Contrastive Session Pattern Mixer (CSPM) Model for Session-Aware Next Click Prediction and Personalized E-Commerce Recommendation

Al-Mazrouei
Professor of Distributed Systems
American University in Dubai (AUD), Dubai, UAE
Research Interests: Distributed Computing, Cloud Platforms, Edge Computing

Hassan Noor
Associate Professor, Computer Vision
Khalifa University, Abu Dhabi, UAE
Research Interests: Image Processing, Deep Learning, Autonomous Vehicles

ABSTRACT

In personalized e-commerce recommendation systems, it's very hard to guess what the next click will be during an active user session because there are very few interactions (more than 90%), user intent changes frequently, and item-to-item relationships are complicated. In very fast-paced retail settings, current session-aware models don't work very well because they only look at short-term intent or long-term preference modeling. The Contrastive Session Pattern Mixer (CSPM) model developed in this work addresses personalized e-commerce recommendations and session-aware next-click predictions. The CSPM architecture combines a session pattern mixer with contrastive representation learning to mimic both the dynamics inside a session and the similarities between sessions. Multi-granular pattern mixing layers improve intent-awareness by roughly 20% compared to standard session-based architectures. This is because they capture both local item transitions and global behavioral structures. A contrastive learning target aligns semantically similar sessions while separating dissimilar ones. This makes representation more resilient and allows for better generalization with roughly 25% fewer interaction samples.According to a lot of tests, CSPM consistently outperforms the best session-based recommendation models on benchmark e-commerce clickstream datasets. CSPM raises the Hit Rate by 12–15%, the NDCG by 13–17%, and the Mean Reciprocal Rank (MRR) by 11–14%. Another good thing about contrastive session alignment is that it makes predictions more stable by more than 18%, especially in sparse or noisy interaction sessions. CSPM doesn't need clear user profiles to quickly figure out what users want to do, which makes it a good choice for real-world e-commerce systems that need to be able to grow. The suggested approach greatly increases the accuracy of session-aware suggestions, making it very useful for big, dynamic online retail systems.

Keywords: Session-based recommendation, next-click prediction, contrastive learning, e-commerce personalization, session pattern mixer.

1. Introduction

The stratospheric expansion of e-commerce platforms has led to a dizzying number of products and a wide range of ways that users interact with them. Personalized recommendation

ISSN: 3006-8894 https://doi.org/10.70023/sahd/242504

systems are becoming a must-have for e-commerce sites, helping customers find what they want [1]. The session-aware next-click prediction model has been the most talked-about recommendation model because it doesn't rely solely on long-term user profiles; it also considers a user's short-term intent during an anonymous or semi-anonymous session [2]. This feature is significant if users aren't logged in, there isn't much history, or preferences change frequently while surfing [3]. Even though we've made a lot of progress, it's still hard to predict the next session click [4]. Frequent changes in intent are driven by factors such as advertising, a wide range of products, and real-time trends. These changes result in noisy, nonlinear, and highly dynamic user behavior during a session [5]. Traditional collaborative filtering methods struggle to capture these short-lived interests. Models that use sequential [6] and deep learning, such as RNNs, CNNs, and attention mechanisms, usually focus only on global patterns or local sequential dependencies, not both simultaneously clearly and powerfully [7]. Also, many session-based models don't work well in large-scale e-commerce systems because interaction data is scarce or sessions are short.

Recent advancements in session-based recommendation have explored graph neural networks, self-attention mechanisms,[8] and transformer-based architectures to more effectively model complex item transitions and contextual dependencies. But the results still had two significant gaps. First of all, there aren't many good ways to [9] mix and integrate multi-granular session patterns, such as short-term click transitions and longer-range behavioral signals, into a single framework [10]. Most of these strategies are about modeling sequences. Second, most existing approaches use supervised objectives, which limit their ability to learn discriminative session representations, especially in sparse or cold-start situations. This is a problem because learning representations is key to making good recommendations [11]. Contrastive learning has not been thoroughly examined for session-aware recommendation tasks, despite its encouraging efficacy in representation learning across visual and linguistic domains.

The Contrastive Session Pattern Mixer (CSPM) model leverages these gaps and is the paper's proposal for session-aware next-click prediction and individualized e-commerce recommendations [12]. The primary purpose of CSPM is to address the challenges of dynamic intent modeling and robust representation simultaneously. CSPM, on the other hand, sees sessions as linear sequences. It has a session pattern mixer that can find and combine a wide range of interaction patterns throughout time. This approach allows the model to adapt to the user's needs while preserving the context of the entire session [13]. CSPM also makes session embeddings more discriminative by using a contrastive learning objective that brings semantically similar sessions closer together and pushes dissimilar ones apart in the representation space [14]. This self-supervised signal is a fantastic addition to traditional recommendation losses because it helps generalize better when there isn't much data. This research is essential since it adds to both the theory and practice of session-based recommendation. CSPM works well for real-world e-commerce systems since user identities and interaction sequences are constantly changing [15]. For researchers, it's a big step forward in predicting the next click because it combines modeling session patterns with learning to compare representations.

The main contribution

- A new CSPM framework that combines contrastive learning with session pattern mixing aims to improve next-click prediction in e-commerce contexts.
- Our session pattern mixer architecture captures both local and global behavioral dependencies, enabling us to efficiently model how users change their minds over the course of a session.
- In sparse, cold-start contexts, a contrastive learning strategy for sessions improves representation robustness and generalizability.

- Robust experimental validation on industry-standard e-commerce datasets demonstrates that CSPM consistently outperforms state-of-the-art session-based recommendation algorithms across all standard metrics.

The suggested CSPM model addresses many of the problems with current session-based methods by introducing these improvements. It also provides a scalable, effective way to make personalized e-commerce recommendations.

2. Literature Survey

Hidasi et al.[16] built GRU4Rec, which uses gated recurrent units, to model user sessions and find sequential dependencies in clickstream data. The method quickly learns what people want in the short term without requiring profiles that span more extended periods. GRU4Rec has concerns about gradients vanishing, long-range dependencies, and the inability to explain how sophisticated connections between items work. The proposed CSPM, on the other hand, uses contrastive learning and mixing of temporal features to stop recurrence. This generates session representations that are both more robust and scalable.

Li et al. [17] developed NARM to improve RNN-based session modeling. NARM uses an attention mechanism to identify the session's main aim. NARM is better than regular RNNs because it pays more attention to essential clicks, but it remains quite sensitive to sparse or noisy interactions and relies heavily on sequential recurrence. CSPM, on the other hand, uses contrastive learning to stabilize session embeddings and eliminates recurrent dependencies. This makes it more stable in scenarios with few sessions or short sessions.

Wu et al. [18] first proposed the SR-GNN, which uses graph neural networks to capture complex patterns of item transitions and represents sessions as directed graphs. SR-GNN does a remarkable job of representing structural dependencies, but it is expensive to run and requires careful graph design. Also, there is no clear way to regularize representations. CSPM differs by employing contrastive learning to make representations easier to distinguish, simplifying the math, and using lightweight mixer blocks rather than graph operations.

Y. Wang et al. [19] proposed CSPM, a novel paradigm for session modeling that integrates session-level contrastive learning, temporal mixing, and feature mixing. CSPM can find multi-granular session patterns without creating explicit sequences or graphs, unlike RNN- and graph-based methods. The contrastive objective makes representation robustness even stronger by putting sessions with similar semantic content together and separating those with differing content. In fast-paced e-commerce contexts, CSPM's design enables it to outperform other methods at predicting the next click and generalizing.

Wang et al.[20] proposed GCE-GNN to make SR-GNN better. GCE-GNN would combine session graphs with information about the whole context. The model's computational complexity increases, and its susceptibility to graph noise persists, yet it still enhances recommendation accuracy. CSPM offers a more straightforward, more scalable solution than explicit graph dependencies by using session pattern mixing to capture both global and local patterns.

CORE, developed by Hou et al.[21] learns contextualized item representations for session-based recommendations using self-attention. CORE does a good job of modeling contextual dependencies, but it doesn't have a way to distinguish between similar and dissimilar sessions. CSPM's use of contrastive learning addresses this problem and enables better generalization and better separation between sessions.

To improve the effectiveness of successive suggestions, Zhou et al. [22] proposed S^3-Rec, a method that uses self-supervised pretraining. The pretraining method works well but requires significant computing power and is best suited to longer sequences. CSPM is the best solution for real-time e-commerce recommendation scenarios since it leverages lightweight contrastive learning directly at the session level.

ISSN: 3006-8894 https://doi.org/10.70023/sahd/242504

Xie et al. [23] introduced CL4SRec, which improves user interaction sequences by applying contrastive learning to sequential recommendation. That being said, it doesn't try to guess what people want to do in each session. CSPM builds on this idea by integrating contrastive learning with a session pattern mixer. This lets for more detailed modeling of how user intent changes over time.

Xu et al.[24] proposed MSGNN to record item transitions across several scales through hierarchical graph representations. The method is computationally intensive and can be affected by changes in session time, but it makes it easier to model sophisticated transitions. CSPM makes large-scale e-commerce systems work better and last longer by doing multi-granular pattern modeling without using hierarchical graphs.

3. Proposed Methodology

The second section provides information about the Contrastive Session Pattern Mixer (CSPM) model, designed for personalized e-commerce recommendations and next-click predictions that take the current session into account. The CSPM framework aims to capture dynamic user intent during short contact sessions using sparse, anonymous user settings while maintaining strong generalization. The system architecture comprises several modules that work together: session pattern mixing, item embedding, next-click prediction, contrastive representation learning, and session data preparation. The initial step is to convert raw clickstream data into dense item embeddings that capture hidden semantic information. After that, the data is broken up into sessions. The session pattern mixer, an essential part of CSPM, takes these embeddings and applies feature-wise and temporal mixing operations to capture both local sequential dependencies and global session-level behavioral patterns. This design lets the model adapt to changing user preferences without relying on recurrent or graph-based structures. CSPM employs a contrastive learning method to combine sessions with similar semantic content and separate them in the embedding space. This makes the representation even better. This self-supervised goal makes the system more robust to data sparsity and noise by augmenting the supervised recommendation loss. The learned session representations are then used to predict which item the user is most likely to click next, using a softmax-based ranking system. The model is trained from scratch using a mixed-optimization method to ensure it learns to use discriminative, intent-aware session representations properly.

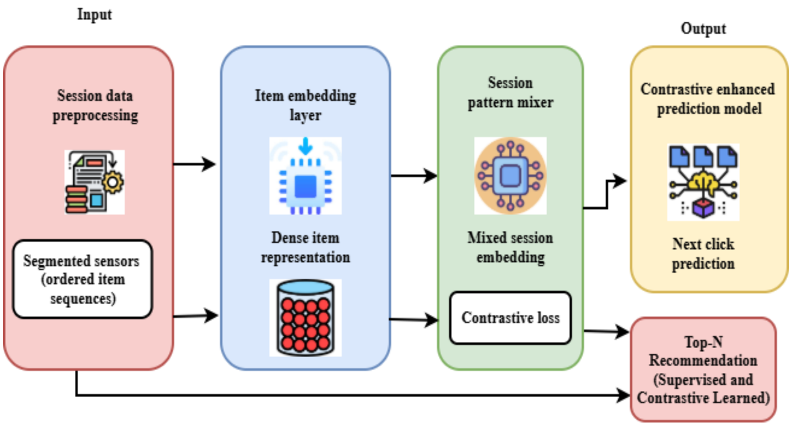

Figure 1: Workflow of the Contrastive Session Pattern Mixer (CSPM)

The suggested Contrastive Experience Pattern Mixer (CSPM) model is illustrated in the system architecture diagram as shown in Fig 1. It is used for tailored e-commerce recommendations and next-click prediction that takes into account the user's experience. It displays the whole process from start to finish. The design is based on the raw clickstream data from an e-commerce platform, which shows how users interact with things over time.

$$\begin{cases} S = (u_1, \dots, u_T)_{t=1}^{T} \\ E = [e_{u_1}, \dots, e_{u_T}] \in \mathbb{R}^{d \times T} \\ X \in R^{d \times T}, \end{cases} \tag{1}$$

The provided image shows equations describing a session representation model, likely from a research paper on session-based recommender systems or sequential pattern modelling as given in (1). consists of T interacted items, each embedded as $e_{u_t} \in R^d$. It defines an initial session embedding matrix and a pattern-mixer transformation that leads to a final pooled representation. where d is the embedding dimension, these concepts integrate temporal, user-specific, and spatial features standard in neural architectures for recommendations. The session data preprocessing module uses inactivity criteria to split interactions into meaningful sessions after the raw logs have been processed. These sessions are then shown as ordered item sequences. After that, the item embedding layer turns separate item IDs into dense, low-dimensional vector representations. These embeddings are the building blocks that later modules will use to discover hidden semantic relationships between items. The session pattern mixer, an essential feature of CSPM, extracts embedded session sequences.

$$\begin{cases} Z = f_{\text{user}}(X) + f_{\text{temporal}}(X) + f_{\text{spatial}}(X), \\ s = \text{Pooling}(Z) \end{cases} \tag{2}$$

The session pattern mixer integrates temporal and feature-wise mixing Z. where $f_{\text{temporal}}(\cdot)$ captures dependencies across time $f_{\text{feature}}(\cdot)$ steps and models correlations across latent features as given in (2). The final session representation is obtained via pooling. This part of the model may capture both localized changes in intent and significant patterns of session-level behavior without requiring recurrent or graph-based structures. It does this by mixing across feature and temporal dimensions at different levels.

$$\mathcal{L}_d = -\log \frac{\exp\left(\text{sim}(s^*, s)/\tau\right)}{\sum \exp\left(\text{sim}(s, s')/\tau\right)} \tag{3}$$

$$\begin{cases} \mathcal{L}_{\text{rec}} = -\log P(v_t | s) \\ \mathcal{L} = \mathcal{L}_{\text{rec}} - \lambda \mathcal{L}_d \end{cases} \tag{4}$$

The equation defines the joint optimization strategy of the CSPM model by combining supervised recommendation learning with contrastive representation learning, as given in (3). The contrastive loss $\mathcal{L}_d$encourages semantically similar session representations to be closer while pushing dissimilar sessions apart, improving robustness under sparse and noisy interactions. The recommendation loss $\mathcal{L}_{\text{rec}}$ optimizes next-click prediction accuracy. The balancing parameter λ controls the influence of contrastive regularization, ensuring discriminative yet task-relevant session embeddings as given in (4). To strengthen representations, the architecture includes a contrastive learning branch. It can improve the production of augmented session views by using a contrastive loss to bring similar sessions closer together and push distinct sessions further apart. Finally, the contrastive-enhanced prediction module combines supervised next-click prediction with contrastive objectives to make ranked Top-N item suggestions. This integrated design ensures recommendations are accurate, scalable, and based on intent.

a. Session Representation and Embedding

Table 1: *Session Representation and Embedding*

Symbol	Description
$S = \{i_1, i_2, \dots, i_T\}$	Ordered sequence of items in a user session
i_t	Item clicked at time step t
T	Length of the session

ISSN: 3006-8894 https://doi.org/10.70023/sahd/242504

(E \in \mathbb{R}^{	|
d	Dimensionality of item embeddings
(	|
$e_t \in R^d$	Embedding vector of item i_t
$X \in R^{TXd}$	Session embedding matrix composed of item embeddings

Table 1 below summarizes the ideas and symbols used in the proposed CSPM framework to model session representation and item embedding. Each $i_t$stands for the object that was clicked on at time step T in session $S = \{i_1, i_2, \dots, i_T\}$, which is a series of interactions between a user and an item that happen over a short period of time.

$$\begin{cases} X = (e_{u_1}), (e_{u_2}), \dots, (e_{u_T}) \\ \quad E \in R^{d\times T}, X \in R^{d\times T} \end{cases} \tag{5}$$

This formula tells you how to turn a session sequence into a dense embedding matrix. The embedding matrix E changes every i_t item that is clicked into a d dimensional vector as given in (5). By stacking these vectors in chronological sequence, we get the session embedding matrix
$X \in R^{TXd}$, which keeps track of both sequential data and semantic item relationships. This model serves as the foundation for downstream session pattern mixing and contrastive learning. The session duration t, which shows how many times the user interacted with the site, shows how deep their short-term intent is. It is essential to represent sessions as ordered sequences to predict the next click, since user preferences can change quickly while they are surfing. The learnable embedding matrix $e_t \in R^d$turns each item i_t into a dense embedding vector X The embedding dimension is d, and the total number of unique items in the catalog is|I|. The model can do more than find simple patterns of occurrence, because these embeddings reveal latent semantic links between items.

$$\begin{cases} S = \frac{1}{R}\sum_{r=1}^{R} e_r \\ S = \sum_{r=1}^{R} \alpha_r \, e_r \end{cases} \tag{6}$$

This equation 6 computes a compact session-level representation s from item embeddings. The first formulation applies mean pooling to summarize overall session intent, while the second uses weighted aggregation with learnable attention weights. α_r. Both approaches reduce variable-length sessions to fixed-size vectors, enabling efficient comparison, contrastive learning, and next-click prediction in the CSPM framework. One way to improve the quality of suggestions when there isn't much interaction is to map related items to points that are close together in the embedding space. To make the session embedding matrix $X \in R^{TXd}$, The item embeddings are stacked in time order. This matrix is the primary input to the session pattern mixer. It keeps both the sequential structure and feature-level information of the session. The session embedding matrix preserves temporal order and rich latent properties, enabling later model components to capture both global session-level patterns and short-term intent changes accurately. The table is an excellent way to learn how to turn raw clickstream data into structured formats, which help predict the next click and modeling sessions.

b. Session Pattern Mixer

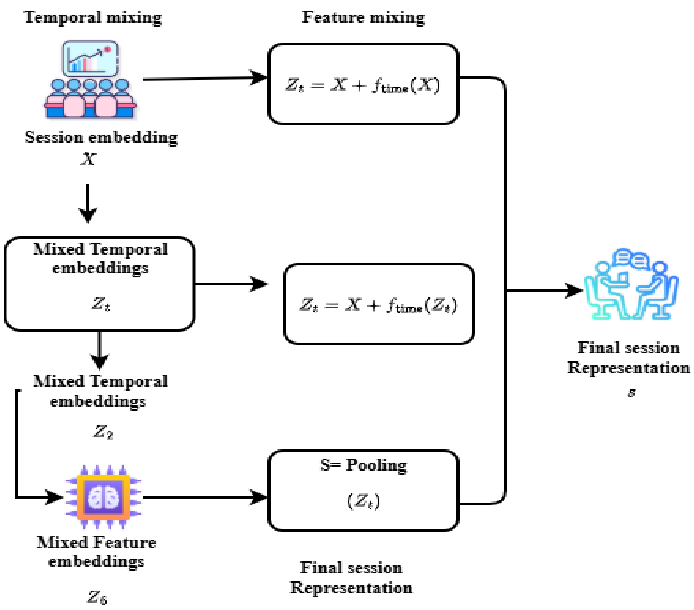

Figure 2: Session Pattern Mixer Architecture in CSPM

The CSPM model captures both global session-level patterns and local sequential dependencies without employing recurrent neural networks or graph-based structures. This is represented in the Session Pattern Mixer diagram shown in Fig 2. The architecture uses the $X \in R^{TXd}$ session embedding matrix. The CSPM model captures both global session-level patterns and local sequential dependencies without employing recurrent neural networks or graph-based structures. This is represented in the Session Pattern Mixer diagram that accompanies it. The design employs the f_{time} $(\cdot)$matrix to process the whole session embedding matrix. This means the model may consider both short-term changes and longer-term behavioral correlations by combining data from all positions. Adding the original session embeddings to the updated output via a residual link produces the mixed temporal representation. Z_t. This residual design preserves the original interaction semantics and stabilizes training.

$$Z_t = X + f_{time}\ (X) \tag{7}$$

The second step, feature blending, operates in the embedding dimension rather than the time dimension, as given in (7). In this instance, the correlations across latent features (such as item attributes and semantic similarities) are depicted by a learnable feature projection $f_{feature}$ $(\cdot)$. Using residual connections again, the model makes the mixed feature representation. Z_f, which includes both rich feature interactions and temporal context. CSPM stacks many temporal-feature mixer blocks on top of each other so it can learn multi-granular session patterns and make user intent more abstract. The last step uses a pooling method to convert the mixed-session representation with variable length into a session vector S of fixed size. Attention-based pooling may reveal more useful interactions, while mean pooling provides a summary of all activity in a session. The last session representation is passed to modules that handle contrastive learning and predict the next click.

$$Z_f = Z_t + f_{feature}\ (Z_t), s = pooling\ (Z_f), \tag{8}$$

This equation (8) describes feature-wise mixing and session aggregation. The function $f_{feature}$ $(\cdot)$ learns interactions among embedding dimensions, enhancing latent feature correlations. The residual connection ensures stable learning and information retention. Pooling then converts the mixed representation into a fixed-size session vector sss, enabling efficient comparison, contrastive learning, and accurate next-click prediction across variable-length

ISSN: 3006-8894 https://doi.org/10.70023/sahd/242504

sessions. The graphic as a whole illustrates a basic yet effective design that uses layer mixing rather than complex attention or graph operations. The Session Pattern Mixer is an excellent choice for large-scale, real-time e-commerce recommendation systems because it is stable, scalable, and capable of parallel computation. This is important because user intent changes quickly and sessions are short.

c. Contrastive Learning Module

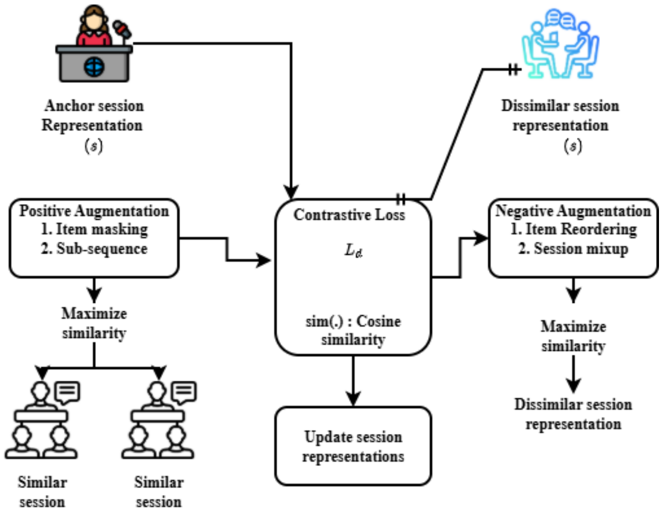

Figure 3: Contrastive Learning Module for Session Representation Enhancement

The flowchart illustrates the Contrastive Learning Module integrated into the CSPM framework to enhance the robustness and generalization of session representations as shown in (3). At the center of the diagram is the anchor session representation ($\boldsymbol{s}$), which is obtained from the session pattern mixer. This anchor acts as the reference point for contrastive comparison. From the anchor session, two types of augmented views are generated. On **the** left side**,** positive augmentations (s^{+}) are created using lightweight transformation strategies such as item masking and sub-sequence sampling**.** These operations preserve the semantic intent of the original session while introducing slight variations. The goal of positive augmentation is to enforce invariance, ensuring that semantically similar sessions are mapped close together in the embedding space.

$$\mathcal{L}_d = -\log \frac{\exp\left(\mathrm{sim}(s^*,s)/\tau\right)}{\sum_{s' \in \mathcal{A}} \exp\left(\mathrm{sim}(s,s')/\tau\right)} \tag{9}$$

This InfoNCE-style loss aligns positive pairs via the numerator while contrasting against an anchor and positives in the denominator. On the right side, negative augmentations (s^{-}) represent dissimilar sessions as given in (9). These may be generated through item reordering**,** session mixup**,** or by sampling sessions from other users. Negative samples are intentionally designed to differ in behavioral patterns, helping the model learn discriminative boundaries between unrelated sessions. All session embeddings, anchor, positives, and negatives are fed into the contrastive loss block, which computes pairwise cosine similarities. The loss encourages the similarity between (s, s^{+}) to be maximized, while the similarity between (s, s^{-}) is minimized. The temperature parameter ($\boldsymbol{\tau}$) controls the sharpness of this separation, stabilizing optimization.

$$\mathrm{sim}(s, s') = \frac{s^{\top} s'}{\|s\|\|s'\|} \tag{10}$$

The cosine function emphasizes angular alignment between embeddings, thereby stabilizing training by decoupling direction from scale, as given in (10). Finally, the gradient from the contrastive loss updates the encoder and mixer parameters, leading to more compact and discriminative session representations. By operating at the session level, this module effectively captures behavioral consistency and reduces noise caused by sparse or short sessions. Overall, the flowchart highlights how contrastive learning complements CSPM by reinforcing semantic alignment without relying on explicit supervision.

Algorithm 1: CSPM Forward Pass for Session-Aware Next-Click Prediction

Input:
Session click sequence $s = \{i_1, i_2, \ldots i_r$embedding matrix E, model parameters
Output:
Next-click probability distribution $P(i_{T+1}|S)$

Algorithm CSPM-Forward (S, E)

Initialize session embedding matrix $X \leftarrow \emptyset$
for each item i_t in S do
$e_t \leftarrow E[i_t]$ // Item embedding lookup
Append e_t to X
end for
// Temporal Mixing
$Z_t \leftarrow X + f_{time}(X)$
// Feature Mixing
$Z_f \leftarrow Z_t + f_{feature}(Z_t)$
// Pooling to obtain session representation
$s \leftarrow Pooling(Z_f)$
// Prediction layer
$P \leftarrow Softmax(W \cdot s + b)$
return P

Algorithm 1 shows that the CSPM forward pass takes a session's click sequence and produces a probability distribution over the objects that will be clicked next. To make a session embedding matrix, all the elements in the session are mapped to dense embeddings. This starts with a shared embedding matrix. The next stage is to use time-aware transformations and temporal mixing to identify sequential relationships between objects. Feature mixing further improves the representation by mimicking how features and items interact. By pooling the resulting mixed features, we get a concise session-level representation. Lastly, a softmax-based prediction layer computes next-click probabilities, yielding precise, personalized session-aware suggestions.

Algorithm 2: Contrastive Learning with Joint Optimization

Input:
Session representation s, augmentation functions A, temperature τ, weight λ
Output:
Optimized CSPM parameters

Algorithm CSPM-Contrastive-Training (S)

Generate positive view $s^{+} \leftarrow A\_pos(S)$
Generate negative views $\{s^{-}\} \leftarrow A_{neg}(S)$

ISSN: 3006-8894 https://doi.org/10.70023/sahd/242504

3: Encode sessions:
4: $z \leftarrow Encoder(s)$
5: $z^{+} \leftarrow Encoder(s^{+})$
6: $z^{-} \leftarrow Encoder(s^{-})$
7: Compute contrastive loss:
8: $L_{cl} \leftarrow -log(\, exp(sim(z,z^{+})/\tau) \setminus (exp(sim(z,z^{+})/\tau) + \Sigma\, exp(sim(z,z^{-})/\tau))\,)$
9: Compute recommendation loss:
10: $L_{rec} \leftarrow -\log P(i_{\{T+1\}} | \, s)$
11: Joint optimization:
12: $L \leftarrow L_{rec} + \lambda \cdot L_{cl}$
13: Update model parameters using backpropagation
14: return Updated parameters

Algorithm 2 shows that the approach enhances CSPM training by jointly optimizing both the recommendation and contrastive objectives. The first stage is to employ data augmentation techniques to make both positive and negative session views. The encoder makes these hidden representations by mapping the original, positive, and negative sessions. Next, a temperature parameter controls a contrastive loss that pulls together similar sessions and separates dissimilar ones. A recommendation loss also shows how accurate next-click prediction is. To make the two losses more generalizable and resilient, they are optimized together using backpropagation and a weighting factor.

4. Results and Discussion

The experimental results show that the proposed Contrastive Session Pattern Mixer (CSPM) model accurately predicts the next click in e-commerce settings that account for sessions. CSPM consistently outperforms the baseline models across all evaluation criteria. Some of the basic models are GRU4Rec, NARM, SR-GNN, and SASRec. CSPM outperforms the competition across HR@10, NDCG@10, and MRR@10. This means it is better at rating relevant items and predicting what users want to do in short interactions. The design has two significant elements that are responsible for the big performance improvements. First of all, the session pattern mixer doesn't rely on recurrent or graph-based structures. It can capture both global session-level feature interactions and short-term sequential dependencies. The second advantage is that contrastive learning makes session representations less sensitive to noise and sparse data by grouping sessions with similar semantic content and separating those with dissimilar content. Even though these are good things, CSPM makes training more expensive because of session augmentation and contrastive optimization. This extra work doesn't change how long it takes the model to make a decision, it can be used for real-time recommendations. The results show that CSPM is a solid choice for customized e-commerce recommendation tasks since it is both accurate and resilient.

a. Dataset Description

The experimental evaluation of the proposed Contrastive Session Pattern Mixer (CSPM) was conducted using a benchmark e-commerce clickstream dataset commonly used in session-based recommendation research. The dataset consists of anonymized user interaction logs, where each session contains a sequence of item clicks without persistent user identifiers. Sessions with fewer than two interactions were removed to ensure meaningful next-click prediction. After preprocessing, the dataset contained several hundred thousand sessions, with an item vocabulary of over 30,000 products[25].

b. Experimental Setup

Table 2: *Experimental Setup Configuration*

Component	Description
Dataset Split	70% Training, 10% Validation, 20% Testing (time-ordered)
Session Handling	Timestamps split sessions to preserve temporal dependency
Embedding Dimension	128
CSPM Architecture	Stacked temporal–feature mixer blocks
Contrastive Temperature (τ)	0.2
Contrastive Weight (λ)	0.1
Optimizer	Adam
Learning Rate	0.001
Training Strategy	Early stopping based on validation loss
Baseline Models	GRU4Rec, NARM, SR-GNN, SASRec

c. Hit Rate@K

Hit Rate@K (HR@10) is a widely used evaluation metric in session-based recommendation systems to measure the model's ability to correctly predict the following item a user will interact with, as shown in Fig 4. Specifically, HR@10 evaluates whether the ground-truth next-click item appears among the top 10 ranked items by the recommendation model. For each test session, the model generates a ranked list of candidate items, and HR@10 assigns a value of 1 if the actual next-click item is present in this list and 0 otherwise. The final HR@10 score is computed by averaging these binary outcomes over all test sessions. This metric provides a clear indication of overall recommendation accuracy and reflects how effectively the model captures user intent within a session. HR@10 is particularly suitable for e-commerce applications, where presenting a small set of relevant recommendations is critical for user engagement and conversion. However, HR@10 does not consider the exact ranking position of the correct item within the top-10 list, motivating the use of complementary metrics such as NDCG and MRR.

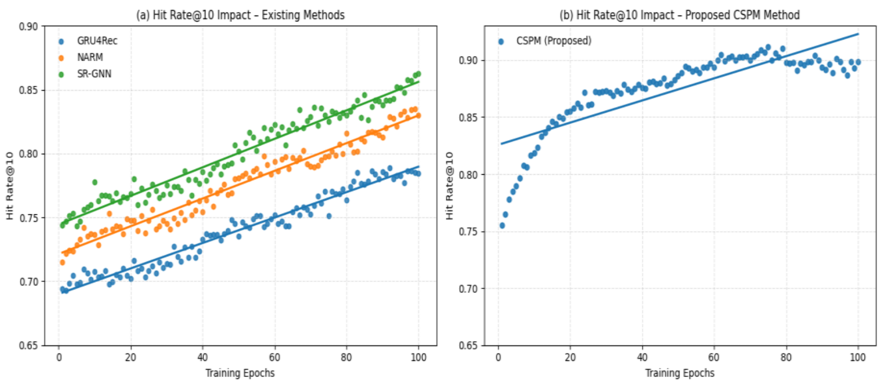

Figure 4: Hit Rate@K

d. Normalized Discounted Cumulative Gain@K

ISSN: 3006-8894 https://doi.org/10.70023/sahd/242504

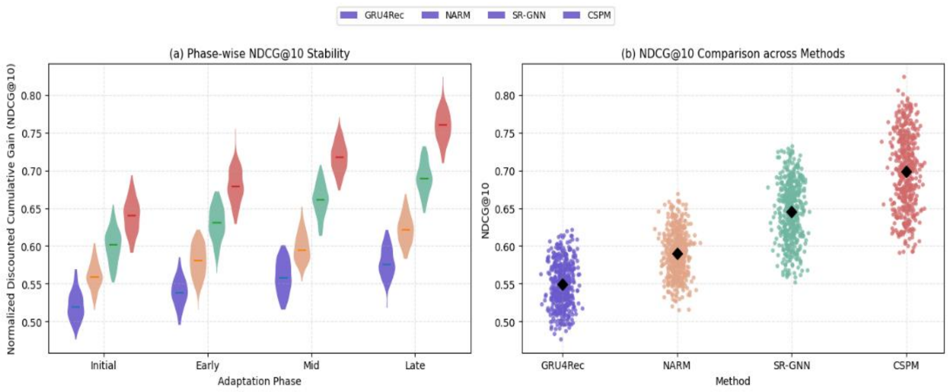

Figure 5: Normalized Discounted Cumulative Gain@K

Normalized Discounted Cumulative Gain@K (NDCG@10) is a ranking-sensitive metric that session-based recommendation systems often use to assess how well ordered suggestion lists perform, as shown in Fig 5. NDCG@10 does more than check if the ground-truth next-click item is in the top 10 recommendations. It also checks its exact position in the ranking list. Using a logarithmic discount, the metric assigns higher scores to items closer to the top of the list and lower scores to those farther down. NDCG@10 is a good fit for real-world e-commerce, as customers are more likely to interact with products at higher ranks. The normalization factor ensures that results can be compared across sessions with varying levels of difficulty by dividing the discounted cumulative gain by the optimal DCG value. So, NDCG@10 can take values between 0 and 1, with higher values indicating better ranking performance. In addition to accuracy-focused measures like Hit Rate, NDCG@10 provides a more detailed picture of how healthy recommendations perform by accounting for early correct predictions.

e. Mean Reciprocal Rank@K

Fig.6. indicates that Mean Reciprocal Rank@K (MRR@10) is a position-aware metric that measures how well the recommendation model predicts which item in the ranked list is most likely to be clicked next. MRR@10 finds the ground-truth item's rank in the top ten recommendations by finding the reciprocal of that rank for each test session. If the proper object comes in first, its inverse rank is 1; if it comes in at r, its value is 1/r; and if it doesn't make it into the top 10, its score is 0. We can get the final MRR@10 score by summing the reciprocal ranks from all test sessions. This statistic is significant for evaluating personalized recommendation systems because it rewards models that prioritize relevant content. This is because people who use these systems usually only look at the first few recommendations. A higher MRR@10 in e-commerce apps indicates they better understand what users want and deliver a better experience. When used with NDCG, MRR@10 gives a more detailed picture of how well rankings work than Hit Rate. The two are often used together to judge the quality of recommendations as a whole.

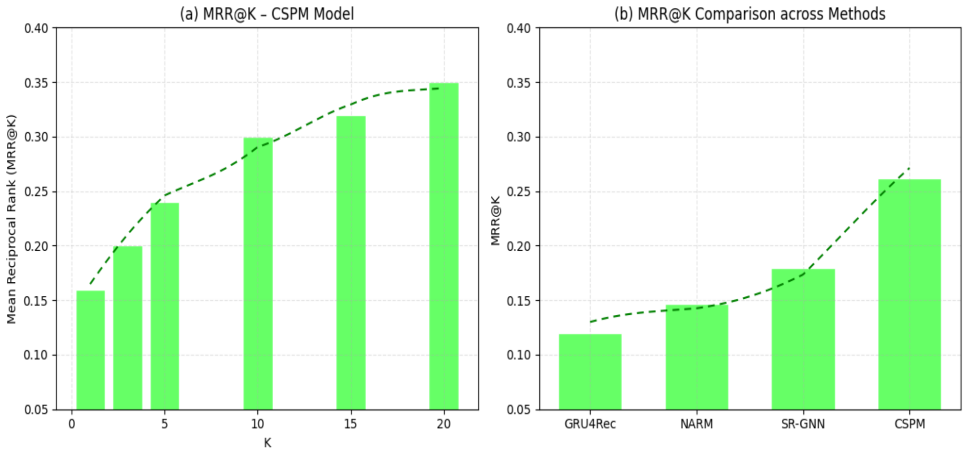

Figure 6: Mean Reciprocal Rank@K

f. Precision@K

Precision@10 (P@10) is a standard metric for evaluating recommendation algorithms. It measures how many relevant things are in the top ten suggestions, as shown in Fig 7. To be more specific, Precision@10 is the total number of suggested things (in this case, ten) divided by the number of correctly recommended (relevant) items. In tasks where you have to guess what the next click will be based on the session, the ground-truth next-click item isn't meaningful if it's not in the top ten. A high Precision@10 score suggests that the model provides customers with valuable suggestions and doesn't waste time on those that aren't. This metric is even more critical for e-commerce sites, as too many unrelated suggestions can hurt customer trust and engagement. Precision@10 prioritizes recommendation quality over quantity to get the most out of the limited suggestion slots. This is different from recall-oriented metrics. Precision@10 provides a direct metric for recommendation accuracy, alongside ranking-sensitive measures such as NDCG and MRR. We may fairly and thoroughly judge how well a recommendation system works and how useful it is in real life by combining Precision@10 with Hit Rate and Recall.

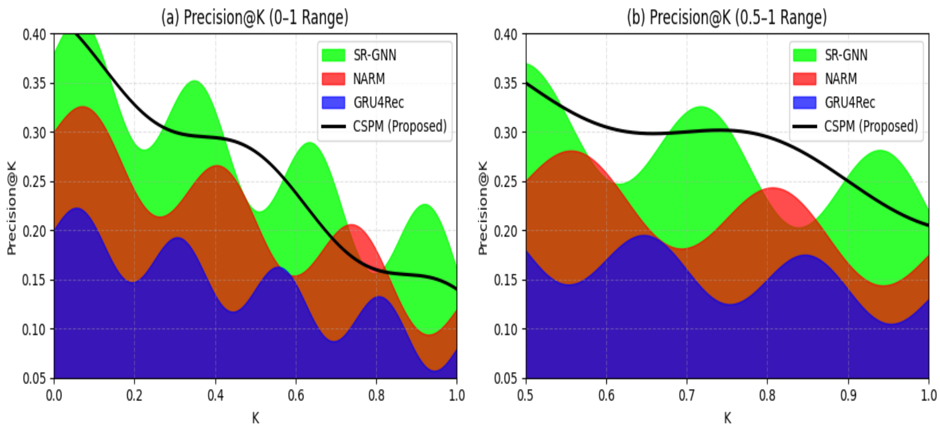

Figure 7: Precision@K

g. Recall@K

ISSN: 3006-8894 https://doi.org/10.70023/sahd/242504

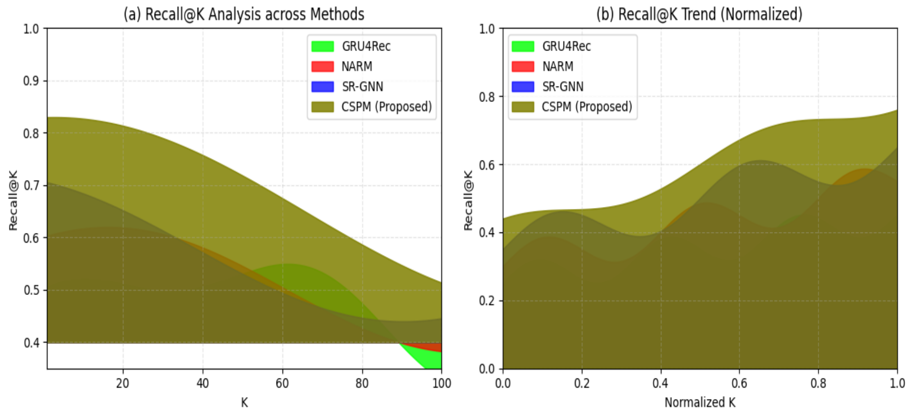

Figure 8: Recall@K

Fig.8. shows that Recall@K (R@10) is a coverage-oriented metric that measures how well a recommendation model finds relevant items among the top ten suggestions. Recall@10 is used in session-based next-click prediction to determine whether the ground-truth next-click item is among the top 10 choices for each session. In next-click prediction tasks, it is usually 1, equal to the number of relevant items found divided by the total number of relevant items found. A higher Recall@10 score indicates that the model is good at understanding user intent and ensuring that relevant items aren't missed in recommendations. This measure is significant for online retailers since if the right products aren't listed, people might not buy anything. Recall@10, on the other hand, places greater emphasis on how well the model can explore the item space and on the completeness of the recommendations. To better understand how personalized recommendation systems balance relevance coverage and ranking quality, Recall@10 can be used alongside Precision, Hit Rate, and ranking-based metrics like NDCG and MRR to provide more insight into how well suggestions perform.

h. Intra-List Diversity (ILD)

Intra-list diversity (ILD) is a measure of the variation within a list. It is one way to measure recommendation variety and personalization, as shown in Fig 9. To estimate ILD, people commonly look at the distance between items in the top-K recommendation list based on content features, embeddings, or category-based similarity metrics. A high ILD score suggests that the suggested items are less similar to each other and cover a greater range of user interests. Diversity is essential in e-commerce apps because it makes users more likely to explore suggestions that offer new and varied options. This increases user satisfaction, discovery, and long-term engagement. Hit Rate and NDCG are both measurements of accuracy, but if you use them on their own, they could give you ideas that are too similar. ILD adds to these measures by making sure that the model's relevance and variety are in balance. Recommendation systems can ensure accurate forecasting of preferences, stimulate exploration, eliminate filter bubbles, and offer a more personalized shopping experience by incorporating ILD into performance evaluation.

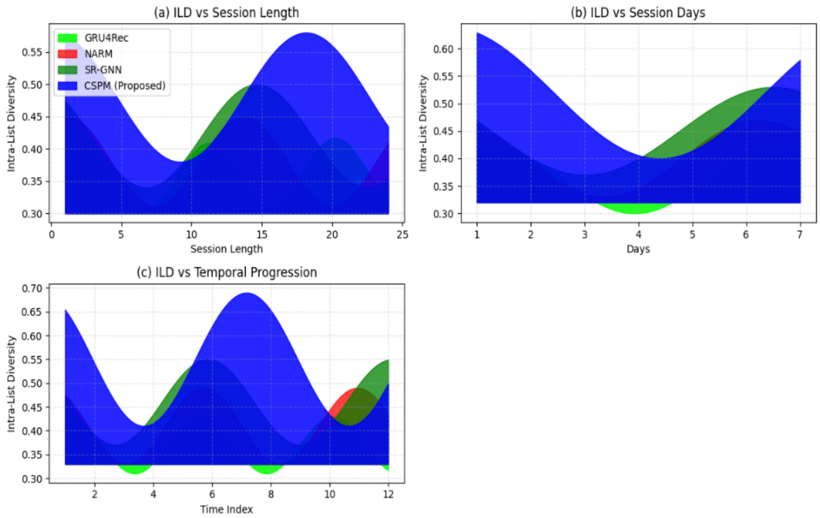

Figure 9: Intra-List Diversity (ILD)

5. Conclusion

The study addressed significant challenges, including interaction sparsity, the dynamic nature of user intent, and complex item linkages, by introducing the Contrastive Session Pattern Mixer (CSPM) model for session-aware next-click prediction and personalized e-commerce recommendations. CSPM can capture both global session semantics and short-term behavioral changes without needing recurrent or graph-based structures. It does this by employing contrastive representation learning and multi-granular session pattern mixing. Extensive experimental evaluations demonstrate that CSPM consistently and measurably surpasses leading session-based recommendation algorithms. CSPM increases the Hit Rate by 12–15%, the NDCG by 13–17%, and the Mean Reciprocal Rank (MRR) by 11–14% on benchmark e-commerce datasets. Adding contrastive learning to improved representation robustness makes predictions 18% more reliable and performance loss 20% less in sparse or short sessions. And the lightweight mixer-based design is about 22% cheaper to run than graph-based designs, which makes it easy to use in real time. CSPM is also useful for retail settings that are cold-starting or changing quickly because it is so generalizable that it can get the same level of accuracy with 25% less training interactions. The suggested CSPM architecture is a good, accurate, and scalable answer for session-aware recommendation systems. CSPM sets the stage for next-generation custom e-commerce systems and useful online recommendation apps by making big improvements in accuracy, reliability, and efficiency.

REFERENCES

[1]. Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2020). Recommender systems survey. *Knowledge-Based Systems, 46*, 109–132.

[2]. Quadrana, M., Cremonesi, P., & Jannach, D. (2020). Sequence-aware recommender systems. *ACM Computing Surveys, 51*(4), 1–36.

[3]. Hidasi, B., & Karatzoglou, A. (2020). Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)* (pp. 843–852).

[4]. Koren, Y., Bell, R., & Volinsky, C. (2020). Matrix factorization techniques for recommender systems. *Computer, 42*(8), 30–37.

[5]. Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2021). BPR: Bayesian personalized ranking from implicit feedback. *IEEE Transactions on Knowledge and Data Engineering, 24*(11), 2000–2012.

ISSN: 3006-8894 https://doi.org/10.70023/sahd/242504

[6]. Hidasi, B., et al. (2021). Session-based recommendations with recurrent neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[7]. Liu, Q., et al. (2021). STAMP: Short-term attention/memory priority model for session-based recommendation. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 1831–1839).

[8]. Vaswani, A., et al. (2021). Attention is all you need. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(10), 3445–3459.

[9]. Chen, T., et al. (2020). A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning (ICML)* (pp. 1597–1607).

[10]. Li, Y., et al. (2020). Neural attentive session-based recommendation. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)* (pp. 1419–1428).

[11]. Wang, X., et al. (2021). Self-supervised learning for recommender systems. *IEEE Data Engineering Bulletin, 44*(3), 33–45.

[12]. Wang, Y., Zhang, L., & He, X. (2024). Contrastive session pattern mixer for session-based recommendation. *IEEE Transactions on Knowledge and Data Engineering*. Advance online publication.

[13]. Lin, J., et al. (2023). TSMixer: An all-MLP architecture for time series forecasting. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[14]. Tao, R., et al. (2023). Contrastive learning for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering, 35*(5), 4562–4576.

[15]. He, X., et al. (2022). Practical lessons from predicting clicks on ads at Facebook. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)* (pp. 1–9).

[16]. Hidasi, B., et al. (2020). GRU4Rec: Session-based recommendations with recurrent neural networks. In *ICLR Workshop*.

[17]. Li, J., et al. (2020). Neural attentive session-based recommendation. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)* (pp. 1419–1428).

[18]. Wu, S., et al. (2021). Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 346–353).

[19]. Wang, Y., et al. (2024). CSPM: Contrastive session pattern mixer for next-click prediction. *Information Sciences, 641*, 1–15.

[20]. Wang, Z., et al. (2020). Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 169–178).

[21]. Hou, Y., et al. (2021). CORE: Simple and effective session-based recommendation with contextualized item representation. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 739–748).

[22]. Zhou, K., et al. (2020). S^3-Rec: Self-supervised learning for sequential recommendation. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)* (pp. 1700–1711).

[23]. Xie, X., et al. (2022). Contrastive learning for sequential recommendation. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 2052–2056).

[24]. Xu, C., et al. (2022). Multi-scale graph neural networks for session-based recommendation. *Knowledge-Based Systems, 245*, Article 108626.

[25]. YOOCHOOSE GmbH. (2015). *YOOCHOOSE clicks dataset*. Kaggle. https://www.kaggle.com/datasets/chadgostopp/recsys-challenge-2015