A Comparative Study of XGBoost and Transformer for E-commerce Purchase Prediction under Class Imbalance

Yaxin Wang
Oxford Brookes College, Chengdu University of Technology, Chengdu, Sichuan, 610059, China
wangyaxin@student.zy.cdut.edu.cn

ABSTRACT

Class imbalance poses a persistent challenge in e-commerce purchase prediction, where positive samples rarely exceed 10% of observations. This study compares XGBoost and Transformer architectures under controlled imbalance conditions using the Tmall Repeat Purchase Prediction dataset (6.12% positive rate). A 2 × 2 factorial design crossed model type (XGBoost vs. Transformer) with balancing strategy (original imbalanced vs. SMOTE-balanced), drawing on 22 tabular features and 20-length behavioral sequences derived from user interaction logs. On the original data, XGBoost outperformed Transformer (AUC 0.657 vs. 0.631; F1 0.194 vs. 0.169). SMOTE, counter to expectations, degraded both models. Ablation experiments confirmed this was not an artifact of the sampling ratio. At extreme imbalance (1% positive rate), the pattern reversed: Transformer proved more robust, its AUC falling by only 1.7% against XGBoost's 9.1%, and surpassing XGBoost on AUC at this level. These results challenge the common assumption that tree-based models are always preferable under class imbalance and offer practical guidance for model selection in real-world e-commerce settings.

Keywords: E-Commerce Purchase Prediction, Class Imbalance, Xgboost, Transformer, SMOTE, Behavioral Sequence Modeling

1. INTRODUCTION

E-commerce platforms operate under intensifying competition, making accurate purchase prediction a decisive lever for marketing return on investment (ROI). Yet purchasing behavior is, by nature, sparse: positive samples rarely exceed single-digit percentages in real-world datasets, and class imbalance remains a central challenge in this domain [1]. Two modeling approaches have received sustained attention. Gradient-boosted decision trees, led by XGBoost [2], are widely used for tabular classification. Transformer architectures [3], meanwhile, have been adapted from natural language processing to capture sequential patterns in user behavior for recommendation. Recent benchmarks show that tree-based models consistently outperform deep learning on tabular data [4].

Although XGBoost and Transformer each have an established track record, two gaps stand out in the current literature. First, Grinsztajn et al. [4] compared tree-based models against deep learning across 45 benchmark datasets and found that tree models generally lead on tabular data. None of those benchmarks, however, involved severe class imbalance (positive rate below 10%), leaving the comparison untested under the very condition that defines real-world purchase prediction. Second, while SMOTE [5] remains a widely adopted remedy for class imbalance, Blagus and Lusa [6] showed that its effectiveness erodes in high-dimensional spaces, where synthetic samples may introduce noise instead of informative variation. Whether this limitation extends to hybrid feature spaces (combining tabular features with behavioral sequences, as is typical in e-commerce prediction) has not been examined. The practical cost of these two gaps is straightforward: the literature offers no clear answer to which architecture performs better at different levels of imbalance severity.

In order to eliminate this gap, this study will raise three questions. RQ1: Which one can give higher AUC and F1, XGBoost or Transformer, for naturally biased data? Rq2: Which model will get better when using smote's oversampling? Which is more affected? RQ3: With the deepening of deviation ratio, which model has less performance degradation?

As far as we know, this is the first time to compare XGBoost and Transformer systematically in a unified experimental framework for e-commerce purchase prediction. Four new achievements have emerged. First, the results show that the over-sampling of SMOTE will reduce the performance of the two models, and ablation experiments confirm that the selection of sampling ratio has not become the main reason for the mixed results. Second, Transformer has demonstrated its outstanding strength in the case of extreme deviation. When the positive rate is 1%, the decrease of AUC is only 1.7% and XGBoost is 9.1%. This result challenges the general idea that the tree-based model is better than usual in the case of

deviation. Thirdly, the results obtained in the experiment are summarized as practical model selection rules with different degrees of deviation. Fourthly, the scope of SMOTE in mixed feature space for additional experiments is defined, and the previous theoretical research [6] is extended to the field of e-commerce prediction.

The rest of this paper is arranged as follows. Section 2 reviews the related research in three fields. How does the domain deal with XGBoost of e-commerce, sequential recommendation using Transformer and class imbalance? Section 3 introduces the process and experimental plan of dataset and feature engineering. Section 4 gives the main results and additional experiments. In Section 5, the significance of this discovery is discussed, and in Section 6, the practical proposals and future directions are summarized.

2. LITERATURE REVIEW

2.1 The Application of XGBoost in E-commerce Forecasting

XGBoost [2] is the task of tabular classification and a good standard. Especially in the Tmall dataset, Jing and Shi [7] used XGBoost to predict repeat purchases. Their research analyzed feature engineering and parameter optimization. Class imbalance is mapped by scale_pos_weight parameter, but it is not the core issue of this study.Tanvir et al. [8] reported effective purchase intention prediction with XGBoost, treating SMOTE as a preprocessing step rather than an object of systematic comparison. Gayathri A. Unni et al. [9] integrated SMOTE and collaborative filtering into an XGBoost pipeline for sales failure prediction, though their focus fell on the fusion architecture rather than on how imbalance interacts with model choice. Across these studies, XGBoost consistently performs well, but its behavior under the extreme imbalance ratios typical of real-world purchase data has not been isolated and tested directly.

2.2 Applications of Transformers in Sequence Recommendation

Vaswani et al. [3] introduced the Transformer, whose self-attention mechanism captures long-range dependencies in sequential data. Kang and McAuley [10] adapted this mechanism to sequential recommendation with SASRec, which weights historical items to predict the next interaction. Closest to the present study, Chen et al. [11] proposed the Behavior Sequence Transformer (BST) for recommendation on Alibaba's e-commerce platform. BST encodes user behavior sequences through a Transformer layer and fuses the resulting representations with user and item features, an architecture that directly informed the Transformer component in this study. Prior evaluations of Transformer-based recommenders have centered on ranking metrics; their behavior under severe class imbalance remains largely unexamined.

2.3 Methods for Handling Class Imbalance

Class imbalance has received extensive treatment in the literature [12]. At the data level, Chawla et al. [5] introduced SMOTE, which generates synthetic minority samples via interpolation in the feature space and remains among the most widely adopted oversampling techniques. Blagus and Lusa [6], however, showed that SMOTE's gains erode in high-dimensional settings, where synthetic samples may introduce noise instead of informative signal. At the algorithm level, Lin et al. [13] proposed Focal Loss, which modulates the loss contribution of individual samples to down-weight easy negatives. Whether SMOTE remains effective in a hybrid feature space, where tabular features coexist with behavioral sequences, has received little direct investigation.

3. METHOD

3.1 Dataset

To address the gaps identified in Section 2, a controlled experimental comparison was conducted using real-world e-commerce data. The study drew on the Tmall Repeat Purchase Prediction dataset from the Alibaba Cloud Tianchi platform [14], which contains 260,864 labeled user–merchant pairs, 54,925,330 user behavior logs (spanning click, add-to-cart, purchase, and favorite actions), and profile information for 424,170 users. Positive class (label = 1) accounts for 6.12% of the observed data. This reflects the poor class imbalance in actual purchase behavior.

3.2 Data Preprocessing and Feature Engineering

3.2.1 Tabular Feature Construction

Users' behavior logs were collected, and 22 tabular features were formed, which were divided into 4 categories.

(1) user–merchant interaction statistics (9 dimensions): total_interactions, purchase_count, click_count and derived ratios including purchase_rate and cart_rate.

(2) User-level global behavior statistics (7 dimensions): user_total_actions, user_unique_merchants, and user_ purchase_ratio.

(3) merchant-level global statistics (4 dimensions): merchant_total_actions, merchant_unique_users, and merchant _purchase_ratio.

(4) User demographics (2 dimensions): age_range and gender.

3.2.2 Sequence Feature Construction

For each user and merchant pair, the latest 20 actions are taken out in the order of time_stamp from small to large, and the sequence of behaviors is made. Because the 90 value of the interaction number is 8, the length of the sequence is set to 20. If the length is 20, it can cover more than 99% of all sample. Sequence with length less than 20 adds 1 to the left to adjust the length. 1 is different from the ordinary action type, so you can clearly distinguish the added position from the real interaction. Then redialed the number of action type. PAD (−1) → 0, click (0) → 1, add-to-cart (1) → 2, purchase (2) → 3, favorite (3) → 4, and set the size of vocabulary to 5.

3.2.3 Data Splitting and Standardization

The dataset is stratified sampling based on label, and it is divided into training set and test set according to the ratio of 8 to 2. Tabular features are standardized in StandardScaler. The arranged features are converted into numbers separately, without standardization. There are 208,691 samples in the final training set and 52,173 samples in the test set. Both of them kept the original positive rate of 6.12%.

3.3 Models

3.3.1 XGBoost

XGBoost was trained with 200 trees (max depth = 6, learning rate = 0.05). Class imbalance was addressed through cost-sensitive weighting: the minority class weight was set to approximately 15.4, matching the observed negative-to-positive ratio. During prediction, the decision threshold was scanned over [0.01, 0.99] and the value maximizing F1 was selected, rather than defaulting to 0.5.

3.3.2 Transformer

Behavioral sequences were mapped to 64-dimensional dense vectors through a learned embedding layer (padding index = 0), and a learnable positional encoding was added. The embedded sequences then passed through two Transformer encoder layers (nhead = 4, feedforward dimension = 256, dropout = 0.2), which capture inter-action dependencies via self-attention. After mean pooling over the sequence dimension, the resulting vector was concatenated with the 22 tabular features and fed into a feedforward classifier (128 → ReLU → Dropout → 32 → ReLU → Dropout → 1) with a final sigmoid activation. The model was trained for 30 epochs using the AdamW optimizer (learning rate = 0.001, weight decay = 1e−4), a cosine annealing learning rate schedule, and binary cross-entropy loss (batch size = 256).

3.4 Experimental Design

3.4.1 Main Experiment

A 2 × 2 factorial design crossed model type (XGBoost vs. Transformer) with balancing strategy (original imbalanced vs. SMOTE-balanced), yielding four experimental conditions: XGBoost on original data, XGBoost with SMOTE, Transformer on original data, and Transformer with SMOTE. SMOTE was applied at a 1:1 sampling ratio (k_neighbors = 3). All four conditions shared identical data splits, feature sets, and evaluation metrics.

3.4.2 Supplementary Experiments

Four supplementary experiments were run. (1) To assess robustness under intensifying imbalance (RQ3), both models were evaluated at positive sample ratios of 1%, 3%, 6%, and 10%. (2) XGBoost feature importance was extracted to identify the most influential features. (3) A SMOTE ablation study tested three sampling ratios (1:1, 1:2, 1:3) to exclude ratio selection as a confounding factor. (4) A sequence-removal experiment tested whether the models' behavior depends on sequence information: behavioral sequences in both the training and test sets were replaced entirely with padding tokens,

and both models were retrained and evaluated at the 1% and 3% positive rates, following the same subsampling procedure as experiment (1).

3.5 Evaluation Criteria

Model performance was evaluated primarily using AUC-ROC, which is threshold-invariant and suited to imbalanced classification, and F1-score, which balances precision and recall. Accuracy, precision, recall, and confusion matrices were also reported for diagnostic purposes.

4. EXPERIMENTAL RESULTS

4.1 Main experimental results

Table 1 reports the performance of the four experimental conditions, and Figure 1 visualizes the comparison. On the original imbalanced dataset, XGBoost achieved the highest AUC (0.657) and F1 (0.194), followed by Transformer (AUC 0.631, F1 0.169). Both F1 scores remained below 0.20, reflecting the inherent difficulty of prediction at a 6.12% positive rate.

Table 1. Performance comparison of four experimental conditions.

Experiment	Accuracy	Precision	Recall	F1	AUC
XGBoost (original)	0.825	0.135	0.344	0.194	0.657
XGBoost + SMOTE	0.755	0.095	0.352	0.149	0.598
Transformer (original)	0.850	0.128	0.250	0.169	0.631
Transformer + SMOTE	0.324	0.063	0.727	0.116	0.512

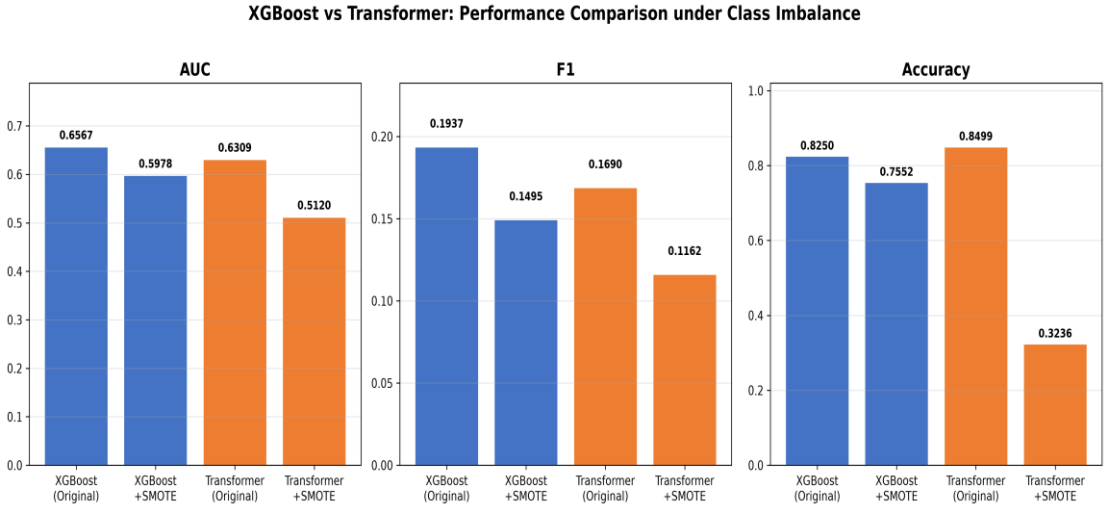

Figure 1. Performance comparison of XGBoost and Transformer under original imbalanced data and SMOTE-balanced data across AUC, F1, and Accuracy.

RQ1 resolves in favor of XGBoost. On the original imbalanced data, XGBoost led Transformer on AUC by 0.026 and on F1 by 0.025. DeLong's test confirmed that the AUC difference was statistically significant ($Z = 7.32$, $p < 0.001$). These results align with Grinsztajn et al. [4], who demonstrated that tree-based models generally outperform deep learning methods on tabular data; notably, the F1 scores in the present study fell well below those reported on the balanced benchmarks in [4], reflecting the severity of the 6.12% positive-rate condition.

RQ2 produces a counterintuitive result. Under SMOTE at a 1:1 ratio, both models deteriorated: XGBoost lost 9.0% in AUC and 23.2% in F1, while Transformer suffered larger losses of 18.9% in AUC and 31.4% in F1. The decline was not

symmetric. Transformer's recall jumped to 0.727, yet its precision collapsed to 0.063, yielding an accuracy of 0.324—the model, in effect, learned to over-predict the minority class. For both architectures, a standard remedy for class imbalance had backfired.

4.2 Performance Degradation at Different Positive Sample Rates

Table 2 and Figure 2 present the performance of both models as the positive sample ratio was progressively reduced from 10% to 1%. Both models exhibited performance degradation as imbalance intensified, but the magnitude of decline differed markedly between the two architectures.

Table 2. Model performance under varying positive sample ratios.

Positive Ratio	XGBoost F1	XGBoost AUC	Transformer F1	Transformer AUC
10%	0.195	0.656	0.170	0.630
6%	0.193	0.657	0.172	0.630
3%	0.183	0.642	0.164	0.627
1%	0.162	0.596	0.163	0.619

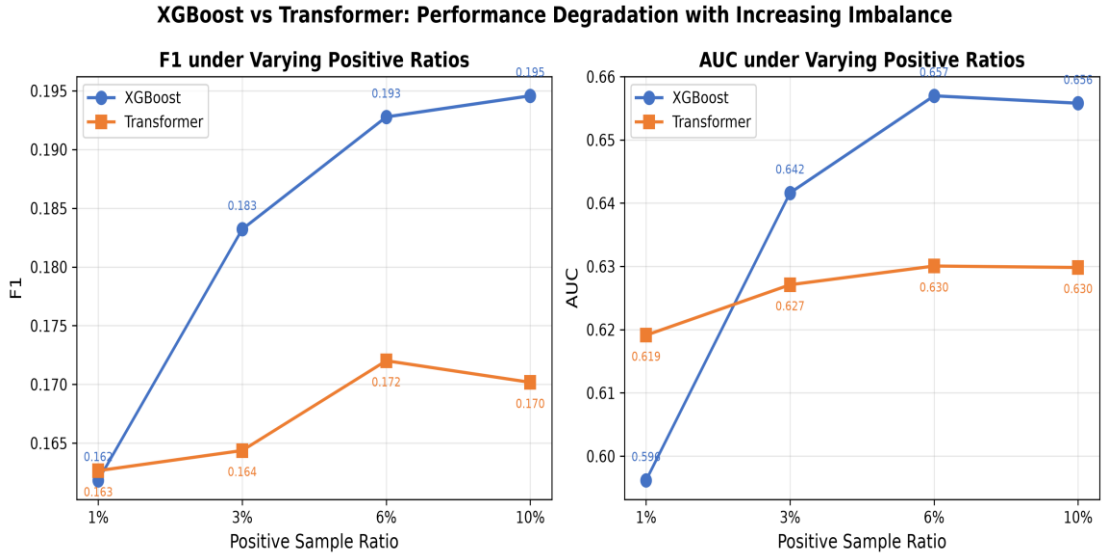

Figure 2. Performance degradation of XGBoost and Transformer as the positive sample ratio decreases from 10% to 1%.

As imbalance intensified from 10% to 1% positive, XGBoost's AUC fell from 0.656 to 0.596 (↓9.1%), reflecting the sharp decline in usable positive training examples. Transformer's AUC, by comparison, slipped only from 0.630 to 0.619 (↓1.7%), and its F1 stayed nearly flat (0.170 at 10%, 0.163 at 1%). At the 1% level, Transformer's AUC (0.619) surpassed XGBoost's (0.596). RQ3 thus points to Transformer as the more robust architecture under extreme class imbalance: its performance held steady where XGBoost's deteriorated markedly.

4.3 Feature Importance and SMOTE Ablation

4.3.1 Feature Importance

As shown in Figure 3, purchase_count ranks first in the importance ranking of XGBoost, with a score of 0.249. This is almost twice as much as 0.129 of the second characteristic quantity total_interactions. The first five features (purchase_count, total_interactions, merchant_total_actions, merchant_unique_users, and merchant_purchase_ratio) captured 51.7% of total importance. This skewed distribution suggests that the discriminative signal concentrates in a small subset of statistical features, with the remaining 17 features contributing marginally.

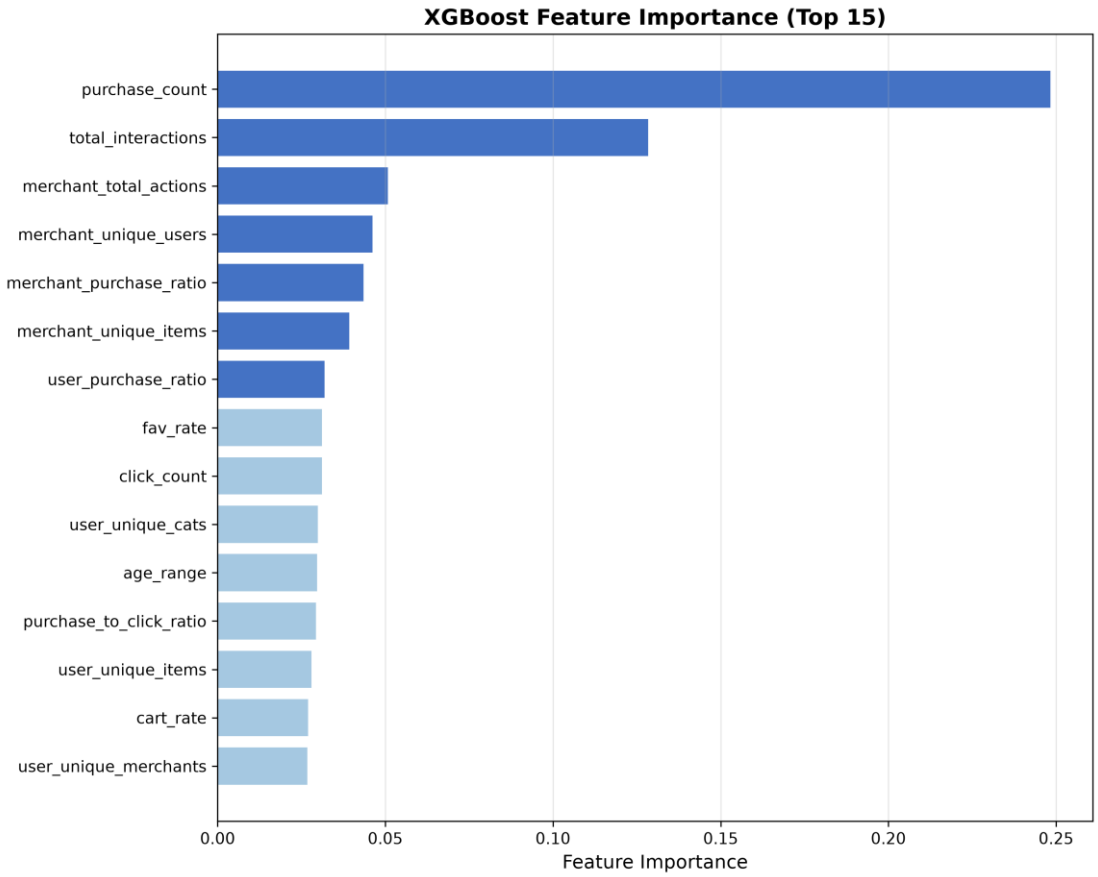

Figure 3. Top 15 feature importance scores from XGBoost trained on the full imbalanced dataset.

4.3.2 SMOTE Ablation

If the negative SMOTE effect in the main experiment arose from the particular sampling ratio chosen, varying that ratio should produce different outcomes. It did not. As reported in Table 3 and visualized in Fig. 4, all three SMOTE ratios (1:3, 1:2, 1:1) underperformed the no-SMOTE baseline (AUC 0.657), and AUC declined monotonically as the proportion of synthetic samples increased: 0.619 at 1:3, 0.613 at 1:2, and 0.598 at 1:1. This monotonic pattern rules out ratio selection as a confounding factor. The negative effect reflects a structural limitation of SMOTE in this feature space, not a parameter choice. The total training size grew alongside the synthetic proportion (from approximately 209K samples in the baseline to 392K at the 1:1 ratio), making it difficult to attribute the exact magnitude of AUC decline to synthetic noise alone. The monotonic pattern, however, holds at every tested ratio.

Table 3. Ablation study on SMOTE sampling ratios.

Strategy	AUC	F1	Recall	Precision
No SMOTE (baseline)	0.657	0.194	0.344	0.135
SMOTE 1:3	0.619	0.167	0.281	0.119
SMOTE 1:2	0.613	0.162	0.322	0.108
SMOTE 1:1	0.598	0.150	0.352	0.095

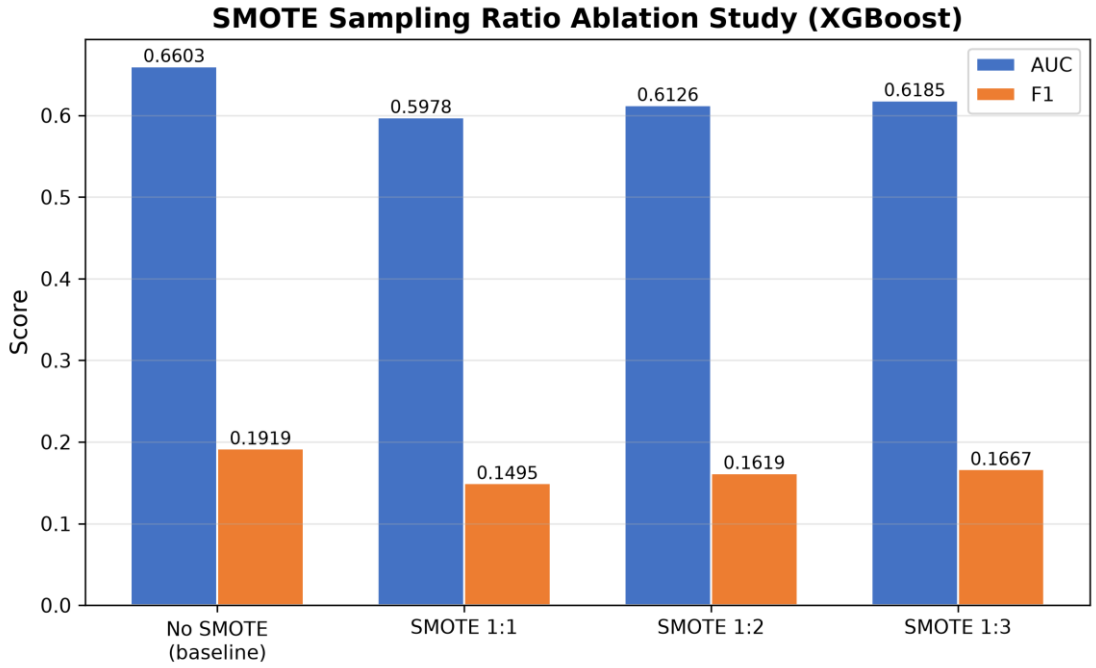

Figure 4. AUC and F1 scores under different SMOTE sampling ratios.

4.4 Sequence Removal Experiment

To isolate the contribution of sequence information, all behavioral sequences in the training and test sets were replaced with padding tokens, and both models were retrained under the 1% and 3% positive-rate conditions following the subsampling procedure of Section 4.2. Because XGBoost consumes only the 22 tabular features, its performance matched the degradation experiment (AUC 0.596 at 1%, 0.642 at 3%). The sequence-free Transformer reached an AUC of 0.616 at the 1% level—barely below the 0.619 achieved with real sequences and still above XGBoost's 0.596. At the 3% level, it scored 0.626 against 0.627 with sequences, with XGBoost leading at 0.642. Removing sequences thus changed Transformer's AUC by less than 0.004 at both levels: the robustness advantage at extreme imbalance persists even without behavioral sequences, and sequence features contribute only limited marginal gain.

5. DISCUSSION

5.1 Why XGBoost Outperforms

Three structural factors favor XGBoost on this task. Tree-based models handle tabular data with natural robustness to missing values and uninformative features, and they preserve feature directionality [4]. Cost-sensitive weighting scales class importance at the gradient level without injecting synthetic noise, which gives it a clean advantage over SMOTE: the gap between the original-data and SMOTE-trained XGBoost (AUC 0.657 vs. 0.598) makes the cost of synthetic noise tangible. The feature importance analysis (Section 4.3) points to a third factor: discriminative signal concentrates in a handful of statistical features, notably purchase_count, while the 20-length behavioral sequence contributes limited marginal gain. This goes some way toward explaining why the Transformer's sequence modeling capacity did not yield a corresponding performance edge.

5.2 Comparison with Prior Work

The two previous study used XGBoost on the same Tmall dataset. Jing and Shi [7] pay attention to the optimization of feature engineering. Ye et al. [15] improved the parameter adjustment and achieved an accuracy of 71.3%. Neither study takes class imbalance as the main variable, nor does it consider other methods of using Transformer. Gayathri A. Unni et al. [9] combined SMOTE, XGBoost and collaborative filtering, and published good results. However, the positive rate of their dataset is quite high, and their analysis is not only to investigate one of the influences of imbalance, but to focus on

the advantages of model fusion. In this study, four levels of imbalance will be compared in a controlled way to make clear which architecture is the better choice under which conditions.

5.3 Why Does SMOTE Fail?

The failure pattern points to the geometry of the feature space. When discriminative information concentrates in a handful of features, as the importance analysis in Section 4.3 showed, SMOTE's 22-dimensional interpolation generates samples that sit between classes rather than within the minority class—blurring the decision boundary instead of sharpening it. Blagus and Lusa [6] recorded that the effect of SMOTE will also weaken in high-dimensional settings. It is found that the reason is that the dispersion of the samples is reduced without increasing information.

This study not only extends this discovery to simple tabular data, but also to the mixed feature space that is common in e-commerce prediction. Here, the arrangement of behavior and statistical characteristics exist together. Importantly, the results of ablation confirm that the performance degradation is not a problem of parameter adjustment. Whatever the ratio, this trend has not changed. From these evidences, we can see that SMOTE has structural limitations in the feature space where information is less concentrated.

5.4 Where Does Transformer's Robustness Come From?

The advantage of the robustness of Section 4.2 records comes from how the types of two features decline when the data is less. When the number of positive samples decreases, the tabular features (most importantly, purchase_count) related to interaction counts will lose their distinguishing ability immediately. Because these reliabilities are proportional to the size of the sample. On the contrary, Behavioral sequences remembers that no matter how many positive examples there are, the time pattern remains stable. The process of "click → add-to-cart → purchase" has useful information whether it appears in 50 positive training examples or 5000. self-attention mechanism will continue to capture these relationships and sum them up. The result of this decline will object to the general idea that "when data classes are biased, tree-based models are inherently good". When the positive percentage is less than 3% and Behavioral sequence data can be used, the architecture based on Transformer deserves careful consideration. Additional experiments using tables only (Section 4.4) will add details to this description. Even if the Behavioral sequences are replaced by filling tags, the Transformer maintains its strong advantages in 1% extreme cases. This shows that the contribution of deep feedforward classifier to robustness has nothing to do with the coding of sequence. Behavioral sequences have information, but the effect is smaller than that inferred from the decline curve alone.

5.5 Practical Implications

If the experimental results are actually used, three model selection rules can be formulated. That is summarized in Table 4. First, when the positioning rate is higher than 5%, XGBoost with cost-sensitive weighting remains the safest choice. Only tabular features have the necessary information, and this model does not belong to the problem of synthetic sampling. Second, when the positive rate is lower than 3%, the Transformer is more robust with or without behavioral sequence. The experiment of eliminating sequence (Section 4.4) shows that even when there is no meaningful information in sequence, in 1% extreme cases, Transformer maintains better performance than XGBoost. Third, SMOTE should be used cautiously when the discriminative signal concentrates in few features and the surrounding feature space is predominantly noise, as the interpolation is then more likely to blur the decision boundary than to enrich the minority class.

Table 4. Recommended model selection strategy by imbalance severity.

Imbalance Severity	Positive Rate	Recommended Approach
Moderate	> 5%	XGBoost + cost-sensitive weighting
Severe	< 3%, with behavioral sequences	Transformer (original data)
Severe	< 3%, without behavioral sequences	Transformer (original data)

Note: The "without behavioral sequences" recommendation is based on the sequence-removal experiment reported in Section 4.4.

5.6 Limitations

Five limitations bound the present findings. First, the experiments relied on a single e-commerce dataset; cross-platform validation is needed to establish generalizability. Second, only a basic Transformer architecture was tested; pre-trained

sequential models such as BERT4Rec [16] may behave differently under severe class imbalance. Third, Focal Loss [13] was not evaluated and offers a natural next benchmark. Fourth, hybrid architectures combining XGBoost's tabular strength with Transformer's sequence modeling capability remain unexplored in the imbalanced setting. Fifth, in the degradation experiment (Section 4.2), varying the positive rate also changed the total training set size, because the subsampling procedure retained all instances of one class while downsampling the other. This confound between class ratio and training size may modestly attenuate the observed degradation curves. A future replication that fixes total training size by subsampling both classes to achieve each target ratio would isolate the pure effect of imbalance severity.

6. CONCLUSION

This study compared XGBoost and Transformer for e-commerce purchase prediction under controlled class imbalance. On the original data (6.12% positive rate), XGBoost led on both AUC and F1, confirming the strength of tree-based models on tabular features in this setting. SMOTE, applied as a standard remedy, backfired: both models lost performance, and ablation experiments traced the degradation to the geometry of the feature space—synthetic interpolation in a 22-dimensional space where signal concentrates in few features blurred rather than sharpened the decision boundary. The finding extends prior work on SMOTE's high-dimensional limits [6] to hybrid feature spaces combining tabular and sequential data.

At extreme imbalance (1% positive rate), Transformer proved more robust, its AUC falling by 1.7% against XGBoost's 9.1%, and supplementary experiments confirmed that this advantage persisted even when behavioral sequences were removed. Taken together, the results support two practical guidelines. At positive rates above 5%, XGBoost with cost-sensitive weighting is the safer choice. At rates below 3%, Transformer offers greater robustness; behavioral sequences, while helpful, are not required for this advantage to hold. SMOTE should be approached with caution when the discriminative signal concentrates in a small number of features, regardless of total dimensionality.

REFERENCE

[1] Zhang, X., Guo, F., Chen, T., Pan, L., Beliakov, G., Wu, J. (2023) A brief survey of machine learning and deep learning techniques for e-commerce research. J. Theor. Appl. Electron. Comm. Res., 18: 2188-2216.

[2] Chen, T., Guestrin, C. (2016) XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco. pp. 785-794.

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I. (2017) Attention is all you need. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017). Long Beach. pp. 5998-6008.

[4] Grinsztajn, L., Oyallon, E., Varoquaux, G. (2022) Why do tree-based models still outperform deep learning on typical tabular data? In: Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks. New Orleans. pp. 1-10.

[5] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. (2002) SMOTE: Synthetic minority over-sampling technique. In: Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI'02). Seattle. pp. 321-357.

[6] Blagus, R., Lusa, L. (2013) SMOTE for high-dimensional class-imbalanced data. BMC Bioinformatics, 14: 106.

[7] Jing, X.L., Shi, M.X. (2023) Repurchase prediction of e-commerce user based on XGBoost. J. Liaoning Univ. (Nat. Sci. Ed.), 50(2): 134-145.

[8] Tanvir, A.A., Khandokar, I.A., Islam, A.K.M.M., Islam, S., Shatabda, S. (2023) A gradient boosting classifier for purchase intention prediction of online shoppers. Heliyon, 9(6): e15163.

[9] Gayathri A Unni, Akshaya, A., Sabu, H., Krishnan, M.S. (2024) Optimized learning for e-commerce sales failure prediction: An XGBoost perspective. In: Proceedings of the 2024 International Conference on Computing and Data Science (ICCDS). Kochi. pp. 1-6.

[10] Kang, W.C., McAuley, J. (2018) Self-attentive sequential recommendation. In: Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM). Singapore. pp. 197-206.

[11] Chen, Q., Zhao, H., Li, W., Huang, P., Ou, W. (2019) Behavior sequence transformer for e-commerce recommendation in Alibaba. In: Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data (DLP-KDD'19). Anchorage. pp. 1-4.

[12] He, H., Garcia, E.A. (2009) Learning from imbalanced data. IEEE Trans. Knowl. Data Eng., 21(9): 1263-1284.

[13] Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P. (2017) Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). Venice. pp. 2980-2988.
[14] Alibaba Cloud Tianchi. (2018) Tmall Repeat Purchase Prediction Competition. https://tianchi.aliyun.com/competition/entrance/231576.
[15] Ye, H., Yuan, K.J., Shen, W., Zheng, Z.Q., Zhang, H.J. (2025) Research on repurchase behavior prediction of e-commerce platform users based on improved XGBoost algorithm. Communications and Information Technology, (2): 11-16. [in Chinese]
[16] Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P. (2019) BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM'19). Beijing. pp. 1-10.