 | 63.11±0.33 | 59.19±0.08 | 1.583±0.004 |
| FCLCNN-LSTM | 91.87±0.18 | 81.83±0.21 | 6.717±0.043 | 86.59±0.09 | 78.74±0.12 | 1.220±0.003 | 74.67±0.07 | 69.93±0.13 | 1.632±0.005 |
| CCP-Net | 92.03±0.10 | 82.16±0.12 | 6.789±0.051 | 87.10±0.05 | 79.89±0.11 | 1.226±0.002 | 78.13±0.12 | 72.32±0.15 | 1.645±0.002 |
| MBST | 91.19±0.04 | 82.78±0.06 | 6.893±0.033 | 86.39±0.16 | 79.50±0.24 | 1.221±0.005 | 88.62±0.90 | 80.66±1.12 | 1.733±0.016 |
| **TBFormer** | **93.04±0.10** | **83.46±0.16** | **6.954±0.033** | **87.26±0.02** | **80.18±0.14** | **1.232±0.004** | **93.30±0.25** | **86.45±0.35** | **1.757±0.007** |

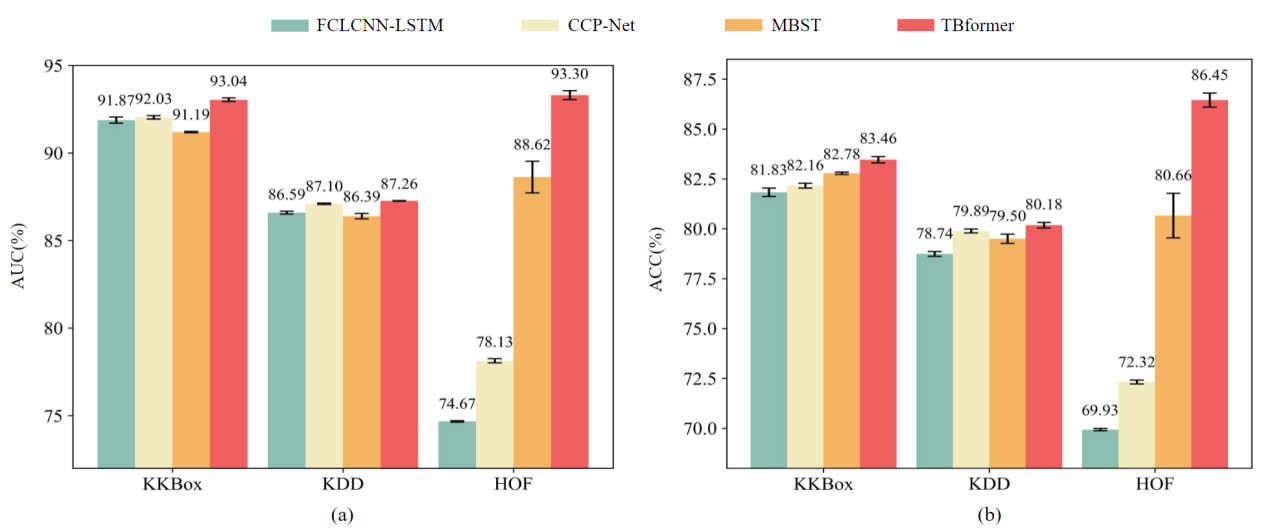


Figure 6. Comparison with the deep learning-based multi-modal baselines. TBformer has a significant performance improvement of (a)AUC and (b)ACC and a narrow confidence interval.

Our proposed TBFormer achieves an average AUC of 91.2% and ACC of 83.36%. Compared to the best baseline MBST, TBFormer improves the average AUC and ACC by 2.47% and 2.38% in Figure 6, across three datasets. The confidence interval of TBFormer is narrow and does not overlap with that of MBST, demonstrating the stability of the results and a significant performance improvement. Furthermore, TBFormer achieves the best TDL, validating that our proposed multi-modal fusion method effectively focuses on key features, thereby enhancing prediction accuracy for high-risk customers. In contrast, MBST based on the Transformer framework only utilizes vector concatenation for feature fusion, thereby introducing redundancy. FCLCNN-LSTM and CCP-Net process static data using one-hot encoding before feeding the multi-modal data into a hybrid deep learning model. They lead to feature confusion, thereby limiting the model's performance. MfGRU performs dynamic weighted data fusion and extracts dynamic features by GRU. Although GRU has good capability in modeling contextual information, it is inferior to Transformer in capturing global information. These methods introduce noise interference through data fusion, increasing computational and resource consumption. Besides, classical methods perform the worst among the models because classical methods heavily rely on feature engineering and lack the ability to extract various high-dimensional features from time series data adaptively. Utilizing statistical features such as the mean and variance of the variables is ineffective in representing all the information in the time series inputs. Among the three, XGBoost performs the worst due to the manual encoding of categorical data. LightGBM accelerates the training process but is prone to overfitting. CatBoost automatically handles categorical data and has stronger generalization capabilities compared to the other two. The superior performance of TBFormer over the current state-of-the-art models based on Transformer and RNN frameworks demonstrates the effectiveness and superiority of the proposed method in customer churn prediction tasks.

Furthermore, with the single-modal input on the KKBox dataset, the average AUC and ACC of XGBoostw/o-dynamic and MfGRUw/o-dynamic are 30.05% and 28.89% higher than those of XGBoostw/o-static and MfGRUw/o-static. Since static data predominates in the KKBox dataset, using only dynamic data leads to poor predictive performance. However, the results are the opposite for the KDD and HOF datasets. It indicates that the contribution of different modalities to customer churn prediction varies on different datasets. Across all three datasets, TBFormer achieves an average AUC of 8.88% and 21.92% higher than TBFormerw/o-static and DeepFMw/o-dynamic, respectively. The superior performance of TBFormer over any single-modal model demonstrates the necessity and effectiveness of fusing static and dynamic features.

4.3 Ablation studies

**Effectiveness of time-behavior module.** As shown in Table 2, to evaluate the feature extraction capability of time-behavior attention on dynamic data, we compare TBFormerw/o-static with seven advanced temporal models [18][11][12][13][14][15][16].

All of these methods are based on the Transformer framework, but the differences are that they utilize different attention modules to model the relationships between dynamic data. The results show that the AUC of vanilla Transformer outperforms other baseline variants except TBformer. It demonstrates that the vanilla Transformer framework is sufficient to capture the temporal correlations necessary for customer churn prediction. The improved attention modules of the other variants are not appropriate to extract logical relationships between different behaviors and disrupt long-term trends, resulting in worse performance. However, our proposed multi-scale time-behavior module is specifically designed to handle characteristics of dynamic data. It can simultaneously extract dynamic features along time and behavior dimensions, achieving better performance. TBformer achieves the best performance because it can additionally capture behavioral relationships beyond what vanilla Transformer can accomplish. These results demonstrate that the time-behavior module is effective for customer churn prediction task.

**Effectiveness of time-behavior attention mechanism.** As shown in Table 5, we compare TBFormerw/o-static with other attention methods to evaluate the effectiveness of time-behavior attention. “Global” denotes replacing time-behavior attention with global self-attention. “Time-behavior&N” denotes replacing the dimensional independent operation. Global selfattention mechanism pays equal attention to all patches across all dimensions, limiting the ability of the model to capture significant information. However, the time-behavior attention mechanism makes the model more focused on patches that are useful for predicting customer churn and directly ignore other irrelevant patches. Besides, independent operation has different focuses in the temporal and behavioral dimensions, respectively, which better matches the semantics of the real-world application scenarios. But nonindependent operation obscures the model’s attention across all dimensions. Therefore, TBformer’s performance best demonstrates that effectiveness of our proposed timebehavior attention mechanism.

Furthermore, time-beahavior attention can significantly reduce FLOPs of the model by time-beahavior attention. Each patch in global self-attention pays attention to patches, while each patch in time-beahavior attention pays attention to only patches. The time complexity of attention module has been reduced from to . operation does not affect FLOPs of the model.

Table 2. Comparison with SOTA temporal models.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | KKBox | | | KDD | | | HOF | | |
| AUC | ACC | TDL | AUC | ACC | TDL | AUC | ACC | TDL |
| Transformer | 66.78±0.20 | **61.92±0.18** | 3.163±0.047 | 86.83±0.06 | 79.78±0.12 | 1.226±0.005 | 90.36±0.08 | 82.30±0.06 | **1.754±0.005** |
| Autoformer | 63.72±0.02 | 58.58±0.02 | 2.945±0.062 | 86.55±0.10 | 79.47±0.24 | 1.229±0.002 | 79.84±0.90 | 72.19±0.55 | 1.685±0.022 |
| FEDformer | 65.86±0.49 | 61.11±0.29 | 3.140±0.035 | 86.76±0.06 | 79.56±0.12 | 1.222±0.003 | 89.95±0.08 | 81.62±0.14 | 1.747±0.006 |
| Pyraformer | 66.53±0.37 | 61.33±0.27 | 3.123±0.051 | 86.73±0.08 | 79.65±0.16 | 1.225±0.003 | 89.99±0.06 | 81.90±0.06 | 1.741±0.004 |
| DLinear | 65.76±0.25 | 60.68±0.22 | 3.016±0.053 | 86.54±0.04 | 78.48±0.31 | 1.225±0.002 | 88.94±0.04 | 80.76±0.14 | 1.719±0.004 |
| PatchTST | 65.74±0.41 | 60.75±0.25 | 3.038±0.054 | 86.57±0.24 | 79.19±0.47 | 1.222±0.006 | 90.10±0.10 | 82.17±0.08 | 1.741±0.004 |
| iTransformer | 66.01±0.31 | 61.45±0.37 | 3.080±0.047 | 86.52±0.08 | 79.05±0.14 | 1.221±0.002 | 90.22±0.41 | 82.26±0.55 | 1.746±0.007 |
| **TBFormerw/o-static** | **67.14±0.24** | 61.77±0.39 | **3.209±0.060** | **87.02±0.02** | **80.05±0.14** | **1.230±0.001** | **92.81±0.14** | **85.87±0.20** | 1.750±0.009 |

Table 5. Comparison with different attention methods. “N” denotes Non-independent operation ,respectively.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attention Method | KKBox | | KDD | | HOF | |
| AUC | FLOPs | AUC | FLOPs | AUC | FLOPs |
| Global | 66.78±0.45 | 3.02M | 86.83±0.14 | 2.26M | 92.45±0.08 | 2.66M |
| Time-behavior&N | 66.71±0.22 | **2.51M** | 86.74±0.25 | **1.96M** | 92.28±0.10 | **2.25M** |
| TBFormerw/o-static | **67.14±0.24** | **2.51M** | **87.02±0.02** | **1.96M** | **92.81±0.14** | **2.25M** |

**Effectiveness of time** **downsampling:** The time downsampling module can downsample the features along the time dimension to obtain dynamic multi-scale features. It can capture short-term fluctuations and long-term trends within behavioral sequences to understand customers' interests and habits. Therefore, multi-scale features contain global details and local structures that enhance contextual information and improve model robustness, leading to better performance. However, the extracted multi-layer features of “**TBFormerw/o-static&downsampling”** are derived from the same time scale，making it difficult to capture behavior changes across different periods. In addition, time downsampling can also reduce FLOPs of the model. The results in Table 6 demonstrate the necessity of time downsampling for dynamic feature extraction.

Table 6. Effectiveness of time downsampling.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | KKBox | | KDD | | HOF | |
| AUC | FLOPs | AUC | FLOPs | AUC | FLOPs |
| TBFormerw/o-static&downsampling | 66.91±0.02 | 3.91M | 86.92±0.08 | 3.08M | 92.52±0.14 | 3.54M |
| TBFormerw/o-static | **67.14±0.24** | **2.51M** | **87.02±0.02** | **1.96M** | **92.81±0.14** | **2.25M** |

**Effectiveness of multi-scale fusion**: As shown in Table 3, we compare various combinations of multi-scale fusion. “Concatenation” denotes replacing with vector concatenation operation. The results show that the fusion module which treats behaviors as independent “I” performs better than the one “N” considering the dependencies between behaviors. Because time-behavior attention has already captured cross-influence between different behaviors, further cross-behavior fusion in a multi-scale fusion module would increase the model complexity and make it harder to converge. So behavior-independent multi-scale feature fusion achieves better performance. Moreover, the bidirectional fusion path combines bottom-up and top-down connections, achieving better results than a single fusion path. Bidirectional connections simultaneously utilize both low-level details and high-level semantic information of customer behaviors, enhancing the model's robustness. TBformer (I&B) improves the AUC by 0.33%, 0.2%, and 1.02% on three datasets compared to simple vector concatenation. The results demonstrate the necessity of behavior-independent multi-scale fusion for dynamic data.

Table 3 Effectiveness of multi-scale fusion. The evaluation matric is AUC(%).

|  |  |  |  |
| --- | --- | --- | --- |
| Fusion Method | KKBox | KDD | HOF |
| Concatenation | 92.71±0.12 | 87.06±0.08 | 92.28±0.78 |
| N&U | 92.84±0.06 | 86.54±0.12 | 90.01±0.10 |
| N&D | 92.78±0.08 | 86.29±0.24 | 89.51±0.33 |
| N&B | 92.91±0.10 | 86.45±0.33 | 89.35±0.06 |
| I&U | 93.01±0.08 | 87.16±0.06 | **93.30±0.22** |
| I&D | 92.99±0.14 | 87.15±0.04 | 91.71±0.74 |
| TBFormer(I&B) | **93.04±0.10** | **87.26±0.02** | **93.30±0.25** |

“N” and “I” denote Non-independent and Independent behaviors,respectively. “U”, “D”, and “B” represent Upward, Downward, and Bidirectional connections.

**Effectiveness of multi-modal fusion**: As shown in Table 4, we compare various multi-modal fusion methods. "Fully Connect" concatenates the vectors and adds an extra fully connected layer, with the output shape matching the multi-modal features. “Cartesian Product” denotes replacing with Cartesian product operation. Simple vector concatenation does not consider the correlation between features of different modalities. Cartesian-product would significantly increase the feature dimensions. Our designed multi-modal fusion module achieves the best performance due to the attention mechanism. It can adaptively control the weights of each feature in the fused result and make the modal attention score of the more significant contribution larger to avoid noise and redundancy. The results demonstrate the superiority of TBFormer in multi-modal fusion.

Table4 Effectiveness of multi-modal fusion. The evaluation matric is AUC(%).

|  |  |  |  |
| --- | --- | --- | --- |
| Fusion Method | KKBox | KDD | HOF |
| Concatenation | 92.48±0.31 | 87.13±0.04 | 93.18±0.22 |
| Fully Connect | 92.80±0.14 | 87.19±0.02 | 93.27±0.73 |
| Cartesian Product | 92.25±0.24 | 86.86±0.10 | 91.26±0.45 |
| TBFormer | **93.04±0.10** | **87.26±0.02** | **93.30±0.25** |

4.4 Sensitivity Analysis

**Effect of patch length**: As shown in Figure 4(a), we conduct experiments on different patch lengths within the behavior-independent embedding module. The results indicate that as the patch length increases from 1, the model’s performance initially improves but then gradually deteriorates. A smaller lacks sufficient information to learn behavioral patterns, whereas a larger makes it difficult to capture short-term variations. On all datasets, TBformer achieves the best performance with the patch length . In short, is the appropriate patch length for the patch embedding module.

**Effect of the layers of the time-behavior module:** Figure 4(b) shows the results for different . A larger  indicates that TBformer samples across more scales. When  is large, adjacent scales provide similar information, resulting in highly correlated features that can lead to overfitting. On the other hand, when  is small, short-term anomalous fluctuations may be overlooked, reducing the model's generalization ability. When , TBformer achieves the highest AUC.

**Effect of the dimension :** In Figure 4(c), we adjust the dimension of TBformer. A smaller limits TBformer's ability to extract information, leading to the loss of important patterns and underfitting. Conversely, when is larger, the features may include redundant or irrelevant information, reducing the model's ability to focus on important features. TBformer performs best when the dimension is set to 32.

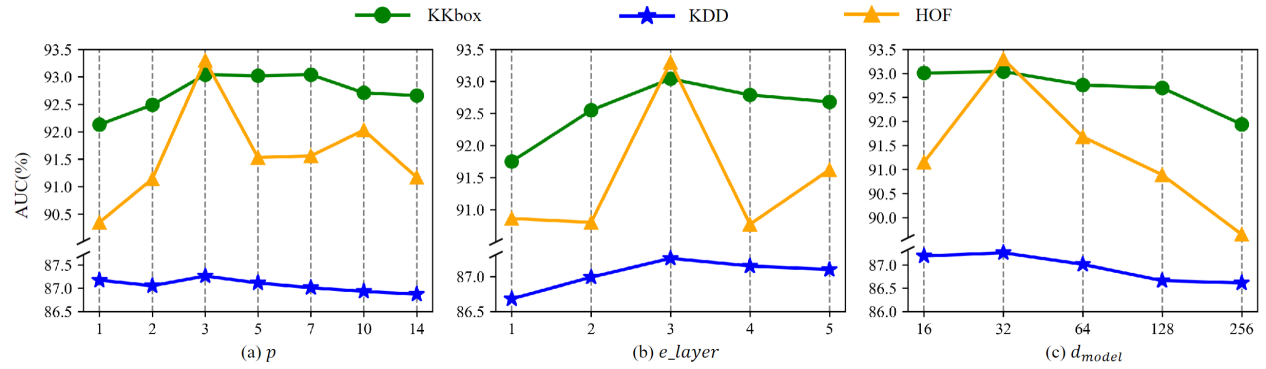


Figure 4. Sensitivity to different hyperparameters on the three datasets. The evaluation metrics are AUC(%). (a) Effect of patch length . (b) Effect of the layers *e\_layer* of the time-behavior module. (c) Effect of the dimension .

5.Conclusion

In this work, we propose a novel multi-modal Transformer with Time-Behavior attention called TBformer for customer churn prediction. TBformer utilizes the time-behavior module to extract dynamic multi-scale features, which can simultaneously capture temporal and behavioral relationships. The behavior-independent bidirectional multi-scale fusion module obtains more comprehensive and rich contextual information. TBformer utilizes DeepFM to extract static features. We also design a multi-modal fusion module based on the attention mechanism to fuse dynamic and static features for better predictive performance efficiently. TBformer breaks through the bottleneck of existing methods using single-modal data. The experimental results on three real-world datasets demonstrate that our proposed TBformer outperforms the current state-of-the-art methods for predicting customer churn. TBformer makes substantial contributions to the consumer technologies domain by improving the accuracy of customer churn prediction in Internet service platforms, providing a robust and practical approach to analyzing customer behavior patterns. TBformer can assist real-world service platforms in customizing customer retention strategies, adjusting subsequent marketing strategies, and having broad application prospects in various fields such as e-commerce, social networks, and online gaming.

However, in real-world scenarios, customer behavior periods and trends are often unstable, and fixed-scale features extracted through downsampling are insufficient to represent dynamic, time-varying data. Future research will explore adaptive time scales to optimize the capture of time-varying behavioral patterns.

Reference

1. C. Fornell, B. Wernerfelt. Defensive marketing strategy by customer complaint management: a theoretical analysis. Journal of Marketing research, 1987, 24(4): 337-346
2. B. Huang, M. T. Kechadi, B. Buckley. Customer churn prediction in telecommunications. Expert Systems with Applications, 2012, 39(1): 1414-1425
3. R. A. Soeini, K. V. Rodpysh. Applying data mining to insurance customer churn management. International Proceedings of Computer Science and Information Technology, 2012, 30(2012): 82-92
4. Kim, Sulim and Heeseok Lee. “Customer Churn Prediction in Influencer Commerce: An Application of Decision Trees.” International Conference on Information Technology and Quantitative Management(2021).
5. Coussement, Kristof and Koen W. De Bock. “Customer churn prediction in the online gambling industry: The beneficial effect of ensemble learning.” Journal of Business Research 66 (2013): 1629-1636.
6. Raj, A. Antony and D. Vetrithangam. “Machine Learning and Deep Learning technique used in Customer Churn Prediction: - A Review.” 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES) (2023): 139-144.
7. E. Stripling, S. vanden Broucke, K. Antonio, B. Baesens, M. Snoeck. Profit maximizing logistic regression modeling for customer churn prediction, in: 2015 IEEE International Conference on Data Science and Advanced Analytics. Paris, France, 19-21 Oct. 2015, IEEE, 2015: 1-10
8. Ahmad A K, Jafar A, Aljoumaa K. Customer churn prediction in telecom using machine learning in big data platform[J]. Journal of Big Data, 2019, 6(1): 1-24.
9. P. Lalwani, M. K. Mishra, J. S. Chadha, P. Sethi. Customer churn prediction system: a machine learning approach. Computing, 2022: 1-24
10. J. Ran, X. Cheng. Airline Customer Value Analysis and Customer Churn Prediction Based on LRFMC Model and K-means Algorithm, in: 2021 2nd International Conference on Computer Science and Management Technology. Shanghai, China, 12-14 Nov. 2021, IEEE, 2021: 185-193
11. H. Wu, J. Xu, J. Wang, M. Long. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting, in: 35th Conference on Neural Information Processing Systems. Virtual, Online, 6-14, Dec. 2021, Neural information processing systems foundation, 2021: 22419-22430
12. T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, R. Jin. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting, in: 39th International Conference on Machine Learning. Baltimore, MD, United states, 17-23, Jul. 2022, ML Research Press, 2022: 27268-27286
13. S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, et al. PYRAFORMER: LOW-COMPLEXITY PYRAMIDAL ATTENTION FOR LONG-RANGE TIME SERIES MODELING AND FORECASTING, in: 10th International Conference on Learning Representations. Virtual, Online, 25-29 Apr. 2022, ICLR 2022, 2022: ByteDance; et al.; Meta AI; Microsoft; Qualcomm; Sea Al Lab
14. A. Zeng, M. Chen, L. Zhang, Q. Xu. Are Transformers Effective for Time Series Forecasting?, in: 37th AAAI Conference on Artificial Intelligence. Washington, DC, United states, 7-14, Feb. 2023, AAAI Press, 2023: 11121-11128
15. Y. Nie, N. H. Nguyen, P. Sinthong, J. Kalagnanam. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers, in: 11th International Conference on Learning Representations. Kigali, Rwanda, 1-5 May. 2023, ICLR 2023, 2023
16. Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, et al. ITRANSFORMER: INVERTED TRANSFORMERS ARE EFFECTIVE FOR TIME SERIES FORECASTING. arXiv preprint, 2023: 2310.06625
17. Wang, Cheng et al. “Risk assessment of customer churn in telco using FCLCNN-LSTM model.” Expert Systems with Applications (2024): n. pag.
18. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al. Attention is all you need, in: 31st Annual Conference on Neural Information Processing Systems. Long Beach, CA, United states, 4 Dec. 2017, Neural information processing systems foundation, 2017: 5999-6009
19. Alboukaey N, Joukhadar A, Ghneim N. Dynamic behavior based churn prediction in mobile telecom[J]. Expert Systems with Applications, 2020, 162: 113779.
20. X. Wu, P. Li, M. Zhao, Y. Liu, R. G. Crespo, E. Herrera Viedma. Customer churn prediction for web browsers. Expert Systems with Applications, 2022, 209: 118177
21. Altairey, Habes Ali and Adel Ismail Al-Alawi. “Customer Churn Prediction in Telecommunication and Banking using Machine Learning: A Systematic Literature Review.” 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS) (2024): 483-490.
22. S. M. Keaveney. Customer switching behavior in service industries: An exploratory study. Journal of marketing, 1995, 59(2): 71-82
23. Y. Xie, X. Li, E. Ngai, W. Ying. Customer churn prediction using improved balanced random forests. Expert Systems with Applications, 2009, 36(3): 5445-5449
24. E. Stripling, S. vanden Broucke, K. Antonio, B. Baesens, M. Snoeck. Profit maximizing logistic regression modeling for customer churn prediction, in: 2015 IEEE International Conference on Data Science and Advanced Analytics. Paris, France, 19-21 Oct. 2015, IEEE, 2015: 1-10
25. S. Höppner, E. Stripling, B. Baesens, S. vanden Broucke, T. Verdonck. Profit driven decision trees for churn prediction. European journal of operational research, 2020, 284(3): 920-933
26. C. Yang, X. Shi, L. Jie, J. Han. I know you'll be back: Interpretable new user clustering and churn prediction on a mobile social application, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018: 914-922
27. A. Zheng, L. Chen, F. Xie, J. Tao, C. Fan, Z. Zheng. Keep you from leaving: Churn prediction in online games, in: Database Systems for Advanced Applications: 25th International Conference, DASFAA 2020, Jeju, South Korea, September 24–27, 2020, Proceedings, Part II 25. Springer, 2020: 263-279
28. S. Yin, L. Lei, H. Wang, W. Chen. Power of attention in MOOC dropout prediction. IEEE Access, 2020, 8: 202993-203002
29. Yiwen X U, Xiaoyang L I, Qiwen D, et al. Merchant churn prediction based on transaction data of aggregate payment platform[J]. Journal of East China Normal University (Natural Science), 2020(5): 167.
30. H. Kwon, H. H. Kim, J. An, J.-H. Lee, Y. R. Park. Lifelog data-based prediction model of digital health care app customer churn: retrospective observational study. Journal of Medical Internet Research, 2021, 23(1): e22184
31. Panimalar S A, Krishnakumar A. Customer churn prediction model in cloud environment using DFE-WUNB: ANN deep feature extraction with weight updated tuned Naïve bayes classification with block-jacobi SVD dimensionality reduction[J]. Engineering Applications of Artificial Intelligence, 2023, 126: 107015.
32. H. Guo, R. Tang, Y. Ye, Z. Li, X. He. DeepFM: A factorization-machine based neural network for CTR prediction, in: 26th International Joint Conference on Artificial Intelligence. Melbourne, VIC, Australia, 19-25, Aug. 2017, International Joint Conferences on Artificial Intelligence, 2017: 1725-1731
33. T. Chen, C. Guestrin. XGBoost: A scalable tree boosting system, in: 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD San Francisco, CA, United states, 13-17, Aug. 2016, Association for Computing Machinery, 2016: 785-794
34. Consumer-Driven Decision-Making Service Model Selection Using Contextualization and Personalization for Consumer Electronics in E-Commerce
35. Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree[J]. Advances in neural information processing systems, 2017, 30.
36. Prokhorenkova L, Gusev G, Vorobev A, et al. CatBoost: unbiased boosting with categorical features[J]. Advances in neural information processing systems, 2018, 31.
37. Wang C, Rao C, Hu F, et al. Risk assessment of customer churn in telco using fclcnn-lstm model[J]. Expert Systems with Applications, 2024, 248: 123352.
38. Liu X, Xia G, Zhang X, et al. Customer churn prediction model based on hybrid neural networks[J]. Scientific Reports, 2024, 14(1): 30707.