International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 38, No. 4, August, 2024, pp. 1235-1242
Journal homepage: http://iieta.org/journals/ria

# PRMNBR: Personalized Recommendation Model for Next Basket Recommendation Using User's Long-Term Preference, Short-Term Preference, and Repetition Behaviour

Kaushlendra Kumar Sinha*, Somaraju Suvvari

Department of CSE, NIT Patna, Patna 800005, India

Corresponding Author Email: kaushlendrakumars.phd18.cs@nitp.ac.in



https://doi.org/10.18280/ria.380417

**Received:** 27 October 2023
**Revised:** 12 December 2023
**Accepted:** 15 January 2024
**Available online:** 23 August 2024

***Keywords:***
*recommendation system (RS), Next Basket Recommendation (NBR), correlation matrix, repetition aware basket, Correlation Sensitive Basket, Long Short-Term Memory (LSTM)*

**ABSTRACT**

Next Basket Recommendation (NBR) tries to recommend items in a user's coming basket by understanding the user's characteristics from the past baskets of the user. The existing deep learning models for recommendation system (RS) are formulated by combining the long-term and short-term preferences of the user successfully. Recent statistical-based models highlight the importance of repeat purchase behavior, especially in the E-commerce industry, as most customer repeatedly purchases items. Including repeat behaviour dynamics can lead to a certain degree of improvement in the deep learning-based NBR models, as shown in a few recent statistical-based works. In this paper, we introduced a mechanism to extract the user's repetition behaviour along with the user's long-term preferences and short-term preferences. To capture the repetition behavior of the user, we introduced the encoded user's baskets as Repeat Aware Baskets, and to extract the correlation between items, we used a Correlation Sensitive Basket. Further, separate embedding is generated with respect to Repeat Aware and Correlation Sensitive Baskets. These embedding are fed parallel to two layered Long-Short Term Memory architecture for analyzing short-term preference. To evaluate the performance of the proposed model, we experimented on two data sets. Our proposed algorithm outperformed various recently developed models over various performance metrics.

## 1. INTRODUCTION

Due to the rapid advancement of technology, most industries now offer a wide range of substitute services. For example, the quantity of goods offered by e-commerce companies like Flipkart, Amazon, Big Basket, and Myntra, as well as the quantity of films and music albums provided by Netflix and other services like Pandora, etc. The difficulty for the customer is in selecting the service that best meets their needs from among the many accessible possibilities. The recommendation system seems to be the most viable solution in this scenario since it has already been used in different contexts.

Recommendation algorithms can be classified as follows: General recommendation [1], Sequential recommendation [2-4], Top-n recommendation, Next Basket Recommendation (NBR) [5-7], and, session-based recommendation [8-10]. Items are recommended by the general recommendation algorithms according to the user's tastes, both short- and long-term. Sequential recommendation algorithms are used in current recommendation literature to extract the user's short-term preferences. In the course of this process, the chronological sequence of the users' buying behavior is taken into consideration. Algorithms that provide recommendations based on a user's top-n interests offer suggestions for the top-n products and services. The session-based prediction could include suggestions for the following session or for the remaining portion of the present session.

The next basket (here the basket refers the collection of items purchased in the same time point) suggestion makes use of the target user's previous historical baskets to make predictions about the things that will be included in the future basket. NBR adheres to the question of what a particular customer is going to buy in their next transaction, given that its historical transaction pattern is known. Giving suitable predictions is a necessary aspect in retail-based services that have a large number of items in their corpus and each of their customer interacts with a few items in their single transactions. It is not feasible for a customer to browse through the entire corpus in a single visit. Here, NBRs come into play, lessening this burden from the shoulders of companies and customers.

The Next Basket Recommendation algorithms solve the task of determining what the users would engage with in their subsequent interactions. As illustrated in Figure 1, NBR models the past transactions of the different users included in the data set in order to anticipate the things that will be included in a future transaction. NBR is used in many businesses; the retail industry makes recommendations for their clients. An in-depth analysis of how user preferences and item desirability evolve over time can be carried out with NBR. Significant progress has been made in NBR in recent years, and a number of models utilizing deep learning, matrix factorization, and pattern mining techniques have been published in recent research.

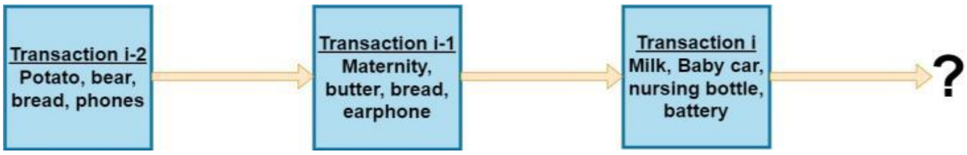



**Figure 1.** Next Basket Recommendation

The collaborative filtering approach was first applied as FPMC [11], which proved to be one of the most successful models during the initial phases of the NBR. They employed a method that integrated the Markov chain and Matrix Factorization techniques to ascertain the user's sequential and long-term characteristics. The score that FPMC gives the recommended things is the outcome of a linear combination of the user's general nature strength and their sequential nature strength. They were only able to assess the user's sequential nature in two successive baskets; they were unable to determine the nonlinear relationship between the users and the objects. The FPMC's main flaw is this, along with the fact that it only looks at Markov chains of length one. As proposed in the study [12], HRM emulates the interactions of neighboring baskets. Nevertheless, HRM uses the pooling process to non-linearly mix the vectors (interactions between users/items). The local sequential behaviour is also extracted using HRM, similar to FPMC. These models utilized a simple pattern mining approach and formed one of the basic works in this field. To address collaborative filtering, HRM introduced the concept of dimensionality in their work. Every user and item are represented by a vector of fixed dimension which forms the basis of recent deep-learning methods.

While basket embedding is also employed to enhance the model's performance, item embedding is the method used in the great majority of models. Basket embedding may ascertain the user's intention during their purchasing experience, whilst item embedding takes into account how similar an item is to another. Numerous studies attempted to improve the model's performance by utilizing different basket encoding techniques. Basket embedding is generated using the average or maximum pooling method. The earlier methods only employed item embedding. DREAM, which was first shown by the study [13], is one of the early deep-learning techniques. It demonstrates that max-pooling outperforms average pooling using basket embedding. DREAM analyses the embedded baskets in the chronological order that they occurred using Recurrent Neural Networks (RNNs), which depict baskets next to one another and allow interaction with the union level (a sequence of two or more baskets effects the target basket). The primary limitation of DREAM is its inability to ascertain an individual's enduring inclinations. To overcome the limitations of FPMC, Tang and Wang [14] presented a model called CASER, which is based on convolutional neural networks. They use a one-dimensional convolutional neural network in order to achieve a satisfactory equilibrium between their long-term and short-term desires. On the other hand, CASER cannot model intricate relationships or long-term dependencies. CosRec, which was given [15], is a model that employs a 2D convolution model for the sequential recommendation; however, CosRec is unable to extract the union-skip-level. In order to accomplish the union skip-level behaviour in the sequential suggestion, recent recommendation models leverage the attention mechanism. Unfortunately, because the attention mechanism includes too many variables, it makes the model much more complex. Intention2Basket, developed by Wang et al. [16], uses the attention process to ascertain what individuals want to accomplish with each contact. Both inter-basket and intra-basket attention mechanisms are used throughout the extraction process. Inter-basket attention is utilized to determine the overall suggestion, and intra-basket attention is used to determine the user’s sequential behavior. The performance is improved, but again it comes with a complicated attention-based process, which makes the system unscalable for a large number of users. When it came to capturing the sequential suggestion, Intention2Basket made use of LSTM, and when extracting the general recommendation, it made use of a correlation matrix with a predetermined order. The primary objective is to determine what should go into the next basket; however, to provide useful ideas, one must also consider how the objects in each basket are related to one another. A correlation matrix is used to investigate the dynamics that are dependent on correlations. The importance of repetition is highlighted by the MPIF model created in the study [17]. It provides data regarding the overall repetition information of various data sets and skilfully applies this understanding to offer a recommendation that is suitable. This statistical model goes on to demonstrate how various deep learning algorithms fall short in capturing the dynamics that rely on recurrence. The findings of this statistical investigation make it abundantly evident that while the function of collaborative filtering is likely to be roughly the same across all data sets, the role of repetition-based dynamics is likely to be distinctive across all data sets. The model [18] is a recently designed neural networks-based model that provides a simple mechanism to capture repetition dynamics. However, that may not be sufficient to capture complex patterns. The repetition ratio is the proportion of products in a given data set that were repurchased by the same customer more than once.

So, we can say that earlier collaborative filtering-based approaches used pattern mining [19] and matrix factorization [20] to achieve collaborative filtering by representing the user and item using a similar latent dimension. Later, researchers moved to deep learning-based models to learn the more accurate latent representations of users. Earlier pattern mining-based approaches can’t process longer sequences. This problem is addressed using RNN and LSTM-based approaches. Further, an attention-based approach solves this problem but adds to the complexity. Nowadays, the model tries to combine correlation information of users and items, repetitive information of items, or any other user and item-based dynamics with a deep learning model, but this has not been fully achieved till now.

In this paper, we are proposing a model, “PRMNBR: Personalized Recommendation Model for Next Basket Recommendation using user’s Long-term Preference, Short-term Preference, and Repetition Behaviour.” In this model, 1) we introduce a mechanism to study repetition-based dynamics missing so far in deep learning-based models. 2) This model also uses a correlation matrix [21] to extract the general recommendations by correlating all the items purchased together. 3) Basket embedding is generated using a binary-encoding of the basket representation. 4) While correlation-sensitive baskets examine the dynamics of linked items bought together, repeat-aware baskets assist in analyzing consumers' recurrent conduct. A prefixed repeat ratio (< 1) is utilized for the repeat-aware basket, as seen in the study [17], which reduces the weight attached to prior baskets and increases preference for the most recent basket. 5) Embedding generated from these baskets is fed to the parallel LSTM architecture, as shown in Figure 2a. Sequential patterns with respect to both repetition and correlation patterns are analyzed using parallel LSTM. The sum of hidden representations is fed to LSTM at the second layer.

## 2. LITERATURE REVIEW

CASER [14] captures sequential characteristics by using previous L items' embedding. These layered embedding are transmitted separately to horizontal and vertical convolutional layers. The output of these layers is concatenated and transferred to a fully connected layer for high-level feature abstraction. It is possible to analyse broad user-related patterns by concatenating vectors that indicate user embedding. Two horizontal filters of dimension h*d are applied to the layered item embedding of size L*d in order to capture sequential features. Horizontal filters work with h items in a consecutive manner. The "union-level" interaction of basket sequences is captured using 1-D filters with different heights. "Point level interaction" is recorded using vertical filters. The fully connected layer receives the output from the two convolution layers. The outcome in this case is for sequential dynamics. Concatenating this output with user embedding allows the incorporation of broad pattern dynamics. During model training, modifications are performed in order to capture skip behaviors. Items that have a target cause other items to take the place of the original item. CASER [14] uses a latent factorization model for general data analysis and 1-D CNN for sequential analysis. Once interactions at the Union level can be computed using this. Another deep learning model called "Correlation-sensitive Next Basket Recommendation" (Beacon) [20] uses a correlation matrix for general recommendations and an RNN for sequential recommendations. More connected basket embeddings could result from adding item-item correlation to basket vectors. When encoding the basket, the importance of the elements and the association between each item pair are taken into consideration. It makes use of LSTM architecture to infer sequential associations along the basket sequence. To create basket vectors, M2pht [22], a comparable model, employs average pooling over item vectors. In order to analyse sequences, the gated recurrent unit (GRU) is also applied to these basket vectors. Users' overall preferences are popularity scores assigned to each item depending on how common it is in their shopping baskets; that is, a higher probability is assigned to an item that a user purchases on a regular basis. Additionally, it makes use of each user's transaction pattern with every item. It is the total of each user's individual basket's binary encoding in a geometric progression. Thus, the least preferred basket is the oldest. With an appropriate weight matrix, these baskets are transformed into vectors with the appropriate dimensions. To obtain the probability of each item, these vectors are concatenated to the last hidden dimension of the sequential pattern. Now, this probability and the users' general dynamics are used to generate the final probability. Appropriate forecasts are made using these probabilities.

A CASER improved version presented in the study [23] considers the sequential features that are linked to each item. In order to accomplish this, a vector of dimension four is concatenated, incorporating additional information such as the purchase time. By taking account of the item's pattern of repetition, this improves the prediction quality even further. For the NBR, this work [17] is a data mining-based strategy. The popularity of the item was used to determine how frequently a buyer bought an item. Their research makes clear how important the item's support (Personalized Item Frequency, or PIF) is. PIF has not received much attention in the current NBR models. The body of research on recommendation systems demonstrates how well the sequential character of the user was recovered by RNN, LSTM, or GRU-based deep learning techniques. Nevertheless, because they lack any dynamics to connect repetitive behaviour with, they are unable to recognize the significance of the PIF. This work [24] proposed a session-based model that makes use of the Repeat Link Effect. The two components of this method are the LSTransformer and the repetitive weighted graph neural network (RGWNN). The first module learns how items are represented in the session graph, and the second module obtains the user's general and sequential preferences. To extract the item-item interactions in the sequence, the weighted Graph Interest Network, or WGIN, was presented. In order to thoroughly assess user preferences, RWGNN pays close attention to the item-item interactions between frequently used items in a session. The repeating behaviour of each item is still not extracted by this model.

In the study [9], authors build a privacy-focused recommendation system based on block-chain technology. The approach involves using the NBR method to present recommendations based on a user's purchasing history and incorporating privacy-protecting data deletion procedures and context-based distributed processing. Recently, many countries have fixed laws regarding the option of data deletion as chosen by the customer. The aim is to develop a decentralized recommendation system that takes into account the preferences of each user and offers the best recommendation. In the study [9], authors use a collaborative filtering approach to find similar users. It uses cosine similarity to find similar users and a k-means algorithm for clustering. Now, few recommendations are generated suitable for every cluster of users. Users belonging to the same cluster are recommended the same items. The recommendation system in this approach uses a simple machine learning approach, but novelty is found in the application of block-chain used for security purposes.

ReCANet, a recently developed model, includes a mechanism to observe repetition dynamics [17]. For every item purchased by a user, it uses a history vector that stores the consecutive difference in transactions for that item. It uses separate user and items embedding. User and item embedding are concatenated and pre-processed with some trainable parameters to generate user item embedding. Next, these user item embedding is concatenated by a history vector to analyse repetition patterns for that item. This input is fed to two layers of LSTM to generate sequential features pertaining to that user and item. Two layered feed forward neural network is used followed by relu activation function. Sigmoid operation is used to generate the probability of occurrence of that particular item in the next transaction.

RDNBR [25] uses repetition behavior similar to this proposed model but this doesn't process repetition and correlation separately.

Most of the deep learning models try to come up with different architectures to extract and combine both short-term and long-term preferences, but none of the deep learning methods combined the short-term, long-term, and user repetition behavior.

## 3. PROBLEM FORMULATION

Assume that $U = \{U_1, U_2, ...., U_n\}$ is a set of $n$ users, and that $I = \{I_1, I_2, ..., I_m\}$ represents a sequence of past interactions with $m$ things. Every past transaction that is referred to as a "basket" is made up of numerous things that belongs to $I$. Predicting the products that users will purchase on their subsequent visit is the work's goal. The historical data for each user can be represented as $S_{Ui}$, where i stands for the unique value assigned to that user. As a result, user $U_i$'s baskets can be expressed as follows: $S_{Ui} = \{B_{i1}, B_{i2}, ..., B_{iT}\}$, where each basket holds a collection of objects that belong to $I$. $B_{iT}$ serves as our testing data and is a representation of the last user contact. Since each user will have a different amount of transactions, $T$ varies for each user. The remaining baskets are for sequential instruction, with the final basket serving as the truth basket. To compare this suggested model with other models, a range of performance criteria will be employed.

## 4. PROPOSED MODEL

**Table 1.** Overview of primary notation

| Abbreviation | Description |
|---|---|
| $C$ | Correlation matrix. |
| $\gamma$ | Pre-defined Repetition Ratio. |
| $x_{i,j}$ | Binary encoding of $j^{th}$ basket of $i^{th}$ user. |
| $CSB_{i,j}$ | Correlation Sensitive Basket calculated using C and $x_{i,j}$. |
| $RAB_{i,j}$ | Relation Aware Basket calculated using $\gamma$ and $x_{i,j}$. |
| $\text{Correlation\_emb}_{i,j}$ | Formed by $CSB_{i,j}$. |
| $\text{Repeat\_emb}_{i,j}$ | Formed by $RAB_{i,j}$. |
| $h\_correlation_{i,j}$ | Hidden representation of first layer of LSTM to process repetition dynamics. |
| $h_{i,j}$ | Hidden representation of second layer of LSTM. |
| $B_{l(s)}$ | Predicted items for every user. |
| $V$ | Negative items from $B_{l(s)}$. |
| $\alpha$ | Pre-defined value. |

Three parameters are mostly used in this work to improve predictions. i) A correlation matrix for general recommendations. ii) Repeat aware basket for analyzing different users' recurring purchase patterns. iii) Sequence analysis using LSTM. The application of correlation and repeat ratio is shown in Figure 2b Strong correlations between the objects in the basket can be inferred from a correlation-sensitive basket. Binary encoded baskets $x_i$ and correlation matrix are used to generate Correlation Sensitive Basket ($CSB_{i,j}$). Binary encoded baskets and repeat ratio are used to generate repeat aware basket $RAB_{i,j}$. $CSB_{i,j}$ and $RAB_{i,j}$ is used to generate $\text{Correlation\_emb}_{i,j}$ and $\text{Repeat\_emb}_{i,j}$ of fixed dimension. These embedding of fixed dimensions are fed to parallel LSTMs in layer1 as shown in Figure 2b. Hidden representations of these parallel LSTMs are added and fed to second layer of LSTM. Summary of various symbols used is mentioned in Table 1.

### 4.1 Correlation Sensitive Basket ($CSB_{i,j}$)

The correlation matrix is a square matrix calculated over the entire data set. It gives an idea of how frequently an item pair interacts with each other. Correlation Sensitive Basket is generated using a correlation matrix and binary encoding ($x_{i,j}$) of every basket as shown in Eq. (1). Firstly, the hadamard multiplication of binary encoding ($x_{i,j}$) of every basket and a trainable weight matrix w of size I is done. Now, the correlation matrix ($C$) is dot product with $x_{i,j}$ and a smaller value is reduced for denoising. The relu function is applied to make negative values zero thus removing unimportant correlation. To compensate for noise cancellation, β is reduced. Smaller values are so reduced to zero. Eq. (1) denotes a vector of ones with size I.

$$CSB_{i,j} = x_{i,j} \odot w + \text{Relu}(x_{i,j} \cdot C - \beta 1) \quad (1)$$

### 4.2 Repeat Aware Basket ($RAB_{i,j}$)

Binary encoding is used to create the repeat aware basket ($RAB_{i,j}$) representation, as shown in Eq. (2). Given the state-of-the-art sequential recommendation systems, it seems to reason that more recent transactions should have higher values than older ones. This is accomplished by using a preset repeat ratio, γ, with values ranging from 0 to 1. When baskets are multiplied by a greater power of repeat ratio, the most recent baskets are given more weight than the earlier ones. A predetermined value unique to a data collection is the repeat ratio. Thus, the repetition dynamics up to that basket are captured by $RAB_{i,j}$.

$$RAB_{i,j} = x_{i,1} * \gamma^{j} + x_{i,2} * \gamma^{j-1} + \ldots + x_{i,j} * \gamma^{1} \quad (2)$$

### 4.3 Basket Embedding

Embeddings of pre-fixed dimensions d are created from $CSB_{i,j}$ and $RAB_{i,j}$ as shown. These embeddings are fed to the LSTM architecture of the model as shown in Figure 2.

$$\text{Correlation\_emb}_{i,j} = \text{Relu}(CSB_{i,j} \cdot \theta + \phi) \quad (3)$$

$$\text{Repeat\_emb}_{i,j} = \text{Relu}(RAB_{i,j} \cdot \theta + \phi) \quad (4)$$

### 4.4 Basket Sequence Encoder

Sequence encoding uses two-layered LSTM to study the personalized sequential dynamics of a particular user. Hidden signals propagation between first layer parallel LSTMs are shown in Eq. (5) and Eq. (6).

$$\text{h\_correlation}_{i,j} = \text{Correlation\_emb}_{i,j} \cdot T + h_{i,j-1} \cdot T_1 + T_2 \quad (5)$$

$$\text{h\_repeat}_{i,j} = \text{Repeat\_emb}_{i,j} \cdot T + h_{i,j-1} \cdot T_1 + T_2 \quad (6)$$

In the second layer of LSTM summation of $\text{h\_correlation}_{i,j}$ and $\text{h\_repeat}_{i,j}$ is used as input. Hidden signal from the last cell is used is used to make score predictions.

$$h_{i,j} = (\text{h\_correlation}_{i,j} + \text{h\_repeat}_{i,j}) \cdot T + h_{i,j-1} \cdot T_1 + T_2 \quad (7)$$

$T$, $T_1$, $T_2$ are random weights of dimension $d_1 * d_2$, $d_2 * d_2$, and $d_2 * d_1$ respectively and output h have dimension $d_1$.

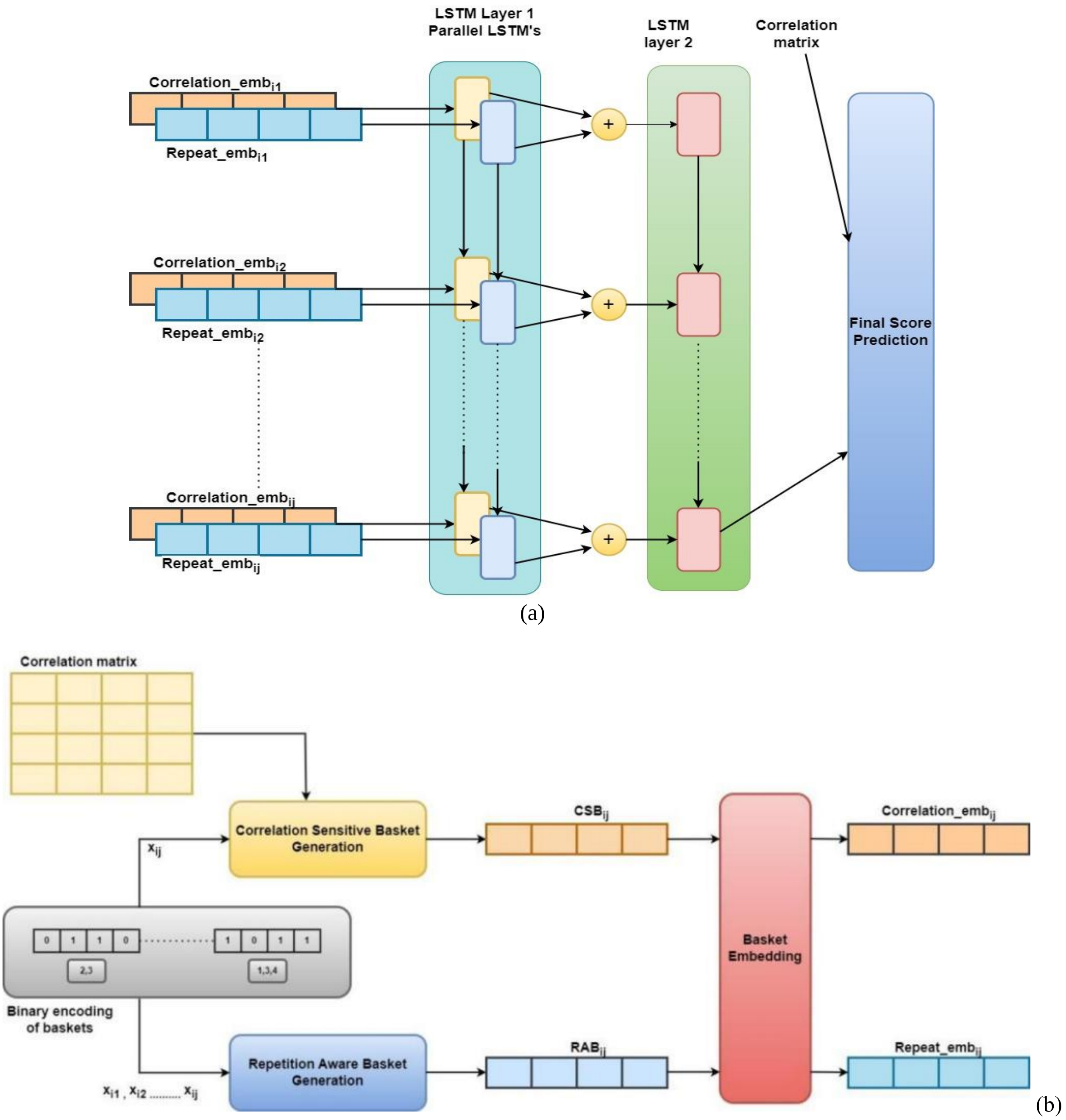



**Figure 2.** Architecture of proposed model. (a) shows the LSTM model to study sequential dynamics (b) shows detailed steps in calculation of Correlation-based embedding and repetition-based embedding.

### 4.5 Score prediction

Final score prediction is done by combing both sequential and general analysis. Sequential signal $s$ is generated using hidden representation $h$ obtained after processing last sequence $S_{Ui}$ of the user. Eq. (8) displays the formula for the sequential signal that is produced after each basket in the user's training data has been processed.

$$s = \sigma( h \cdot T_3 ) \tag{8}$$

This produces a probability for each item, as Eq. (9) illustrates. High-scoring items are anticipated for the following basket. Size $d_1$*I trainable weight matrix $T_3$.

$$score = \alpha * (s \odot w + s \cdot C) + (1 - \alpha) * s \tag{9}$$

In Eq. (9) first part denoted as $(s \odot w + s \cdot C)$ represents general analysis and $s$ represents sequential analysis signal. α ϵ (0,1) is a hyper-parameter deciding weight-age of sequential and general recommendation for that data.

## 5. OPTIMIZATION

For optimization purpose last basket is put into testing and predicted score is used to weight training purpose. Negative items are penalized to favor selecting correct items.

$$\mathrm{L(S)} = \frac{1}{||\mathrm{B}_{\mathrm{I(s)}}||} * \sum_{\mathrm{i}\,\epsilon\,\mathrm{B}_{\mathrm{I(s)}}} (\log(\sigma(score_i))) \\ \frac{1}{||\mathrm{V}/\mathrm{B}_{\mathrm{I(s)}}||} * \sum_{\mathrm{i}\,\epsilon\,\mathrm{V}/\mathrm{B}_{\mathrm{I(s)}}} (\log(\sigma(score_j - score_m))) \tag{10}$$

**Table 2.** Comparative analysis of models over *Tafeng* data set

| Tafeng | | Recall | Precision | F1score | PHR | MRR | MAP |
|---|---|---|---|---|---|---|---|
| | Caser | 0.1320 | 0.0763 | 0.0967 | 0.4750 | 0.1488 | 0.0691 |
| | CosRec | 0.0183 | 0.1071 | 0.0311 | 0.1071 | 0.0164 | 0.0812 |
| K=15 | RecaNet | 0.1558 | 0.0531 | 0.0676 | 0.4708 | 0.2361 | 0.1852 |
| | **PRMNBR** | **0.1694** | **0.1735** | **0.1743** | **0.5884** | **0.3297** | **0.1922** |
| | Caser | 0.1717 | 0.0664 | 0.0957 | 0.5453 | 0.1542 | 0.0691 |
| | CosRec | 0.0811 | 0.0916 | 0.0860 | 0.3497 | 0.0853 | 0.0812 |
| K=20 | RecaNet | 0.1684 | 0.0445 | 0.0611 | 0.4978 | 0.2376 | 0.1800 |
| | **PRMNBR** | **0.1770** | **0.1820** | **0.1720** | **0.6146** | **0.3589** | **0.3073** |
| | Caser | **0.2063** | 0.0593 | 0.0921 | 0.5958 | 0.1571 | 0.0691 |
| | CosRec | 0.1381 | 0.0773 | 0.0991 | 0.4853 | 0.1363 | 0.0812 |
| K=30 | RecaNet | 0.1841 | 0.0339 | 0.0509 | 0.5249 | 0.2387 | 0.1739 |
| | **PRMNBR** | 0.1937 | **0.2016** | **0.1975** | **0.6222** | **0.3949** | **0.3266** |

**Table 3.** Comparative analysis of models over *Dunnhumby* data set

| Dunnhumby | | Recall | Precision | F1score | PHR | MRR | MAP |
|---|---|---|---|---|---|---|---|
| | Caser | 0.0776 | 0.0459 | 0.0577 | 0.3397 | 0.1308 | 0.0348 |
| | CosRec | 0.0865 | 0.0354 | 0.0503 | 0.3744 | 0.1094 | 0.0375 |
| K=15 | RecaNet | 0.0668 | 0.0443 | 0.0408 | 0.3700 | 0.1910 | 0.1471 |
| | **PRMNBR** | **0.1107** | **0.1291** | **0.1191** | **0.4324** | **0.2563** | **0.2073** |
| | Caser | 0.0878 | 0.0361 | 0.0511 | 0.3722 | 0.1326 | 0.0348 |
| | CosRec | 0.0975 | 0.0305 | 0.0464 | 0.4016 | 0.1120 | 0.0375 |
| K=20 | RecaNet | 0.0720 | 0.0372 | 0.0383 | 0.3866 | 0.1909 | 0.1401 |
| | **PRMNBR** | **0.1234** | **0.1322** | **0.1276** | **0.4481** | **0.2831** | **0.2231** |
| | Caser | 0.0979 | 0.0304 | 0.0577 | 0.3989 | 0.1343 | 0.0348 |
| | CosRec | 0.1141 | 0.0244 | 0.0464 | 0.4338 | 0.1135 | 0.0375 |
| K=30 | RecaNet | 0.0775 | 0.0277 | 0.0329 | 0.3991 | 0.1925 | 0.1359 |
| | **PRMNBR** | **0.1411** | **0.1523** | **0.1465** | **0.4639** | **0.3086** | **0.2366** |

Here, $score_i$ represents score of positive items $score_k$ represents scores of negative items and $score_m$ represents maximum score given. $B_{l(s)}$ gives the set of predicted items for every user. $V/B_{l(s)}$ represents the set of negative items predicted to each user in $B_{l(s)}$. Loss function given in Eq. (10) is optimized using rmsprop.

## 6. RESULTS

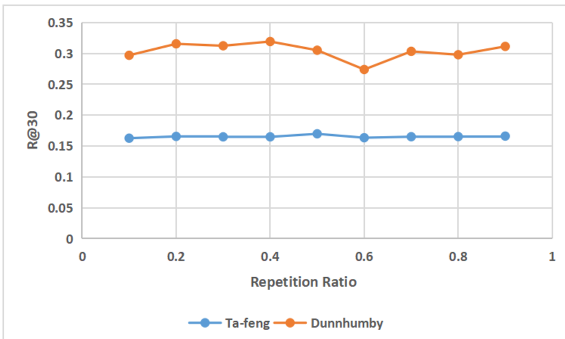
This section shows results experimental results over various real-world data sets, namely Tafeng (https://www.kaggle.com/datasets/chiranjivdas09/ta-feng-grocery-dataset) and Dunnhumby (https://www.dunnhumby.com/source-files/). This section shows comparative analysis with the other state of arts as shown in Table 2 and Table 3. Details of the two real world data sets are as follows:

- *TaFeng* - TaFeng is a grocery data set which contains 2391 users and 4648 items. It presents us with their customers shopping data of one year. Purchases made on same day has been kept in same basket
- *Dunnhumby* - Dunnhumby is also grocery data set containing 8057 users and 4990 items. The transaction made available expands to 48 weeks. Items brought on same date has been kept in same basket.

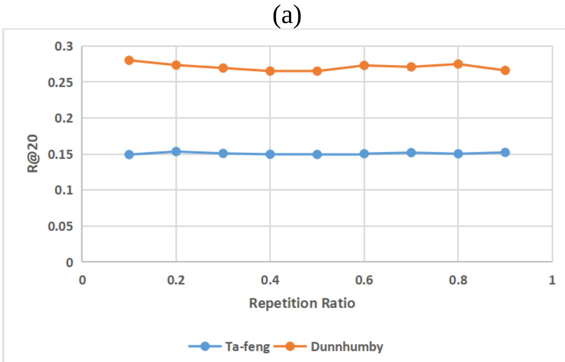
### 6.1 Preprocessing

Every user's transactions are arranged in sequential order so that it could be given in temporal order to the RNN architecture. Arranging the baskets temporally and feeding them to the model makes the study of sequential dynamics easier. The last basket is treated as truth data. Users with 3 or less baskets are removed. Each transaction is regarded as a sequential process. We have kept the value of $d_1$ at 64 for training. Training data is used for the construction of correlation matrix. Since this introduces the repetition-based concept, this model may underperform in data sets that have low repetition ratios or where users don't repetitively buy the same items.

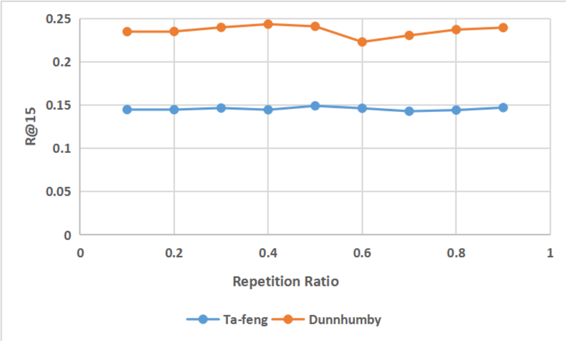
## 7. MODEL AND PARAMETER ANALYSIS



(a)

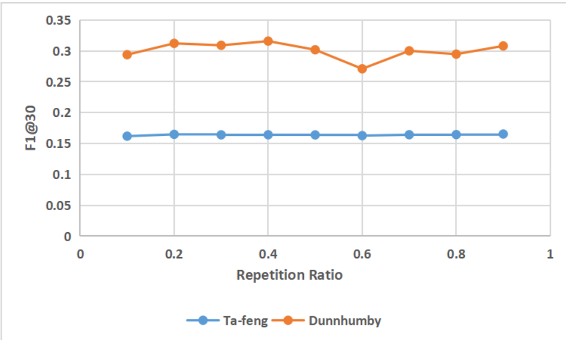


(b)

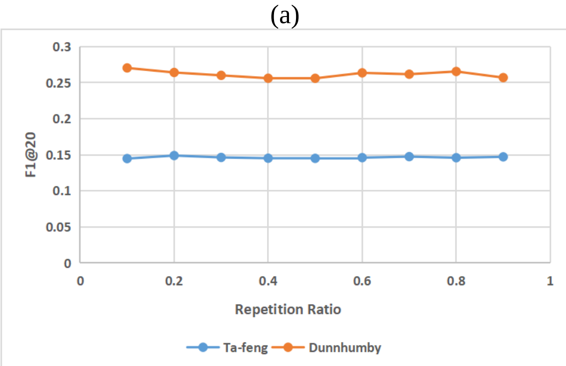



(c)

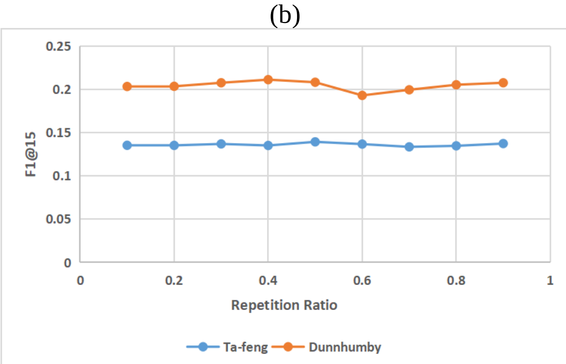
**Figure 3.** Variations of Recall against various values of repetition ratio




(a)




(b)




(c)

**Figure 4.** Variations of F1 score against various values of repetition ratio

To fine-tune sequential and general dynamics, this model uses two hyper-parameters: repetition ratio (γ) and another parameter, α. Given that the values of these two parameters vary depending on the data, the optimization is data-specific. These two hyper-parameters cannot be utilized as training weights because they may differ significantly for various data sets. The significance of repetition in a given data collection is measured by the repetition ratio. The distinguishing factor between sequential and general recommendations is established using α. The best result for the Dunnhumby data set is obtained at α value of 0.3, whereas the best result for Tafeng is found at α value of 0.5. The implementation is performed several times with different settings for these parameters, and the results are displayed graphically. It is evident that the best results are attained at repetition ratios of 0.7 and 0.3 for the Dunnhumby and Tafeng data sets, respectively. It makes it very evident that user repetition rates across data sets might differ significantly. Results clearly show that when the number of recommendations is increased, this model outperforms other models in an experimental setting. Figure 3 and Figure 4 display variation in precision and F1score along γ.

## 8. CONCLUSION

Existing Deep learning-based recommendation models do not include the study of repetition-based dynamics. Beacon [21] highlights the necessity of repetition dynamics in real-world recommendation data. New mechanisms can be introduced to analyze repetition-based dynamics. This work introduces Repeat Aware Basket to include the repetition behavior of the user into the model. Repeat aware baskets points out the repetition patterns of user or further how frequently a user interacts with particular item. As per our knowledge, this is the first work to include the new feature, repetition behavior, in recommendation models. The novelty in this work is more at the pre-processing level rather than at the model level. Repetition dynamics is introduced in the form of vector to be fed to LSTM. Repeat-aware baskets successfully capture repetition dynamics specific to that user. It gives the intuition of how long time a user will take to repeat an item. In order to enhance the performance, a correlation matrix is used to capture item-item correlation efficiently for making general recommendation. Further, embedding generated using correlation dynamics and repetition dynamics fed to the two layered parallel LSTM architecture. Experimental results show that our model outperforms other state-of-the-art models over two real-world data sets. Future works can be done by creating models that utilize repetition behaviour more effectively and further establishes a fine-tune balance between general and sequential recommendation.

## REFERENCES

[1] Che, B., Zhao, P., Fang, J., Zhao, L., Sheng, V.S., Cui, Z. (2019). Inter-basket and intra-basket adaptive attention network for next basket recommendation. IEEE Access, 7: 80644-80650. https://doi.org/10.1109/ACCESS.2019.2922985

[2] Gan, M., Ma, Y. (2022). DeepInteract: Multi-view features interactive learning for sequential recommendation. Expert Systems with Applications, 204: 117305. https://doi.org/10.1016/j.eswa.2022.117305

[3] Zhang, L., Wang, P., Li, J., Xiao, Z., Shi, H. (2021). Attentive hybrid recurrent neural networks for sequential recommendation. Neural Computing and Applications, 33: 11091-11105. https://doi.org/10.1007/s00521-020-05643-7

[4] Ding, C., Zhao, Z., Li, C., Yu, Y., Zeng, Q. (2023). Session-based recommendation with hypergraph

convolutional networks and sequential information embeddings. Expert Systems with Applications, 223: 119875. https://doi.org/10.1016/j.eswa.2023.119875

[5] Van Maasakkers, L., Fok, D., Donkers, B. (2023). Next-basket prediction in a high-dimensional setting using gated recurrent units. Expert Systems with Applications, 212: 118795. https://doi.org/10.1016/j.eswa.2022.118795

[6] Li, M., Bao, X., Chang, L., Gu, T. (2022). Modeling personalized representation for within-basket recommendation based on deep learning. Expert Systems with Applications, 192: 116383. https://doi.org/10.1016/j.eswa.2021.116383

[7] Fouad, M.A., Hussein, W., Rady, S., Philip, S.Y., Gharib, T.F. (2022). An efficient approach for rational next-basket recommendation. IEEE Access, 10: 75657-75671. https://doi.org/10.1109/ACCESS.2022.3192396

[8] Pan, Z., Cai, F., Chen, W., Chen, H. (2022). Session-based recommendation with an importance extraction module. Neural Computing and Applications, 34(12): 9813-9829. https://doi.org/10.1007/s00521-022-06966-3

[9] Hai, T., Zhou, J., Lu, Y., Jawawi, D.N., Sinha, A., Bhatnagar, Y., Anumbe, N. (2023). Posterior probability and collaborative filtering based Heterogeneous Recommendations model for user/item Application in use case of IoVT. Computers and Electrical Engineering, 105: 108532. https://doi.org/10.1016/j.compeleceng.2022.108532

[10] Gwadabe, T.R., Liu, Y. (2022). IC-GAR: Item co-occurrence graph augmented session-based recommendation. Neural Computing and Applications, 34(10): 7581-7596. https://doi.org/10.1007/s00521-021-06859-x

[11] Rendle, S., Freudenthaler, C., Schmidt-Thieme, L. (2010). Factorizing personalized Markov chains for next-basket recommendation. In Proceedings of the 19th International Conference on World Wide Web, pp. 811-820. https://doi.org/10.1145/1772690.1772773

[12] Wang, P.F., Guo, J.F., Lan, Y.Y., Xu, J., Wan, S.X., Cheng, X.Q. (2015). Learning hierarchical representation model for NextBasket recommendation. In Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 403-412. https://doi.org/10.1145/2766462.2767694

[13] Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T. (2016). A dynamic recurrent model for next basket recommendation. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 729-732. https://doi.org/10.1145/2911451.2914683

[14] Tang, J., Wang, K. (2018). Personalized top-n sequential recommendation via convolutional sequence embedding. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 565-573. https://doi.org/10.1145/3159652.3159656

[15] Yan, A., Cheng, S., Kang, W.C., Wan, M., McAuley, J. (2019). CosRec: 2D convolutional neural networks for sequential recommendation. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 2173-2176. https://doi.org/10.1145/3357384.3358113

[16] Wang, S., Hu, L., Wang, Y., Sheng, Q.Z., Orgun, M., Cao, L. (2020). Intention2basket: A neural intention-driven approach for dynamic next-basket planning. In Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence {IJCAI-PRICAI-20}.

[17] Hu, H., He, X., Gao, J., Zhang, Z.L. (2020). Modeling personalized item frequency information for next-basket recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1071-1080. https://doi.org/10.1145/3397271.3401066

[18] Ariannezhad, M., Jullien, S., Li, M., Fang, M., Schelter, S., de Rijke, M. (2022). ReCANet: A repeat consumption-aware neural network for next basket recommendation in grocery shopping. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1240-1250. https://doi.org/10.1145/3477495.3531708

[19] Zimdars, A., Chickering, D.M., Meek, C. (2013). Using temporal data for making recommendations. arXiv preprint arXiv:1301.2320. https://doi.org/10.48550/arXiv.1301.2320

[20] Du, R., Kuang, D., Drake, B., Park, H. (2017). DC-NMF: Nonnegative matrix factorization based on divide-and-conquer for fast clustering and topic modeling. Journal of Global Optimization, 68: 777-798. https://doi.org/10.1007/s10898-017-0515-z

[21] Le, D.T., Lauw, H.W., Fang, Y. (2019). Correlation-sensitive next-basket recommendation. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 2808-1814.

[22] Peng, B., Ren, Z., Parthasarathy, S., Ning, X. (2022). $M^2$pht: Mixed models with preferences, popularities and transitions for next-basket recommendation. IEEE Transactions on Knowledge and Data Engineering, 35(4): 4033-4046. https://doi.org/10.1109/TKDE.2022.3142773

[23] Bhat, A., Chandra, R. (2021). Sequential recommendation with temporal context via convolutional sequence embedding. In 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, pp. 989-995. https://doi.org/10.1109/ICICCS51141.2021.9432323

[24] Yang, Z., Wang, H., Zhang, M. (2020). WGIN: A session-based recommendation model considering the repeated link effect. IEEE Access, 8: 216104-216115. https://doi.org/10.1109/ACCESS.2020.3041772

[25] Sinha, K.K., Suvvari, S. (2024). Repetition dynamics-based deep learning model for next basket recommendation. SN Computer Science, 5(1): 1-11. https://doi.org/10.1007/s42979-023-02403-x