Customer-to-Manufacturer Product Innovation: Uncovering Personalized and Integrated Demands in E-Commerce Platforms

Gang Xue
School of Economics and Management
Tsinghua University
Beijing China
xuegang@sem.tsinghua.edu.cn

Yunhui Liu
School of Economics and Management
Tsinghua University
Beijing China
liuyunhui@tsinghua.edu.cn

Jian Chen*
School of Economics and Management
Tsinghua University
Beijing China
chenj@sem.tsinghua.edu.cn

Hao Hu
Department of Automation
Tsinghua University
Beijing China
h-hu23@mails.tsinghua.edu.cn

Yongzhi Qi
Department of Intelligent Supply Chain Y
JD.com
Beijing China
qiyongzhi1@jd.com

Ruoxing Zhang
Department of Intelligent Supply Chain Y
JD.com
Beijing China
zhangruoxing@jd.com

Ningxuan Kang
Department of Intelligent Supply Chain Y
JD.com
Beijing China
kangningxuan@jd.com

Ziruo Cui
Investment Banking Department
Dongguan Securities
Beijing China
17120571@bjtu.edu.cn

ABSTRACT

This study explores the Customer-to-Manufacturer (C2M) product innovation model within e-commerce platforms, focusing on the precise identification of personalized and integrated consumer demands. We propose a novel analytical framework that integrates deep customer segmentation with integrated mining of product and attribute preferences. In deep customer segmentation, we introduce a clustering loss function for outlier removal and a preference loss function, jointly optimizing the autoencoder and clustering algorithms to enhance the performance of user segmentation. For the integrated mining of product and attribute preferences, we employ sequential pattern mining techniques to reveal integrated product demands, while using the multinomial Logit model to analyze consumer purchase intentions, thereby precisely extracting integrated demands related to product attributes. Experimental results across multiple e-commerce datasets show exceptional performance in customer segmentation and integrated product purchase intentions. Our research offers a practical, data-driven approach for C2M product innovation.

CCS CONCEPTS

• Information systems • Information systems applications • Data mining • Data stream mining

*Corresponding author

WOODSTOCK'18, June, 2018, El Paso, Texas USA

https://doi.org/10.1145/1234567890

KEYWORDS

Customer-to-Manufacturer; Market segmentation; Personalized demands; Product innovation; User behavior modelling

1 Introduction

In the industrial internet epoch, the Customer to Manufacturer (C2M) paradigm serves as a shortcut economic model, directly linking consumers and manufacturers. This innovation disrupts conventional supply chain management and retail modalities. The essence of C2M lies in amplifying the consumer's dominant role, materializing the personalization and customization of products [1]. Under this paradigm, consumers transition from passive acceptors of market-available products to key influencers over product design and manufacturing processes [2]. C2M enables consumers to directly interface with manufacturers, conveying their unique demands and preferences, thereby fostering product innovation and diversification. This direct consumer-to-manufacturer connection not only facilitates the fulfillment of personalized demand but also brings new challenges and opportunities to the manufacturing sector [3].

E-commerce platforms, with their proprietary data and channel strengths, act as a critical conduit within the C2M model [3-4]. Through deep data analysis, they can precisely capture and forecast consumer trends towards personalization, providing manufacturers with invaluable market insights [5]. Consequently, the C2M model not only meets the rising demand for personalized and custom-made products but also propels the supply chain transformation from a traditional production-driven model to a demand-driven one, rendering the manufacturing process more agile and efficient [3,6].

As an exemplar, JD.com, a titan in the Chinese e-commerce industry, pivots its C2M platform around consumer demand to innovate supply chain management. It advances new products through phases of trend analysis, design, testing, and market feedback. The platform leverages big data and industry trends to provide manufacturers with detailed advice on pricing, style, target markets, and sales volumes. For example, JD.com's C2M platform, focusing on the shampoo market, precisely discerned the demands of a male consumer segment, setting targeted product key selling points, pricing, and packaging, which led to a 131% increase in sales revenue on the launch day compared to past performances (https://c2m.jd.com/caseDetail/15). Such strategy effectively reduces the product development cycle, enhances the speed of market response, and satisfies the users' personalized demands [3].

In the digitized era, the evolution of supply chains is being propelled by intelligent digital technologies. Supply chains have transformed into ecosystems, characterized by interdependent and complementary supply and demand relationships [7-8]. Leading enterprises are harnessing digital technologies such as mobile internet and cloud computing to interconnect various products and services, thereby meeting consumers' integrated demand [9]. For instance, intelligent toilets have gone beyond their traditional functions to provide health checks like urinalysis and urine flow rate tests, interacting with the healthcare services supply chain through digital devices. Similarly, Apple's vision Pro, by leveraging virtual and augmented reality technologies, fulfills integrated consumer demand by combining functionalities like high-performance gaming and private cinema experiences in one product. Hence, in a digitalized environment, businesses within the ecosystem must focus not only on the supply chain of single products or services but also on the supply chains of related products and services. Thus, C2M platforms should not only concentrate on meeting personalized demands for specific product markets but should also identify and satisfy cross-boundary integrated demands, adapting to the growing trend of diversified consumer consumption and driving the supply chain towards more intelligent and collaborative development.

To achieve personalization and integrated product innovation, C2M platforms must accurately identify target markets to provide corresponding integrated products and services. In this process, model development faces two major challenges: 1) How to accurately position target consumers. Existing user segmentation models, often based on background information such as gender, income, and age, can effectively capture the general demand for specific products and provide clear interpretability [10-11]. However, they tend to overlook the personalization and dynamic nature of user demands. Personalized recommendation systems based on user behavior may offer some insights, but they typically only analyze existing products and lack the ability to identify integrated demands [12-13]. To effectively identify and meet personalized and integrated demands, C2M platforms need to adopt a new type of user segmentation model that transcends traditional market segmentation standards. This model should emphasize the commonalities in user behavior patterns and have the capability to innovate product combinations, flexibly adapting to and integrating users' diverse demands. 2) How to determine which products and product attributes to integrate. The existing data-driven approach to product innovation utilizes online reviews to mine consumer preferences [10, 14]. However, methods based on review text mining face limitations: First, user reviews often contain rich personal emotions and subjective opinions, which may lead to product attribute analysis that cannot fully or objectively reflect users' true demands and preferences [15]. Second, review texts can only represent a minority of users who provide feedback, failing to cover the opinions and demands of all users comprehensively [16]. Furthermore, the content of reviews usually reflects users' past experiences and opinions, which may not capture the latest market trends and current user demand in a timely manner [17]. Therefore, seeking new methods to uncover unmet or unnoticed user demand is key. The challenge lies in how to obtain users' direct and explicit feedback and delve deeply into their implicit demands.

To address these challenges, our solution focuses on developing an advanced data analysis framework specifically designed to identify and meet personalized and integrated product demand. Our method includes two key components: 1) Deep customer segmentation: By conducting an in-depth analysis of user behavior data, we can segment consumers into groups with similar behavior patterns and preferences. Utilizing user behavior feature embedding, autoencoders, and clustering models, combined with specially designed clustering loss functions for outlier removal and preference loss functions to retain user preferences, we can more accurately identify target consumer groups and capture their implicit demand. 2) Integrated mining of product and attribute preferences: Building on deep customer segmentation, we use sequential pattern mining methods and attribute selection methods based on multinomial Logit models to deeply analyze users' clicking, carting, and purchasing behaviors. This not only reveals users' integrated demands for specific product categories but also helps us understand consumers' detailed preferences for product attributes.

The innovations of this study are summarized as follows: 1) We developed a novel analytical framework based on user behavior sequences, combining deep learning with data analysis techniques to reveal users' personalized and integrated product demand. This approach surpasses traditional methods based on user reviews or basic features, providing more in-depth and precise insights to understand users' complex and diverse demand. 2) We developed a deep segmentation model suitable for mining personalized integrated demands. By integrating autoencoders with clustering algorithms and introducing specially designed loss functions, we effectively remove outliers and retain cross-category preference patterns of users. This joint training approach significantly enhances the model's effectiveness in user segmentation and preference understanding, offering a powerful tool for more precisely capturing user preferences. 3) We constructed a model capable of accurately mining integrated product and attribute sets. Through sequential pattern analysis, we can identify unmet integrated product demands hidden within consumer purchasing and searching behavior patterns. Then, applying the multinomial Logit model, we can more precisely predict which attribute combinations are most likely to be favored by consumers. The core of this method is that it can mine product-level demands and accurately grasp attribute-level demands, thus providing strong data support for personalized and integrated product innovation.

2 Related works

1) Sequential user behavior modeling: In the field of sequential user behavior modeling, various methods have been employed to enhance the understanding and prediction of user actions. Wang et al. [12] utilized graph embedding techniques to address issues of scalability, sparsity, and cold start in data, improving model accuracy by mapping similarities between items. The Transformer model, adopted by Chen et al. [13], effectively captures sequential signals in user behavior, demonstrating superiority in processing sequential data. Yin et al. [18] introduced a transferable parameter learning approach targeting the long-tail distribution of user behavior data, incorporating gradient alignment optimizers and adversarial training schemes. Yuan et al. [19] developed PeterRec, a parameter-efficient transfer learning architecture, showcasing its flexibility in adapting to various downstream tasks. Bian et al. [20] proposed a contrastive curriculum learning framework that generates effective representations of user behavior sequences through data augmentation and curriculum learning strategies. Additionally, Baltescu et al. [21] employed item embedding methods in Pinterest's ItemSage product, combining text and image information to significantly enhance the relevancy of recommendations. The application of these methods indicates that the field of sequential user behavior modeling is evolving towards more accurate and effective understanding and prediction of user behavior. Particularly, the Transformer model and item embedding methods have shown remarkable advantages in feature extraction. The Transformer model's ability to process long sequence data and capture complex dependencies within sequences is crucial for understanding historical user behavior patterns. Item embedding methods enhance the understanding of user preferences by learning the embedded representations of items, effectively capturing the relationships between them. The integration and application of these technologies not only improve model performance but also deepen the predictive capabilities regarding user behavior.

2) Deep clustering: In the realm of deep clustering, numerous studies have significantly enhanced the performance of clustering algorithms through innovative methods, focusing on two main aspects: deep embedded representations and deep clustering constraints. Leiber et al. [22] delved into the application of deep clustering algorithms in handling complex and large-scale datasets, highlighting challenges in standard structures and programming overhead. Wang et al. [23] further emphasized the critical role of deep clustering in data mining, particularly how its robust neural network representation capabilities facilitate data processing. Regarding deep clustering constraints, Li et al. [24] proposed an adaptive attribute enhancement mechanism that enriches the original topological structure and integrates deep clustering into latent embedding learning, providing inherent constraints for the clustering structure. Leiber et al. [25] introduced the DipDECK algorithm, an innovative method capable of simultaneously estimating cluster quantity and improving clustering objectives based on deep learning. Li et al. [26] implemented a graph clustering, using a random walk algorithm, demonstrating its superior performance on multiple real-world datasets. In summary, the trend in deep clustering involves combining deep embedded representations with deep clustering constraints to enhance clustering performance. These methods leverage the powerful representation capabilities of neural networks, thereby improving the understanding and processing of complex data structures. However, the effectiveness of different methods varies with task characteristics; hence, in practical applications, it's essential to adjust and optimize the corresponding constraints according to specific task features to ensure optimal clustering results.

3 Methods

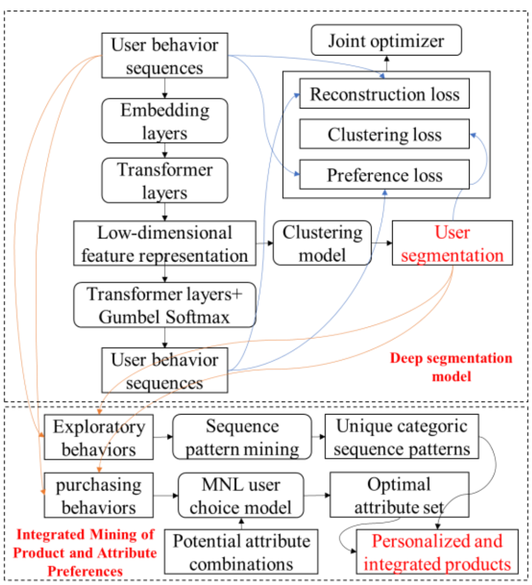

Figure 1: Overall framework.

The overarching framework of this study is designed to harness deep learning and data analysis techniques for uncover personalized, integrated demands. Figure 1 shows the overall framework. The framework is bifurcated into two interrelated core components: **1) Deep Customer Segmentation:** The objective of this segment is to understand and identify the unique behavioral patterns and preferences of different consumer groups. We utilize user behavior feature embedding, autoencoders, and clustering models, along with specially designed clustering loss functions for outlier removal. Deep customer segmentation not only lays the foundation for a profound understanding of consumer behavior but also establishes the crucial data groundwork for mining personalized, integrated demands. **2) Integrated Mining of Product and Attribute Preferences:** Building on the foundation of deep customer segmentation, we delve deeper into the integrated mining of category and attribute preferences. In this segment, we employ sequential pattern mining methods to analyze user browsing and carting behaviors, revealing the demands for integrated products. Concurrently, we utilize multinomial Logit models to analyze group purchase intentions, thereby accurately extracting integrated demands related to product attributes. Additionally, constructing a Multinomial Logit (MNL) model for each user can test whether the user is willing to purchase the products obtained.

3.1 Deep customer segmentation

The essence of Deep Customer Segmentation is to cluster user behavior sequences. "User behavior sequence" and "embedding of discrete features" aim to convert user actions into information that deep learning models can understand. "Encoding and decoding" are used to obtain a low-dimensional representation suitable for traditional clustering algorithms (usually, deep learning needs to be combined with traditional clustering algorithms to complete deep clustering). A clustering algorithm with an outlier removal mechanism can cluster the low-dimensional representations of user behavior sequences, thus dividing users into multiple groups to achieve customer segmentation. The details are as follows:

3.1.1 User behavior embedding. **User Behavior Features:** In our study, we utilized two key types of features to represent an item: the item ID and category ID. It is important to note that while an item might have hundreds of features, representing an item in the user behavior sequence with all features is prohibitively costly. Following the research by Chen et al. [5], item ID and category ID have been found sufficient to optimize the performance of deep embedding models. Hence, we selected these as sparse features to represent each item in the user behavior sequence. **User Behavior Sequence:** In this research, user behavior sequences were employed for an in-depth analysis and understanding of users' behavior patterns and preferences. As a structured form of data representation, the dimensions fed into the model for user behavior sequences are defined as [minibatch, behavior_len, feature_number]. Here, 'minibatch' represents the total number of users in the sample, 'behavior_len' indicates the number of actions considered in the sequence, and 'feature_number' refers to the number of key features included in each action record. These features might include the type of user action, item IDs related to the action, and the item's category ID. **Embedding of Discrete Features:** Handling the sparsity of discrete features in user behavior sequences poses a challenge, which we address through the use of deep Embedding layers. The primary function of the Embedding layer is to transform sparse discrete features in user behavior sequences into dense, low-dimensional vectors. In the data preprocessing phase, missing or invalid data points are marked as zero to effectively handle sparse data, ensuring model stability in the face of incomplete or missing user behavior information.

3.1.2 Auto-encoder and clustering model. **Sequence Feature Extractor:** In this study, we employed the multi-head attention mechanism. The multi-head attention allows the model to focus on different parts of the sequence simultaneously by performing multiple attention operations in parallel. The mathematical expression for this mechanism is

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\dots,\text{head}_h)W^0, \quad (1)$$

where each head_i, is computed as

$$\text{Attention}\left(QW_i^Q, KW_i^K, VW_i^K\right), \quad (2)$$

and the basic attention function is defined as

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V. \quad (3)$$

Additionally, to endow the model with the capability to perceive the position of elements in sequence data, we introduced position embedding. Position embedding is calculated using sine and cosine functions, where for position pos and dimension i, the position embedding $PE(pos, i)$ is defined as

$$PE(p, 2i) = sin(\frac{p}{10000^{\frac{2i}{d}}}), \quad (4)$$

$$PE(p, 2i+1) = cos(\frac{p}{10000^{\frac{2i}{d}}}). \quad (5)$$

Encoding and Decoding: The encoder compresses user behavior sequences into multi-dimensional feature vectors, preserving essential information. Then, the decoder reconstructs this data, generating probability distributions for different categories like user actions and item types. We ensure data fidelity throughout by applying reconstruction loss functions. Additionally, our model incorporates multi-head attention and position embedding. Multi-head attention allows simultaneous focus on various sequence parts, capturing detailed and contextual data insights. Position embedding adds precise positional context to each sequence element, enhancing the model's effectiveness in handling time-series data.

Clustering Algorithm: In our study, we used k-means++ algorithm with an outlier removal method for data clustering. K-means++ reduces randomness in centroid selection, improving clustering effectiveness. Outlier removal further enhances this by excluding points too far from centroids, thereby reducing noise and ensuring more distinct clustering outcomes.

3.1.3 Joint Optimization. **Clustering Loss Function:** In user segmentation, removing outliers is particularly crucial, especially in scenarios of reverse customization, where we focus on satisfying the personalized needs of a group of users with highly similar preferences. Eliminating outliers aids in ensuring the high performance of clustering results, thereby enabling us to more precisely identify and fulfill these closely-aligned user preferences, ensuring the quality and effectiveness of personalized services. To optimize the clustering process and minimize the impact of outliers, we employed a special clustering loss calculation method during the training of the autoencoder. This method considers the negative impact of outliers on clustering results and introduces a weighting factor to mitigate their influence. Let $\mathbf{x}_i$ be a point in the dataset, and $c_{\text{cluster}}(\mathbf{x}_i)$ be the nearest clustering center to $\mathbf{x}_i$. For each point $\mathbf{x}_i$, we calculate its distance to $c_{\text{cluster}}(\mathbf{x}_i)$, denoted as $d(\mathbf{x}_i, c_{\text{cluster}}(\mathbf{x}_i))$. If this distance exceeds a predetermined threshold, then $\mathbf{x}_i$ is marked as an outlier. The specific clustering loss formula is as follows:

$$L_{\text{cluster}} = \frac{1}{N - N_{\text{outlier}}} \sum_{i=1}^{N} w_i \cdot \|\boldsymbol{x}_i - c_{\text{cluster}}(\mathbf{x}_i)\|^2, \quad (6)$$

where N represents the total number of data points, N_{outlier} is the number of points identified as outliers, and the distance between each data point $\boldsymbol{x}_i$ and its nearest clustering center $c_{\text{cluster}}(\mathbf{x}_i)$ is calculated and squared. The weight factor w_i is used to adjust the impact of outliers, which can be set to zero for outliers and typically one for non-outliers. This method allows us to effectively reduce the interference of noise data while maintaining the quality of clustering. **Preference Loss Function:** The preference loss function is utilized during the training of the autoencoder to quantify and optimize the representation of preferences within user behavior patterns. The objective of this function is to minimize the differences between user behavior sequences in the feature space. Specifically, the preference loss can be calculated by comparing the feature representations of two sequences. The preference loss function is defined as:

$$L_{\text{preference}} = \sum_{i \in I} \sum_{j \in J} \sum_{u \in U} w_j \cdot \text{freq}_{i,j}(u)(\text{freq}_{i,j}(i) - \overline{\text{freq}}_{i,j}(u))^2, \quad (7)$$

where $\text{freq}_{i,j}(i)$ represents the proportion of interactions of user i with item u for behavior type j in the original sequence, $\overline{\text{freq}}_{i,j}(u)$ represents the proportion of interactions in the reconstructed sequence, and w_j is the weight of behavior type j. Sets I, J, and U represent users, behavior types, and items, respectively. By minimizing this preference loss, the model is guided to learn to map user behavior sequences with similar preferences to proximate points in the feature space, thereby effectively capturing and representing user preferences. **Joint Optimization Process:** The joint optimization process of the autoencoder emphasizes improving the model's performance in user segmentation and understanding user preferences. This process consists of two key stages. Initially, during the basic autoencoder training, the model learns to extract encoded representations and reconstruct outputs from each mini-batch of data, optimizing parameters through the computation of reconstruction loss. Then, in the joint training phase, in addition to continuing to optimize reconstruction loss, we introduce the calculation of clustering loss and preference loss. The joint loss function enables the model not only to reconstruct data but also to identify and analyze subtle differences within user groups. Algorithm 1 delineates the joint optimization process within the deep segmentation model, detailing how customer segmentation outcomes are derived.

Algorithm 1: Joint Optimization Process of Deep Customer Segmentation

Initialize: Autoencoder model, Optimizer
1) Pretrain autoencoder:
For each epoch, do:
  For each minibatch in the data, do:
    1. Compute encoded representation of the minibatch using the encoder
    2. Reconstruct output using the decoder
    3. Calculate Reconstruction Loss:
      - Compute reconstruction loss for user behavior, items, categories
    4. Update model parameters using backpropagation
2) Joint optimization autoencoder and clustering model:
For each epoch, do:
  For each minibatch in the data, do:
    1. Compute encoded representation of the minibatch using the encoder
    2. Reconstruct output using the decoder
    3. Calculate Reconstruction Loss
    4. Calculate Clustering Loss
    5. Calculate Preference Loss
    6. Compute Total Loss: Reconstruction Loss + α * Clustering Loss + β * Preference Loss
    7. Update autoencoder parameters using backpropagation
    8. Update parameters of the cluster model
3) Obtain customer segmentation
1. Input all user behavior sequences into the encoder and output the representations
2. Input the representations into the clustering model and output the clustering labels of each user behavior sequences
The labels are the deep customer segmentation results

3.2 Extracting integrated preferences

3.2.1 Integrated category demand mining based on sequential patterns. In past research, scholars have primarily focused on analyzing consumer preferences through online reviews [10, 14]. However, this approach has limitations: user comments are often influenced by personal emotions and specific experiences, leading to potential biases; moreover, only a minority of users leave reviews, so these data might not fully represent the behaviors and preferences of all users. To overcome these limitations and delve deeper into consumers' latent needs, this study shifts focus to analyzing users' browsing and carting behaviors. These behaviors, as a form of "silent" expression of user preferences, provide more raw and unfiltered data on user interests [13].

In this study, we employ sequence pattern mining methods to analyze the sequences of users' browsing and carting behaviors. We utilize the PrefixSpan algorithm [27], an efficient pattern-growth method for identifying frequently occurring subsequences in large datasets. Through this method, we can identify typical interaction patterns between users and various product categories. These patterns not only reveal users' direct interests but may also indicate a demand for new product combinations or service bundles, providing inspiration for businesses to create innovative integrated products. The unique sequence patterns for each cluster are the category sets that need to be integrated. The technical details are as follows:

Data Preprocessing: Preprocess the user behavior data by filtering for events of type "PageView" and "AddToCart" and extracting their key attributes. Let the original user behavior data be denoted as D, with each event denoted as e, where e_{prop} represents the key attributes of the event. The preprocessed data is denoted as D. The preprocessing can be represented as:

$$D' = \{e_{\text{prop}} \mid e \in D \wedge (e_{\text{type}} = \text{"View"} \vee e_{\text{type}} = \text{"Cart"})\}, \quad (8)$$

Minimum Support and PrefixSpan Algorithm: Combine the definition of minimum support (min_sup) with the use of the PrefixSpan algorithm to find frequent patterns. Let the volume of data for category c be denoted as n_c with a predefined ratio factor k to adjust the size of minimum support. For a sequence dataset S, the process of defining minimum support and finding frequent patterns involves identifying all sequence patterns P such that the occurrence of P in S meets the following condition:

$$\text{count}\,(P, S) \geq \frac{n_c}{k}. \quad (9)$$

The core implementation of the PrefixSpan algorithm can be briefly described by the following steps: 1) Scan the dataset S item by item. For each possible prefix, create a corresponding "projected database" containing the rest of the sequences that include that prefix. 2) Recursively perform the same process on each projected database until all frequent sequences that meet the minimum support threshold are found. This process can be simplified with pseudocode as Algorithm 2:

Algorithm 2

```
PrefixSpan(S, prefix, min_sup):
 for each item i in S:
  if count({prefix + i }, S) >= min_sup:
   emit({prefix + i })
   S_i = project(S,{prefix + i})
   PrefixSpan(S_i, {prefix + i }, min_sup)
```

where, ' $\text{project}(S, \{\text{prefix} + i\})$' is the process of generating a projected database based on the current prefix, and 'emit ((prefix $+i$}) ' indicates that a sequence pattern meeting the minimum support condition is recorded.

Identifying and Excluding Common Patterns: To identify unique patterns across different customer segments, first, identify the common patterns (CommonPatterns) across all segments, and then exclude these common patterns from the frequent patterns of each segment. Assume there are N categories, with the frequent pattern set for each category denoted as Pattern class $_i$. The formula for calculating common patterns is:

$$\text{CommonPatterns} = \bigcap_{i=1}^{N} \text{Patterns}_{\text{class }i}. \quad (10)$$

After excluding common patterns, the unique pattern set for each category ($UniquePatterns_{\text{class }i}$) is:

$$Unique\text{Patterns}_{\text{class }i} = \text{Patterns}_{\text{class }i} - \text{CommonPatterns}, \quad (11)$$

3.2.2 Integrated attribute demand mining based on consumer purchase intent. In this section, we explore a method for mining integrated product attribute demands based on consumer purchase intentions. This approach primarily focuses on analyzing purchasing behaviors to capture the actual choice characteristics of users. We employ an attribute selection method based on the MNL model, aimed at extracting attribute sets that best represent consumers' purchasing intentions from user behavior data. The first step of this method is to generate candidate attributes based on the frequency of attributes in purchased products. This step mainly focuses on examining the frequency of attributes chosen by consumers when purchasing products. However, it should be noted that not all these high-frequency attributes accurately reflect the true needs of users. Some attributes may appear frequently due to product characteristics, rather than being the focal point of active consumer attention.

Initially, the MNL model is used to estimate the impact of different attribute combinations on consumers' choice probabilities. The data set used for model fitting includes the products purchased by each consumer segment and their attributes, as well as the corresponding attributes of non-purchased products. The model fitting can be expressed by the following formula:

$$P(y = j) = \frac{e^{\beta' x_j}}{\sum_{k=1}^{J} e^{\beta' x_k}}, \quad (12)$$

where $P(y = j)$ is the probability of choosing product j, J is the total number of products, $\boldsymbol{x_j}$ is the attribute feature vector of product j, and β is the vector of parameters to be estimated. This formula links the consumer's choice probability with product attributes through a logistic regression model. After fitting the MNL model for attribute set selection, the next step is to determine the attribute set that maximizes the choice probability. This is achieved by iterating through all possible attribute combinations based on candidate attributes and calculating the choice probability for each combination. The attribute combination with the highest choice probability can be determined using the following formula:

$$\text{Maximized Choice Set} = \arg\max_j P(y = j), \quad (13)$$

where we calculate the choice probability for each attribute combination j and select the combination that maximizes this probability as the final choice. The specific execution process of Integrated Attribute Demand Mining can be found in Algorithm 3.

Algorithm 3

1. Initialize:
   - Collect user behavior data including products purchased and not purchased
   - Extract attributes of all products
2. Generate Candidate Attributes:
   - For each purchased product, record its attributes.
   - Count the frequency of each attribute across all purchases
   - Select attributes with high frequency as candidate attributes
3. Fit MNL Model:
   - For each consumer segment, prepare data:
     * Attributes of products purchased (positive examples)
     * Attributes of products not purchased (negative examples)
   - Define the MNL model
   - Estimate the model parameters (β) using the data
4. Determine Optimal Attribute Set:
   - For each combination of candidate attributes:
     * Calculate the choice probability P(y=j) using the fitted MNL model
   - Find the attribute set that maximizes the choice probability
5. Output:
   - Return the attribute set with the highest consumer purchase intent

4 Experimental results

4.1 Experimental setting

Dataset: 1) UserBehavior Dataset (UserBehavior): Provided by Alibaba. It covers about one million users from November 25 to December 3, 2017. Each data record includes user ID, product ID, product category ID, behavior type, and timestamp. The dataset contains approximately 4.16 million products. 2) JD Dataset (JDD): This dataset records data between February 1 and April 20, 2016. It includes about 105,000 users, around 29,000 products. 3) Multi-category Store E-commerce Behavior Dataset (MCD): Contains user behavior data of a large multi-category online store from October 2019 to April 2020, covering 285 million user events. This dataset includes about 160,000 products across 624 categories. Its data structure is also similar to the UserBehavior dataset.

Evaluation metrics: We utilized two clustering metrics to assess the effectiveness of user segmentation, indicating better targeting of the audience for personalized integrated demand mining. Additionally, we used a purchase intention metric to evaluate the effectiveness of personalized integrated demand mining. These metrics quantify the framework's ability to identify who to sell to and what to sell. 1) User Segmentation Silhouette Coefficient (SC): Its formula is: $s(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))}$, where, $a(i)$ is the average distance of sample i to other samples in the same segment, and $b(i)$ is the average distance to samples in the nearest neighboring segment. $s(i)$ ranges from -1 to 1, with higher values indicating better segmentation. 2) User Segmentation Calinski-Harabasz Index (CH): The formula is: $CH(k) = \frac{\mathrm{Tr}\,(B_k)/(k-1)}{\mathrm{Tr}\,(W_k)/(n-k)}$, where $\mathrm{Tr}\,(B_k)$ is the total dispersion between segments, $\mathrm{Tr}\,(W_k)$ is the total dispersion within segments, k is the number of segments, and n is the total number of samples. Higher scores indicate better segmentation. 3) Mean Purchase Probability (MPP): In the process of mining integrated product attribute sets, it's assumed that each user segment group has homogeneous preferences, meaning all users within the same segment share identical product attribute preferences. This assumption allows researchers to use a common MNL model to represent the attribute selection preferences of the entire segment group. This identifies universally applicable product attribute sets for the entire segment group. During the evaluation phase, an individual MNL model is fitted for each user, giving each user unique attribute selection parameters. The integrated product attribute set identified for each user's segment group is input into their respective MNL models to calculate the purchase probability for each user. The formula for calculating the average user purchase probability is: $\mathrm{MPP} = \frac{1}{N}\sum_{i=1}^{N} P_i\left(y=1 \mid X_{\text{segment}_i}\right)$, where $P_i\left(y=1 \mid X_{\text{segment}_i}\right)$ represents the probability of user i choosing the integrated product with attributes X_{segment_i}, and N is the total number of users.

Experimental Implementation: All our experiments were conducted in a Python 3 environment, using the TensorFlow framework. The computational resources included a quad-core Intel Core i7-7700 CPU (3.60 GHz clock speed) and an Nvidia GeForce RTX 2080 GPU. We set the batch size to 32 for the experiments. The initial learning rate was set to 0.001 with a polynomial decay strategy for adjustment. To prevent overfitting, a dropout strategy was applied to 20% of the parameters. In the experimental process, the autoencoder first underwent 10 epochs of pre-training, followed by 100 epochs of joint training. To ensure the reliability of the experimental results, we repeated the experiments 10 times and averaged the evaluation metrics. Key hyperparameters such as α, β, and the outlier threshold were set within the range of [0, 0.1, 0.5, 1]. The range for the length of user behavior sequences was set to [100, 200, 500]. As the datasets did not provide explicit product attributes, we used the multi-dimensional vector output from the Item Embedding layer processed through the Sigmoid activation function as a representation of discrete product attributes. The source codes related to this project can be found at the following link.

4.2 Performances of customer segmentation

Table 1: Comparison with Baseline Models

	UserBehavior			JDD			MCD		
Method	SC	CH	MPP	SC	CH	MPP	SC	CH	MPP
ECS	0.300	2603.407	0.339	0.354	4840.907	0.301	0.471	7034.541	0.486
MST	0.289	2585.724	0.395	0.383	5500.594	0.356	0.490	7051.984	0.402
CDRNN	0.264	2500.276	0.402	0.346	4609.418	0.301	0.545	7175.538	0.487
DC	0.257	2707.471	0.364	0.404	5289.170	0.336	0.598	7187.809	0.437
DEC	0.206	1840.411	0.331	0.333	3753.706	0.287	0.491	5282.242	0.325
Our model	**0.358 †**	**3214.131 †**	**0.455 †**	**0.485 †**	**6125.131***	**0.411 †**	**0.672 †**	**8125.131***	**0.571 †**

*Note: In the table, the highest score in each column is highlighted in bold. The symbols * and † indicate statistical significance in comparison to the best baseline model, corresponding to $p < 0.05$ and $p < 0.01$, respectively.*

4.2.1 Deep segmentation benchmarks. In our study, we compared a range of recent customer segmentation models, sequence clustering models, and classic deep clustering models as baseline models. To ensure fairness in comparison, we replace the proposed deep customer segmentation model with benchmark models, while keeping the method for extracting integrated preferences (as described in Section 3.2) unchanged. In this manner, only the first core component of the overall framework (the method described in Section 3.1) is replaced, leaving the second core component unchanged. This approach is used to assess whether our proposed deep customer segmentation model can enhance the overall performance. The specific baseline models are as follows: 1) Efficient Customer Segmentation (ECS) [28]: An unsupervised deep learning model based on Self-Organizing Maps (SOM) and an improved Social Spider Optimization algorithm. 2) Market Segmentation Trees (MST) [11]: A model combining tree models and user response models. 3) Clustering Deep Recurrent Neural Network (CDRNN) [29]: A deep clustering model focused on learning the relationships of sequence structures. 4) DeepClustering (DC) [30]: A classic deep clustering model using pseudo labels and initialized parameters guided by Gaussian distribution. 5) Unsupervised Deep Embedding for Clustering (DEC) [31]: A classic clustering model that learns feature representations and clustering assignments through deep neural networks, mapping data to a low-dimensional feature space and iteratively optimizing the clustering objective in this space. To maintain consistency and fairness in our comparative analysis, all benchmark models were configured with parameter settings and data processing techniques that align with our proposed framework.

4.2.2 Comparison results with deep segmentation Benchmarks. Table 1 presents a performance comparison between our proposed model and multiple baseline models. From Table 1, we can draw the following conclusions: 1) Significant Improvement in User Segmentation: Across the three datasets, our model significantly outperforms the existing baseline models in terms of the SC and CH indicators. This superiority is statistically significant, as confirmed by two-sided t-tests. This outcome highlights the robust capability of our model in user segmentation, demonstrating its effectiveness in capturing user behavior patterns. 2) Notable Advantages in Purchase Probability Prediction: In predicting users' purchase intentions, our model scored 0.455, 0.411, and 0.571 on the MPP metric for the 3 datasets, respectively. These scores are not only the highest across all datasets but also represent statistically significant improvements over the best baseline models, demonstrating the efficacy of our model in enhancing the purchase likelihood of integrated attributes.

4.2.2 Deep segmentation parametric analysis. The parameter α represents the weight of the clustering loss, while β represents the weight of the preference loss. The settings of these two parameters directly influence how the model balances different types of losses, thereby achieving optimal performance in user segmentation accuracy and integrated attribute purchase intention prediction. Therefore, we control all other parameters of the proposed framework to remain constant, and investigate the changes in algorithm performance as α and β vary. Figure 2 shows a comparison of different values of the hyperparameter α, which represents the weight of the clustering loss. The results indicate that the MPP reaches its optimal performance at α=0.1. This suggests that incorporating clustering loss enhances the users' purchase intention of integrated attributes. Clustering loss helps the model recognize and leverage the latent structure in the data, enabling better generalization. However, when α is increased to 0.5 or 1, although the SC and CH increase, the MPP generally decreases. This implies that an excessively high weight on the clustering loss can lead to an imbalance with other loss functions in the model. Too much emphasis on data clustering may compromise the model's learning of other important features, leading to a decline in performance. Therefore, selecting an appropriate weight for clustering loss is crucial.

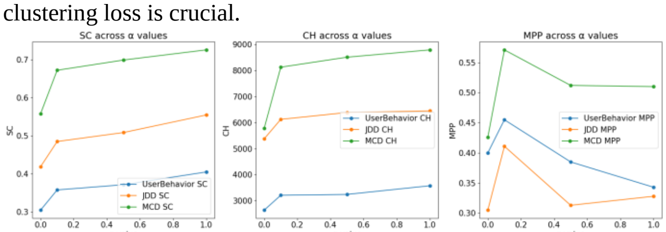

Figure 2: Comparison of Different Values of α

Figure 3 presents a comparison of different values of the β, which is the weight of the preference loss. The results show that the SC and CH metrics perform best at β=0.5 across all datasets. This weight strikes a fine balance in the model's loss functions, avoiding overfitting while not neglecting important structural features. Similarly, the MPP metric also peaks at β=0.5, suggesting that the model achieves the best overall performance at this weight. β set at 0.5 for preference loss likely facilitates an effective trade-off between clustering and preserving preference models, allowing the model to optimize one aspect without sacrificing the other. Overall, β as the weight parameter for preference loss at a setting of 0.5 seems to provide the model with the optimal level of regularization. This balance results in the model performing well across all metrics.

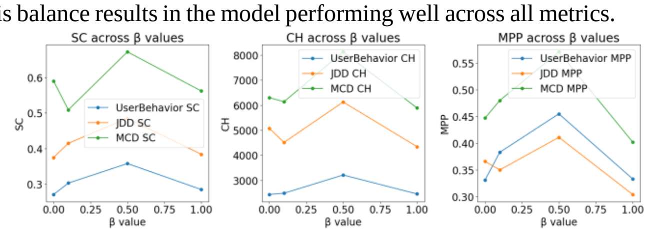

Figure 3: Comparison of Different Values of β

4.3 Results of integrated demands

4.3.1 Comparison of variants of integrated attribute mining framework. In this section, we primarily focus on identifying which Source Behaviors in Integrated Demand Mining yield the optimal User Purchase Intention. These behaviors include actions like "Browsing", " Carting", and "Purchasing". Our objective is to pinpoint the combination of source behaviors that maximizes the efficacy of Integrated Category Mining and Integrated Attribute Mining. These behaviors are either used individually or in combination as key drivers for mining integrated demands. Throughout the experimental process, all method settings remain unchanged, with only the data type applied to the method being altered.

Table 2: Source Behavior of Integrated Demand Mining and User Purchase Intention

Integrated category mining (3.3.1)	Integrated attribute mining (3.3.2)	MPP: UserBehavior	MPP: JDD	MPP: MCD
Browsing+Carting	Purchasing	0.455	0.411	0.571
Browsing	Purchasing	0.429	0.383	0.551
Carting	Purchasing	0.416	0.372	0.521

Table 2 showcases the three best-performing behavior combinations in terms of enhancing user purchase intention. Notably, we discovered that the combination of "Browsing + Adding to Cart" in Integrated Category Mining effectively identifies user interests, thereby meeting specific user needs. Meanwhile, in Integrated Attribute Mining, using "Purchasing" as a source behavior helps in identifying the attributes most strongly associated with the highest purchase intention. This combination has demonstrated optimal results across three different datasets.

These findings are not only consistent with the theoretical framework presented in Section 3.2 of this paper but also further validate the practical applicability of the Integrated Demand Mining framework we developed. The "Browsing + Adding to Cart" combination allows us to delve deeply into user interests and needs, while analyzing "Purchasing" behavior enables us to accurately identify which attributes most significantly influence user purchase decisions.

4.3.2 Examples of integrated demand results. This study utilized three datasets, among which only the MCD dataset provided visible category information. Consequently, we selected two case studies from this dataset to demonstrate how our methodology fulfills specific personalized integrated demands. The MCD dataset offers three-level category information, such as "furniture. living_room. sofa". We present analysis results from a dataset comprising three primary categories: furniture, appliances, and kids, with ten customer segmentation categories in total.

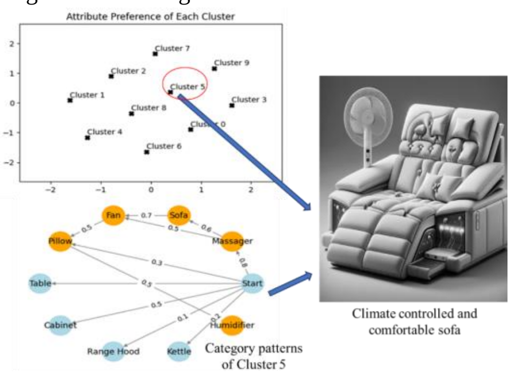

Figure 4: Integrated Demand Case Study 1

As shown in Figure 4, we analyzed the characteristics of the fifth customer segmentation group, which comprises about 12.07% of the total users. The lower left corner of Figure 4 details the user behavior sequence pattern for this group. In this pattern, nodes highlighted in orange indicate product categories with higher support, revealing these users' integrated demand for massagers, sofas, fans, pillows, and humidifiers. The upper left corner of Figure 4 displays the distribution of attribute preferences for each segmented group, processed using t-SNE technology (reducing multidimensional data to two dimensions). This distribution clearly shows the differences in attribute preferences among different segmented groups, with the marked points representing the attribute preferences of specific segments. Combining insights into integrated product demands and the attribute preferences of each segmented group, we can design products that cater to the personalized integrated needs of specific user groups. For instance, the right side of Figure 4 displays a conceptual sketch of an innovative product. This design integrates a massager and climate control features with a sofa and pillows. In the era of digitalization, such integrated product designs have become feasible. Although we lack precise background information about these target users, our methodology enables us to effectively meet the integrated and personalized needs of users with specific demands for environmental comfort.

Figure 5 analyzes the characteristics of the sixth user segment, which constitutes approximately 9.23% of the total user base. The lower left corner of the figure, highlighted with orange nodes, reveals product categories with high support, indicating this user group's integrated demand for climate control devices, fans, air conditioners, dolls, and children's diapers. The upper left corner of Figure 5 has the same meaning as the upper left corner of Figure 4. On the right side, Figure 5 displays a potential product combination for sale and an innovative concept illustration. This device integrates multiple functions such as humidification, ventilation, and temperature control, ensuring a comfortable and suitable living environment for children. Additionally, this device is conceptualized to be sold in combination with dolls and diapers, or to include them as complimentary gifts with the purchase, potentially enhancing its market appeal.

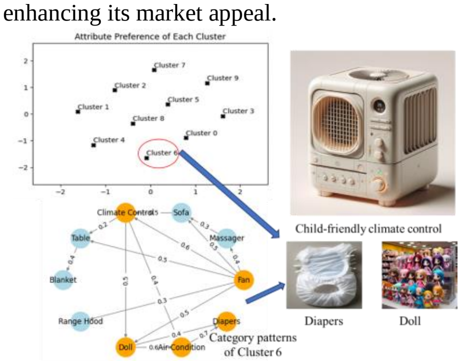

Figure 5: Integrated Demand Case Study 2

Overall, although it's challenging to directly discern the specifics of each product attribute set due to the difficulty in obtaining attribute data, our framework demonstrates how to employ deep analysis to identify user needs. We are confident that by designing integrated products with attributes that align well with the preferences of segmented users, the product innovations guided by our method will be highly attractive and significantly boost the interest and satisfaction of the target user groups.

4.4 Ablation experiments

In the ablation experiments, we made four different modifications to the proposed model to verify the impact of each component on model performance. These modifications include: 1) In the first framework (Framework 1), we removed the mechanism for eliminating outliers in the clustering process. 2) In the second framework (Framework 2), we removed the clustering loss function. 3) In the third framework (Framework 3), we removed the preference loss function. 4) In the fourth framework (Framework 4), we removed the mechanism for eliminating the shared sequence modes.

According to the results shown in Table 3, we can see that the model's performance decreases after removing any of the aforementioned components. This indicates that each component of the proposed model has a significant impact on the overall performance of the model, and the complete model (Our model) is statistically significantly superior to all frameworks after ablation of the components. This result validates the rationality and effectiveness of the proposed model design.

Table 3: Results of Ablation Experiments

	UserBehavior			JDD			MCD		
Method	SC	CH	MPP	SC	CH	MPP	SC	CH	MPP
Framework 1	0.338	3001.690	0.435	0.456	5871.047	0.394	0.622	7340.102	0.520
Framework 2	0.345	3017.011	0.435	0.448	5697.016	0.374	0.627	7641.190	0.545
Framework 3	0.332	2929.433	0.413	0.438	5627.948	0.382	0.647	7415.342	0.522
Framework 4	/	/	0.448	/	/	0.401	/	/	0.565
Our model	**0.358 †**	**3214.131 †**	**0.455 †**	**0.485 †**	**6125.131 †**	**0.411 †**	**0.672 †**	**8125.131***	**0.571 †**

*Note: In the table, the highest score in each column is highlighted in bold. The symbols * and † indicate statistical significance in comparison to the best baseline model, corresponding to $p < 0.05$ and $p < 0.01$, respectively.*

5 Discussions

Regarding the latest progress of this work, our research primarily concentrates on a joint forecasting model for pricing and sales volume of personalized integrated products. This model aims to provide a scientific basis for pricing integrated products by deeply analyzing the comprehensive attribute value of products and the interaction between price and sales volume. Specifically, the model targets utilizing purchasing data from users within specific market segments to analyze the interactions among product attributes, prices, and sales volumes, predicting the optimal price points and sales volumes based on a new set of product attributes. This framework includes three levels: firstly, the data layer, which collects information on product attributes, prices, and sales volumes from segmented groups, serving as the foundation for model learning; secondly, the multi-task feature extraction layer, which converts product attributes into feature vectors of functional and non-functional attributes (through embedding or one-hot encoding), and further extracts features for predicting price points and sales volumes; and finally, the optimization layer, which involves constructing an overall objective function aimed at minimizing the error between price and sales volumes while maximizing total profit. This model not only explores the direct connections between integrated product attributes and their prices and sales volumes but also reveals the complex dynamic relationship between price and sales volume, offering insights for maximizing profit through pricing of integrated products. Additionally, considering that the new products being developed have no precedents in the market and all training data come from existing products, there is a significant domain shift problem. Therefore, we are committed to building an unsupervised domain adaptation regression model to enhance the framework's performance, currently engaging in related experiments through techniques like adversarial training.

Regarding the JD.com's strategy for applying this work in practical scenarios is divided into two main aspects: 1) Collaboration with Enterprises for Product Innovation: JD.com provides comprehensive support to partner companies in product design and marketing strategies. This includes insights from this work on the target consumer groups, necessary product features and specific attributes, as well as pricing and recommendation strategies currently under research. Based on these recommendations, the partner companies will complete the specific development and production of the products. Subsequently, JD.com will provide testing (such as trial distribution, feedback collection, etc.) and sales channels for these products to achieve mutual profitability. 2) Independent Design, Development, and Production Organization: Through its own brand "JD Jingzao", JD.com independently manages the entire cycle from product design and development to production. With a vast supply chain network, JD Manufacture is capable of independently completing the entire process from design to production, ensuring a smooth transition of products from concept to end-users and maximizing brand value.

6 Conclusions

In e-commerce's C2M model and supply chain ecosystem context, this study aims to construct a novel framework to uncover personalized and integrated product demands. Specifically, we developed a deep segmentation method, integrating deep learning with user behavior analysis, to accurately divide consumers into groups with similar patterns and preferences. The process begins with collecting user behavior data, transformed into quality, compact vectors using autoencoder technology. We then apply custom loss functions for joint training with clustering algorithms, enhancing user segmentation and uncovering purchase intentions for integrated attributes. Our in-depth user behavior analysis through sequence mining and Logit models precisely identifies needs for specific product types. Results show our method excels in user segmentation accuracy and intent prediction across various datasets, validating our framework's innovation and practicality. This study marks a significant step in understanding and meeting e-commerce's personalized and integrated demands, offering a data-driven innovation approach in the C2M model and demonstrating the power of deep consumer demand analysis in supply chains.

Limitations and future development opportunities: 1) We couldn't access detailed product attribute data, especially unstructured data, limiting our analysis. Future research could employ natural language processing to handle unstructured data better, enhancing attribute mining. 2) Our research didn't examine the relationship between integrated product pricing and sales volume. Future work could analyze this dynamic to offer comprehensive market insights and pricing strategies. 3) While focusing on product and category-based integrated demands, we didn't fully explore the influence of supply chain structures on product innovation and customer needs. Future studies could consider these structures to optimize supply chain design for integrated product demands, potentially leading to new innovation insights.

REFERENCES

[1] Zhang, X., Ming, X., Liu, Z., Qu, Y., & Yin, D. (2019). State-of-the-art review of customer to business (C2B) model. *Computers & Industrial Engineering*, 132, 207–222. https://doi.org/10.1016/j.cie.2019.04.039

[2] Fan, X., Wang, S., & Wang, J. (2021). The value of introducing customer-to-manufacturer model by the online retailer. *International Transactions in Operational Research*, 29(4), 2566–2585. https://doi.org/10.1111/itor.13023

[3] Mak, H. Y., & Max Shen, Z. J. (2020). When triple-a supply chains meet digitalization: The case of JD.com's C2M model. *Production and Operations Management*, 30(3), 656–665. https://doi.org/10.1111/poms.13307

[4] Li, Y., Li, X., & Cai, J. (2021). How attachment affects user stickiness on live streaming platforms: A socio-technical approach perspective. *Journal of Retailing and Consumer Services*, 60, 102478.

[5] Yang, M., Zhang, T., & Wang, C. X. (2021). The optimal e-commerce sales mode selection and information sharing strategy under demand uncertainty. *Computers & Industrial Engineering*, 162, 107718.

[6] He, B., Mirchandani, P., & Yang, G. (2023). Offering custom products using a C2M model: Collaborating with an E-commerce platform. *International Journal of Production Economics*, 262, 108918.

[7] Orenstein, P. (2020). The changing landscape of Supply Chain Networks: An empirical analysis of topological structure. *INFOR: Information Systems and Operational Research*, 59(1), 53–73. https://doi.org/10.1080/03155986.2020.1785263

[8] Legenvre, H., Hameri, A. P., & Golini, R. (2022). Ecosystems and supply chains: How do they differ and relate. *Digital Business*, 2(2), 100029.

[9] Holmström, J., Holweg, M., Lawson, B., Pil, F. K., & Wagner, S. M. (2019). The digitalization of Operations and Supply Chain Management: Theoretical and methodological implications. *Journal of Operations Management*, 65(8), 728–734. https://doi.org/10.1002/joom.1073

[10] Timoshenko, A., & Hauser, J. R. (2019). Identifying customer demand from user-generated content. *Marketing Science*, 38(1), 1–20. https://doi.org/10.1287/mksc.2018.1123

[11] Aouad, A., Elmachtoub, A. N., Ferreira, K. J., & McNellis, R. (2023). Market segmentation trees. *Manufacturing & Service Operations Management*, 25(2), 648–667. https://doi.org/10.1287/msom.2023.1195

[12] Wang, J., Huang, P., Zhao, H., Zhang, Z., Zhao, B., & Lee, D. L. (2018). Billion-scale commodity embedding for e-commerce recommendation in Alibaba. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. https://doi.org/10.1145/3219819.3219869

[13] Chen, Q., Zhao, H., Li, W., Huang, P., & Ou, W. (2019). Behavior sequence transformer for e-commerce recommendation in Alibaba. *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*. https://doi.org/10.1145/3326937.3341261

[14] Fainmesser, I. P., Olié Lauga, D., & Ofek, E. (2021). Ratings, reviews, and the marketing of new products. *Management Science*, 67(11), 7023–7045. https://doi.org/10.1287/mnsc.2020.3848

[15] Han, S., & Mikhailova, D. (2024). Reducing the Bias in Online Reviews Using Propensity Score Adjustment. *Cornell Hospitality Quarterly*, 19389655231223364.

[16] Kerkhof, A., & Münster, J. (2019). Detecting coverage bias in user-generated content. *Journal of Media Economics*, 32(3-4), 99-130.

[17] Stevens, J. L., Spaid, B. I., Breazeale, M., & Jones, C. L. E. (2018). Timeliness, transparency, and trust: A framework for managing online customer complaints. *Business Horizons*, 61(3), 375-384.

[18] Yin, J., Liu, C., Wang, W., Sun, J., & Hoi, S. C. H. (2020). Learning transferrable parameters for long-tailed sequential user behavior modeling. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. https://doi.org/10.1145/3394486.3403078

[19] Yuan, F., He, X., Karatzoglou, A., & Zhang, L. (2020). Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. https://doi.org/10.1145/3397271.3401156

[20] Bian, S., Zhao, W. X., Zhou, K., Cai, J., He, Y., Yin, C., & Wen, J.-R. (2021). Contrastive Curriculum Learning for Sequential User Behavior Modeling via data augmentation. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. https://doi.org/10.1145/3459637.3481905

[21] Baltescu, P., Chen, H., Pancha, N., Zhai, A., Leskovec, J., & Rosenberg, C. (2022). ItemSage: Learning product embeddings for shopping recommendations at Pinterest. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/3534678.3539170

[22] Leiber, C., Bauer, L. G., Schelling, B., Böhm, C., & Plant, C. (2021). DIP-based deep embedded clustering with K-estimation. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. https://doi.org/10.1145/3447548.3467316

[23] Wang, X., Jing, L., Liu, H., & Yu, J. (2023). Structure-driven representation learning for deep clustering. *ACM Transactions on Knowledge Discovery from Data*, 18(1), 1–25. https://doi.org/10.1145/3623400

[24] Li, Y., Fu, Y., Su, H., & Li, J. (2021). Adaptively attribute-enhanced graph embedding via deep clustering constraints. *2021 10th International Conference on Computing and Pattern Recognition*. https://doi.org/10.1145/3497623.3497633

[25] Leiber, C., Miklautz, L., Plant, C., & Böhm, C. (2023). Application of deep clustering algorithms. *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. https://doi.org/10.1145/3583780.3615290

[26] Li, X., Li, D., Jin, R., Ramnath, R., & Agrawal, G. (2022). Deep graph clustering with random-walk based scalable learning. 2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). https://doi.org/10.1109/asonam55673.2022.10068646

[27] Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., ... & Hsu, M. C. (2004). Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on knowledge and data engineering*, 16(11), 1424-1440.

[28] Wang, C. (2022). Efficient Customer Segmentation in digital marketing using Deep Learning with swarm intelligence approach. *Information Processing & Management*, 59(6), 103085. https://doi.org/10.1016/j.ipm.2022.103085

[29] Zhong, W., & Gu, F. (2022). Predicting local protein 3D structures using clustering deep recurrent neural network. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 19(1), 593–604. https://doi.org/10.1109/tcbb.2020.3005972

[30] Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 132-149).

[31] Xie, J., Girshick, R., & Farhadi, A. (2016, June). Unsupervised deep embedding for clustering analysis. In *International conference on machine learning* (pp. 478-487). PMLR.