Customer Segmentation and Personalized Recommendation Based on Machine Learning

Lingyun Li *

Software Engineering, Xiamen University Malaysia, Sepang, Malaysia

* Corresponding Author Email: SWE2409027@xmu.edu.my

Abstract. Today, with the rapid development of digitalization and networking, enterprises and institutions can collect customer data on an unprecedented scale, including demographic information, consumption behavior records, interaction logs, and social media information. These data provide a rich foundation for in-depth understanding of customer behavior and prediction of future demands. However, in the face of high-dimensional, multi-modal and dynamic data, how to effectively extract information, conduct refined customer segmentation and provide efficient personalized recommendations has become a research hotspot of common concern in both academia and industry. This paper systematically reviews the research progress of machine learning in customer segmentation and personalized recommendation in the past five years (2020-2025), covering aspects such as data feature construction, unsupervised and supervised clustering methods, gradient boosting tree models (LightGBM, CatBoost), deep neural networks (Transformer, GNN), core architectures and hybrid strategies of recommendation systems, model evaluation methods, interpretability and privacy protection. At the same time, it combines actual cases from industries such as retail, e-commerce, finance and healthcare to analyze the application effects and challenges of algorithms in real business, and looks forward to future development trends such as real-time personalization, multi-modal fusion, federated learning and green AI.

Keywords: Customer Segmentation, Personalized Recommendation, Machine Learning, Deep Learning, Recommendation System.

1. Introduction

With the advancement of the global digitalization trend, the operational environment of enterprises has undergone significant changes. The explosive growth of customer data has opened up opportunities for precise marketing. According to Statista's data, the global digital data volume is expected to reach 175 ZB by 2025, with customer data accounting for an important part, not only in terms of quantity but also in terms of variety. At the same time, enterprises have accumulated rich user profiles, transaction behaviors, browsing logs, and social interaction data in areas such as e-commerce, financial services, healthcare, and media entertainment.

Traditional marketing relies on empirical rules and coarse-grained clustering techniques, such as the RFM (Recency, Frequency, Monetary) model, which classifies customers based on three dimensions: the most recent purchase time, purchase frequency, and consumption amount. It is simple and effective, and has been widely adopted by enterprises for a long time. However, the RFM model is fundamentally based on simple statistics and is difficult to capture the complexity and dynamic changes of user behavior. When faced with increasingly diverse and high-dimensional customer data, it shows significant limitations.

With the rapid development of machine learning and artificial intelligence, academia and industry have introduced complex models and algorithms to deeply mine and intelligently analyze customer data, achieving more refined customer segmentation and personalized recommendations. Machine learning technology, through nonlinear modeling of high-dimensional, multimodal, and time-series data, significantly improves the accuracy of user interest identification and demand prediction, creating considerable business value for enterprises.

For example, Alibaba Group has built a customer profiling system based on machine learning, achieving multi-level clustering and precise recommendations for hundreds of millions of users, significantly improving conversion rates and user stickiness. Amazon's personalized

recommendations contribute over 35% of sales, and Netflix has achieved personalized and diversified content recommendations through deep learning, greatly enhancing user retention and satisfaction.

Therefore, in-depth research and systematic summary of machine learning-based customer segmentation and personalized recommendation technologies not only have significant theoretical significance but also possess extensive industrial application value and social impact.

Over the past five years, a large number of innovative algorithms and theoretical achievements have emerged in the field of customer segmentation and recommendation systems. Unsupervised clustering algorithms have expanded from the classic K-Means [1] to deep clustering methods (such as clustering based on autoencoders [2]) and graph neural network clustering [3], effectively handling non-linear and complex data structures. Supervised learning and semi-supervised learning methods combine business labels to enhance the targeting of clustering.

Recommendation systems have evolved from traditional collaborative filtering to sequence-based recommendation based on deep learning (such as the BERT4Rec architecture [4]), achieving long-distance dependency modeling of user behavior time series. Hybrid recommendation strategies combine recall, sorting, and re-ranking to comprehensively improve recommendation quality. The rise of privacy protection technologies such as federated learning and differential privacy provides theoretical guarantees for the secure modeling of large-scale sensitive data [5, 6].

In addition, issues such as model interpretability and model fairness have gradually become hot topics, promoting the applicability and compliance of algorithms in key fields. Representative academic works include: Chen et al. proposed XGBoost, which greatly promoted the application of GBDT in recommendation tasks [7]; Ke et al. developed LightGBM, which is efficient and resource-consumption-limited and has become the mainstay in industry [8]; Prokhorenkova et al. published CatBoost, which natively handles categorical features, alleviating the pressure of feature engineering [9]; Kang et al. applied the variational autoencoder to deep embedding clustering, enhancing the nonlinear modeling ability for customer segmentation [2]; He et al. proposed the neural collaborative filtering (NCF) that integrates the advantages of traditional matrix factorization and deep learning [9].

The core challenges faced by the industry include large-scale data, diverse features, strict requirements for online inference latency, and data privacy compliance. Top internet companies have built complex multi-stage recommendation architectures: In the recall stage, neighbor search, embedding retrieval, and collaborative filtering are utilized to quickly generate candidate sets and ensure real-time performance; in the sorting stage, GBDT or deep neural network models are employed to finely rank the candidates, with optimization guided by business indicators such as CTR and CVR; finally, in the re-ranking and post-processing stage, outputs from multiple models are integrated to achieve adjustments for diversity and fairness.

For example, Alibaba's "One Person, One Face" recommendation system achieved real-time updates of user profiles and diversified recommendations, relying on a powerful big data platform and machine learning pipeline. Netflix utilized large-scale distributed training and real-time log stream processing technologies to continuously optimize recommendation models.

Furthermore, user privacy regulations such as GDPR and CCPA require enterprises to strictly control privacy risks in data processing, prompting the rapid implementation of federated learning, differential privacy, and other technologies in the industry.

This paper aims to comprehensively review and summarize the customer segmentation and personalized recommendation technologies based on machine learning in the past five years through a systematic review. The highlights of the content include Systematic review of the main research progress and algorithm development in the past five years; Comparing the advantages and disadvantages of different methods in terms of data types, performance, interpretability, and industrial usability; Analyzing the implementation and challenges of algorithms through real case studies; Proposing future research directions and potential solutions.

2. Data Sources and Feature Types

In customer segmentation and recommendation systems, data sources typically include demographic data such as age, gender, occupation, education, and income, which provide static but valuable background information [1]; transactional data like purchase amounts, frequency, categories, and timestamps, reflecting actual consumption behavior [10]; interaction data such as clicks, browsing, cart additions, and reviews, which reveal preferences and potential needs [7]; contextual and temporal data including holidays, seasons, weather, and location, which influence consumer decisions [9]; and external data such as economic indicators, competitor pricing, and social media sentiment [8]. To effectively utilize these, various feature types and encoding strategies are employed: numerical features are typically scaled with Min-Max or Z-score normalization to address scale differences [2]; categorical features are encoded using methods such as One-Hot for low cardinality, Target Encoding with safeguards against overfitting, CatBoost Encoding to avoid leakage, and neural embeddings to learn dense vector representations [3, 4]; textual features are processed with TF-IDF, word embeddings (e.g., Word2Vec, GloVe), or advanced contextual embeddings from pretrained language models like BERT and GPT [11, 12]; while image and video features are extracted using deep CNNs such as ResNet, EfficientNet, or ViT, and combined with temporal models for video data [5].

3. Feature Engineering and Data Processing

3.1. Feature Construction

Feature engineering in customer segmentation and recommendation systems often involves statistical features, such as total purchase count, total spending, average order value, and repurchase rates; time decay features, which apply exponential decay functions to emphasize recent behaviors while gradually reducing the influence of older actions, controlled by a decay coefficient [13]; cross features, which capture nonlinear interactions by combining variables (e.g., income × education) or concatenating categorical attributes, with automated tools frequently used to generate large numbers of such features to uncover complex patterns [14]; behavior sequence features, which leverage ordered user actions such as recent clicks or purchases as inputs for deep learning models; and contextual features, which integrate external factors like weather, holidays, and location to better model user behavior.

3.2. Encoding and Normalization

To enhance model performance, it is important to apply appropriate encoding methods based on feature types and model requirements [3], normalize numerical features to stabilize training and improve convergence [9], and use embedding techniques for high-cardinality categorical features to reduce dimensionality while capturing semantic relationships [4].

3.3. Feature Selection

Feature selection can be enhanced by employing filtering methods such as correlation, variance, and chi-square tests, wrapper methods like recursive elimination, and embedded methods including tree-based importance and L1 regularization to remove redundant and noisy features, while also leveraging automated feature engineering and AutoML tools to enable efficient feature discovery and selection [15].

3.4. Industrial Applications

Alibaba E-commerce: Implements automated daily pipelines generating thousands of features; uses time decay to model interest fade, boosting ad click rates; combines cross features from user profiles and item attributes to optimize recommendation rankings [16].

ByteDance: Applies Z-score normalization and embeddings for high-cardinality categorical features; integrates encoding into scalable feature platforms for real-time updates [17].

JD.com: Utilizes LightGBM for dynamic weekly feature importance evaluation; employs recursive feature elimination to reduce features by over 70%, achieving a threefold training speedup [8].

4. Customer Segmentation Methods

Customer segmentation is a foundational step for precision marketing. The goal is to divide customers into homogeneous groups with high internal similarity and significant differences between groups. Traditionally dominated by unsupervised clustering, recent developments integrate supervised and deep learning methods.

4.1. Unsupervised Clustering Methods

4.1.1. K-Means Clustering:

K-Means clustering aims to minimize the sum of squared distances between samples and their cluster centroids. The process involves randomly initializing k cluster centers, assigning each sample to the nearest centroid based on distance, updating centroids as the mean of assigned samples, and repeating these steps until convergence or reaching an iteration limit. K-Means is efficient, easy to implement, and scalable to large datasets, but it requires a preset cluster count k, is sensitive to outliers, assumes convex spherical clusters, and struggles with complex shapes; K-Means++ improves initialization to reduce the risk of local optima. For example, an e-commerce platform uses K-Means to cluster users based on transactions, age, and purchase frequency, identifying "high-frequency buyers" and "potential new customers" to guide recommendations, with the method's simplicity and low cost making it suitable for offline analysis.

4.1.2. Hierarchical Clustering:

Hierarchical clustering can be agglomerative (bottom-up), starting with each sample as a cluster and iteratively merging the closest clusters, or divisive (top-down), starting with all samples in one cluster and recursively splitting them. Distance metrics such as single-linkage, complete-linkage, and average-linkage are used to guide merging or splitting. The process involves computing a pairwise distance matrix, merging or splitting clusters based on distances, and generating a dendrogram from which different cluster levels can be selected. Hierarchical clustering does not require a predefined number of clusters, produces a hierarchical structure, and works well for small datasets, but it has high computational cost (up to $O(n^3)$), is sensitive to noise, and is less suitable for large-scale data, though optimizations for scalability have been proposed by Müllner (2011). For example, financial institutions use hierarchical clustering to segment high-risk customers, understand risk hierarchies, and support loan approval and risk control, primarily in offline analyses due to resource demands.

4.1.3. Density-Based Clustering (DBSCAN, HDBSCAN):

Density-based clustering identifies clusters as dense regions separated by low-density areas [10]. DBSCAN defines core points (with sufficient neighbors within a radius ε), border points, and noise points, while HDBSCAN extends DBSCAN to handle varying densities. The process involves calculating neighborhood density for each point, labeling core, border, and noise points, expanding clusters from core points to include all density-reachable points, and iterating until all points are assigned. This approach can automatically detect the number of clusters, find arbitrarily shaped clusters, and is robust to noise, but it is sensitive to parameter settings, computationally intensive, and struggles in high-dimensional spaces. For example, an online ad platform uses density-based clustering to group user behavior trajectories, identifying active, dormant, and anomalous users to improve targeting accuracy.

4.1.4. Gaussian Mixture Model (GMM):

Gaussian Mixture Model (GMM) clustering assumes that data are generated from a mixture of Gaussian distributions, with each cluster corresponding to one Gaussian, and estimates membership probabilities using the Expectation-Maximization (EM) algorithm. The process involves initializing mixture weights, means, and covariances; performing the E-step to compute posterior probabilities of cluster memberships; performing the M-step to update parameters to maximize the expected likelihood; and repeating these steps until convergence. GMM provides soft clustering, models overlapping clusters, and captures non-spherical distributions, but it is sensitive to initialization, computationally expensive, and less scalable. For example, banks use GMM soft clustering to model customer credit behavior, identify latent risk groups, and support risk alerts and differentiated credit policies.

4.2. Semi-supervised and Supervised Clustering Methods

4.2.1. Constrained Clustering:

Constraint-based clustering enhances traditional clustering by incorporating prior knowledge through constraints, such as must-link, which requires certain samples to be in the same cluster, and cannot-link, which requires certain samples to be in different clusters.

This approach typically modifies algorithms like K-Means or spectral clustering by adjusting distances or cluster assignments to respect these constraints. For example, e-commerce platforms can improve clustering results by integrating business rules or expert knowledge as constraints.

4.2.2. Supervised Classification for Clustering:

When partial labels or historical segments exist, supervised models can predict cluster labels for new users. Common models include gradient boosting trees like LightGBM and CatBoost [2, 8], which handles high-dimensional categorical features trains quickly and update online and offer good interpretability.

4.3. Deep Clustering Methods

4.3.1. Autoencoder-based Clustering:

Neural networks compress high-dimensional data into low-dimensional latent space, minimizing reconstruction error. Clustering is performed in this latent space. Variants include Variational Autoencoders (VAE) that enhance generative ability [18].

4.3.2. Graph Neural Networks (GNN):

Utilize node features and graph structure for representation learning, suitable for user-item interaction graphs and social networks.

Example: PinSage embeds user behaviors within graph structures for personalized segmentation and recommendation [19].

5. Personalized Recommendation Methods

Personalized recommendation systems aim to provide accurate, preference-aligned content or products. As data volume and complexity grow, recommendation technologies evolved from traditional collaborative filtering to multi-stage deep learning pipelines. Core tasks include candidate recall, ranking, and multi-objective optimization.

5.1. Collaborative Filtering (CF)

5.1.1. User-based Collaborative Filtering:

User-based collaborative filtering (CF) relies on the principle that users with similar interests tend to like similar items. For a target user, the system identifies similar users (“neighbors”) based on rating or behavior similarity, often measured by cosine similarity over commonly rated items,

and predicts ratings as a weighted average of neighbors' ratings. The process involves computing a similarity matrix for all user pairs, selecting the top-K neighbors for the target user, aggregating neighbors' ratings on unseen items for prediction, and generating a top-N recommendation list. This approach is simple, interpretable, and grounded in real user behavior, but it can be computationally expensive and suffers from cold start and data sparsity issues. For example, a video site implemented a "users who watched this also watched" feature using user-based CF, which increased user engagement by 20%.

5.1.2. Item-based Collaborative Filtering:

Item-based collaborative filtering recommends items similar to those a user has liked by computing item–item similarity measures such as cosine similarity or Pearson correlation [20]. Its advantages include a smaller similarity matrix, more stable computation, and consistent recommendations, while its drawbacks are weaker performance for long-tail items and persistent cold-start issues. A common use case is in e-commerce platforms, where "You may like" modules suggest related products, leading to a 15% improvement in repurchase rates through near real-time similarity updates.

5.1.3. Matrix Factorization (MF):

Matrix factorization decomposes the user–item rating matrix into low-dimensional user and item latent factors to capture latent attributes, with optimization typically performed using ALS or SGD alongside regularization to prevent overfitting, and predictions generated as the inner product of latent vectors. The process involves randomly initializing latent matrices, alternately updating user and item latent factors until convergence, and then predicting unknown ratings through dot products. This method offers high model capacity, effectively captures complex interactions, and outperforms traditional collaborative filtering, though it depends on sufficient rating data and faces challenges with sparsity. For example, Netflix applies matrix factorization with distributed ALS training to scale across billions of users and items, thereby significantly improving recommendation accuracy [21].

5.2. Content-based Recommendation

Content-based recommendations work on the principle of suggesting items with explicit features similar to those the user has previously liked, making it particularly effective for cold-start items. Its advantages include high interpretability and strong performance with new items, while its drawbacks lie in heavy reliance on expert-driven feature engineering and the neglect of collaborative signals. A practical example is news recommendation, where article topics and keywords are matched with user interest vectors, resulting in a 10% increase in click-through rate [22].

5.3. Model-based Recommendation

5.3.1. Gradient Boosted Decision Trees (GBDT):

Gradient Boosting Decision Trees (GBDT) operate on the principle of an additive model that combines multiple decision trees, where each tree fits the residuals of its predecessors, with efficient implementations such as LightGBM and CatBoost designed to handle categorical features and large-scale data [6, 7]. GBDT offers advantages including fast training, the ability to manage high-dimensional sparse data, and interpretability, though it is weaker in modeling sequential dependencies. For example, Taobao employs LightGBM for candidate ranking, leveraging user profiles and behavior sequences to improve click rate by 8% while maintaining online latency under 100 ms.

5.3.2. Deep Neural Network-based Recommendation:

Models such as Wide & Deep, which combines linear (wide) and deep models to balance memorization and generalization, DeepFM, which merges factorization machines with DNNs to

automatically capture feature interactions, and Transformer-based approaches like BERT4Rec, which leverage self-attention to model long-range dependencies in user behavior sequences, have been widely adopted in recommendation systems. These methods offer strong representational power and effectively capture complex interactions and sequential patterns, but they also come with disadvantages such as high training costs and the need for large-scale data. For example, Spotify applies Transformer models on user listening sequences to enhance recommendation diversity and accuracy, utilizing multi-GPU clusters while maintaining online latency at around 200 ms.

5.4. Hybrid Recommendation Methods

5.4.1. Typical Architecture:

In the recall stage, candidates are quickly generated using collaborative filtering and content features, followed by the ranking stage, where GBDT or deep models are applied to refine the candidate list; subsequently, multi-objective optimization is performed to jointly optimize CTR, conversion, and user experience. For instance, Amazon employs a two-stage recall and ranking system that ensures both high coverage and precision, leveraging distributed stream processing to deliver timely recommendations.

6. In-depth Analysis of Algorithm Implementation and Challenges: Insights from Real-World Case Studies

6.1. Practical Challenges in Large-Scale Industrial Deployment

Many recommendation systems struggle with extreme sparsity in user–item interaction matrices, where traditional matrix factorization methods (e.g., ALS, SGD-optimized MF) can capture latent relationships but perform poorly on ultra-sparse datasets; to address this, companies such as Spotify have adopted Transformer-based sequential models (e.g., BERT4Rec), which leverage self-attention to capture long-range user behavior dependencies and significantly improve recall quality. However, these models require substantial computational resources, making distributed multi-GPU training and mixed precision techniques essential for maintaining efficiency and controlling online latency.

Feature engineering remains critical to model performance; for example, Bytedance combines Z-score normalization with embedding encoding to stabilize numerical features and represent high-cardinality categorical variables, while incorporating time decay features to capture evolving user interests. However, managing thousands of complex features in a real-time streaming environment introduces significant architectural challenges, necessitating high-throughput feature platforms with strong consistency guarantees to avoid cold-start and stale feature issues [17].

Density-based clustering methods such as DBSCAN and HDBSCAN are highly sensitive to neighborhood radius and minimum points parameters, where incorrect settings may lead to over- or under-clustering; however, traditional grid search is infeasible in dynamic online scenarios, making automated hyperparameter optimization with Bayesian methods and reinforcement learning increasingly mainstream. For example, JD.com dynamically refines feature sets on a weekly basis using LightGBM feature importance and Recursive Feature Elimination (RFE), thereby balancing training speed with model generalization [8].

Platforms such as Taobao rely heavily on interpretable tree-based models like LightGBM for candidate ranking, as they facilitate feature contribution analysis and anomaly detection, which are crucial for compliance and risk control; in contrast, deep learning models, particularly multi-layer Transformers, provide superior accuracy but often function as “black boxes,” thereby limiting their adoption in regulated sectors such as finance and healthcare where transparency is paramount.

6.2. Technical Case Studies

Netflix employs distributed ALS for matrix factorization over billions of user-item pairs, incorporating regularization to mitigate overfitting. Time decay features adjust user preferences dynamically, enhancing recommendation relevance over time [21].

Alibaba implements an automated feature pipeline that uses exponential decay weights to emphasize recent user behaviors, effectively capturing interest decay and improving ad click-through rates. The pipeline supports automated refresh and incremental computation to keep recommendations timely [16].

Amazon's Two-Stage Recall and Ranking Architecture: Amazon integrates collaborative filtering and content-based models for rapid candidate recall, followed by sophisticated ranking using GBDT and deep neural networks optimized for multiple objectives (CTR, CVR, user satisfaction). Their system employs coaching and asynchronous updates to achieve high concurrency and low latency.

7. Future Research Directions and Potential Solutions

7.1. Innovations in Deep Representation Learning

Traditional recommendation models largely rely on supervised learning with explicit user-item interaction labels, which often suffer from sparsity and noise issues. Recent advances in self-supervised learning have enabled models to leverage unlabeled data via proxy tasks such as masked item prediction and next-item prediction, greatly enhancing the robustness of learned embeddings. Moreover, contrastive learning techniques applied on graph structures have shown significant promise by increasing node discriminability in sparse interaction graphs, improving generalization across user cold-start scenarios.

Additionally, the fusion of multi-modal data—such as text, images, videos, and geospatial information—with behavioral logs remains an open and promising area. Designing neural architectures capable of effectively integrating heterogeneous data sources can deepen user preference modeling and alleviate cold-start issues.

7.2. Explainability and Fairness in Recommendation

While deep models deliver superior predictive performance, their "black-box" nature often hampers interpretability. Developing explainable recommendation models through attention mechanisms or post-hoc interpretability tools such as SHAP and LIME can provide transparent decision rationales, essential for regulated industries and enhancing user trust [23].

Moreover, fairness concerns have risen due to demographic biases embedded in training data, leading to potential discrimination in recommendations. Research on fairness-aware algorithms involves defining appropriate fairness metrics, employing bias mitigation techniques during training, and ensuring equitable treatment across user groups without compromising accuracy.

7.3. Efficient Online Learning and System Architecture

Real-time adaptation to dynamic user interests is crucial for maintaining recommendation relevance. Lightweight incremental learning frameworks, along with meta-learning strategies, enable rapid model updates for new users and items, avoiding costly full retraining [24].

Furthermore, edge computing has emerged as a key technology for low-latency and energy-efficient recommendations by deploying compact models through pruning, quantization, and knowledge distillation on mobile and IoT devices, broadening the application scope.

7.4. Automated Feature Engineering and Multi-Task Learning

Automated Machine Learning (AutoML) techniques for feature generation and selection reduce manual efforts, discovering high-value cross features and derived attributes to boost model performance.

Multi-task learning frameworks simultaneously optimize several business objectives, such as CTR, conversion, and retention, by uncovering shared representations and inherent correlations, thus enhancing overall recommendation quality.

7.5. Privacy Protection and Security Assurance

Amidst growing privacy regulations, federated learning enables collaborative model training without exposing raw user data, while differential privacy provides mathematically rigorous privacy guarantees during data processing, balancing utility and security.

Recommender systems are also vulnerable to adversarial attacks such as data poisoning and model manipulation. Developing robust detection and defense mechanisms is vital to safeguarding system integrity and trustworthiness.

8. Conclusion

In summary, this study has reviewed the principles, implementations, and real-world applications of customer segmentation and personalized recommendation algorithms, with a focus on gradient boosting methods such as LightGBM and CatBoost, as well as collaborative filtering and hybrid approaches. Through the analysis of industrial case studies, we have highlighted both the effectiveness and the challenges of deploying these algorithms in large-scale, dynamic, and heterogeneous environments.

The findings suggest that while current models offer significant advantages in predictive accuracy and handling complex feature interactions, practical deployment often encounters issues related to scalability, cold-start scenarios, data sparsity, and compliance with privacy regulations. Addressing these challenges will require advances in adaptive modeling, distributed training, and privacy-preserving computation.

Looking ahead, integrating interpretability, hybrid algorithm design, and domain adaptation techniques will be crucial for improving trustworthiness and maintaining long-term relevance in rapidly evolving markets. By bridging the gap between theoretical models and industrial requirements, future research can further enhance the efficiency, adaptability, and transparency of personalized recommendation systems, ultimately delivering greater value to both businesses and end-users.

References

[1] Arthur, D. and Vassilvitskii, S.: “k-means++: The advantages of careful seeding,” in *Proc. 18th Annual ACM-SIAM* Symposium *on Discrete Algorithms (SODA)*, 2007, pp. 1027–1035.

[2] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. and Gulin, A.: “CatBoost: unbiased boosting with categorical features,” in *Proc. 32nd Int. Conf. on Neural Information Processing Systems (NeurIPS)*, 2018, pp. 6638–6648.

[3] Hamilton, W., Ying, Z. and Leskovec, J.: “Inductive representation learning on large graphs,” in *Proc. 31st Int. Conf. on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 1025–1035.

[4] Sun, F., Liu, J., Wu, Z. and Wang, H.: “BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer,” in *Proc. 28th ACM Int. Conf. on Information and Knowledge Management (CIKM)*, 2019, pp. 1441–1450.

[5] He, K., Zhang, X., Ren, S. and Sun, J: “Deep residual learning for image recognition,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[6] Zhang, F., Duan, J. and Chen, Z.: “Feature engineering for purchase prediction in e-commerce,” *Expert Systems with Applications*, vol. 95, pp. 30–42, 2018.

[7] Chen, T. and Guestrin, C.: “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[8] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T. Y.: “LightGBM: A highly efficient gradient boosting decision tree,” in *Proc. 31st Conf. on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 3146–3154.

[9] He, X., Liao, L., Zhang, H., Nie, L., Hu, X. and Chua, T. S.: “Neural collaborative filtering,” in *Proc. 26th Int. Conf. on World Wide Web (WWW)*, 2017, pp. 173–182.

[10] [10] Ester, M., Kriegel, H.-P. Sander, J. and Xu, X.: “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, 1996, pp. 226–231.

[11] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: “Efficient estimation of word representations in vector space,” in *Proc. 1st Int. Conf. on Learning Representations (ICLR)*, 2013.

[12] Devlin, J., Chang, M. W., Lee, K. and Toutanova, K.: “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. 2019 Conf. of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019, pp. 4171–4186.

[13] Cui, S., Yang, X. and Guo, Y.: “User interest modeling with time decay for personalized recommendation,” Information *Sciences*, vol. 512, pp. 1106–1122, 2020.

[14] Cheng, H., Koc, L., Harmsen, J. et al.: “Wide & deep learning for recommender systems,” in *Proc. 1st Workshop on* Deep *Learning for Recommender Systems*, 2016, pp. 7–10.

[15] Hutter, F., Kotthoff, L. and Vanschoren, J.: *Automated Machine Learning: Methods, Systems, Challenges*. Springer, 2019.

[16] Liu, Z., Zhang, Y. and Yang, Q.: “Feature engineering for large-scale click-through rate prediction in Alibaba,” Data *Mining and Knowledge Discovery*, vol. 34, no. 3, pp. 1079–1099, 2020.

[17] Chen, J., Liu, M. and Liu, K.: “Real-time feature engineering for large-scale recommendation systems at ByteDance,” *IEEE Transactions on Big Data*, vol. 7, no. 4, pp. 852–863, 2021.

[18] Kingma, D. P. and Welling, M.: “Auto-encoding variational Bayes,” in *Proc. 2nd Int. Conf. on Learning* Representations *(*ICLR*)*, 2014.

[19] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L. and Leskovec, J.: “Graph convolutionnal neural networks for web-scale recommender systems,” in *Proc. 24th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining (KDD)*, 2018, pp. 974–983.

[20] Sarwar, B., Karypis, G., Konstan, J. and Riedl, J.: “Item-based collaborative filtering recommendation algorithms,” in *Proc. 10th Int. Conf. on World Wide Web (WWW)*, 2001, pp. 285–295.

[21] Koren, Y.: “Factorization meets the neighborhood: a multifaceted collaborative filtering model,” in *Proc. 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2008, pp. 426–434.

[22] Adomavicius, G. and Tuzhilin, A.: “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[23] Lundberg, S. and Lee, S. I.: “A unified approach to interpreting model predictions,” in *Proc. 31st Conf. on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 4765–4774.

[24] Finn, C., Abbeel, P. and Levine, S.: “Model-agnostic meta-learning for fast adaptation of deep networks,” in *Proc. 34th Int. Conf. on Machine Learning (ICML)*, 2017, pp. 1126–1135.