Information Technology and Management (2023) 24:133–146
https://doi.org/10.1007/s10799-022-00358-8

Hybrid time centric recommendation model for e-commerce applications using behavioral traits of user

B. R. Sreenivasa[1] · **C. R. Nirmala**[2]

Accepted: 7 September 2021 / Published online: 15 March 2022

Abstract
In today's online market, recommendation systems have become universal and are an aspect of any online shopping portal. The traditional approach uses the subscriber's historical knowledge, and this technique is not adequate for resolving problems with a cold start. These issues include recommendations for non-registered users or newly added customers and new items added. Session-based recommendations based on recurrent neural networks are gaining popularity for product recommendations. This is due to recurrent neural networks' ability to record sequential feature data more effectively throughout the current session, which results in more similarity between consumer behaviour sequences. Nevertheless, most state-of-the-art recurring neural networking systems completely ignore the long-term details of multiple sessions and concentrate solely on short-term communication in a single session. This paper presents a hybrid time-centric prediction model to address research issues that learn the customers' short and long-term behaviours. Experiments on the recsys challenge data set are carried out to assess the efficiency of the hybrid time-centric prediction models over the existing hybrid models in terms of HitRate and Mean-Reciprocal Rate.

Keywords Deep learning · Machine learning · Bayesian personalized recommendation · Behaviour modelling · Recommendation system · Recurrent neural network · Session-based

1 Introduction

Online shopping has become a ubiquitous way of shopping. More than 79% of people in the United States visit online shopping portals, as stated in [1]. However, the purchase of products ends at just 2–5% of these portions[1, 25]. This process is called the procurement conversion rate of online shopping portals (PCR). Considering that, as indicated by [40], present e-Commerce is estimated at more than US$ 460 billion. The income/revenue of the e-commerce platform would be increased even if the PCR online is little changed. The e-commerce modelling behaviour of subscribers is a research interest field for quite a long time to gain insight into subscriber decision making (DM). They are also used by visiting online shopping sites to boost customer service and increase sales.

Recently, numerous online shopping portals use RS techniques [26, 31, 32] to create customized items that may satisfy the user's favourites/preferences. The CF [9, 34] and content-based (CB) techniques are widely utilized RS strategies. The CB method utilizes the user's history log data stored in the e-Commerce portal to extract user preferences. By and large, a subscriber's log information is constructed using the historical data of items procured or viewed by the subscribers or using comments. The ratings are given to items by the subscribers and so on. The collaborative filtering-based techniques [29] suggest that the items depend on the clients with identical preferences for a specific client. While the content-based techniques suggest the items that resemble similarities for items accessible in the subscriber's preference historical log data. In [19], a content-based collaborative filtering technique for fashion

✉ B. R. Sreenivasa
br.sreenu@gmail.com

C. R. Nirmala
nirmala.cr@gmail.com

1 Department of Information Science & Engineering, Bapuji Institute of Engineering & Technology, Visvesvaraya Technological University, Karnataka, Davangere, India

2 Department of Computer Science & Engineering, Bapuji Institute of Engineering & Technology, Visvesvaraya Technological University, Karnataka, Davangere, India

retail shopping environments was introduced. The model built recommendations using both online and offline data.

This model, however, does not produce a successful outcome over time. Also, the precise use of content-based and collaborative filters is limited by the accessibility of the subscriber's preferences' historical data. Those content-based and collaborative filtering techniques cannot provide legitimate/appropriate item advice to their subscribers if their historical preferential information is not accessible[4, 6].

Besides, personalized recommendation [24] is considered one of the most challenging tasks of the user rating system [37]. Customer intent/compliance is complex to predict/model. During the present and previous sessions, different internal and external features affect. Several state-of-the-art methods are running on a context-based recommendation system[27], investigating context data, including spatial position, temporal data [11], historical log data of the subscriber [30] or domain unique characteristics [28], using past user rating and query information employing reinforcement learning [37].

Similarly, numerous other refined neural network methods have been modelled, similar to time-based multi-task behavioural modelling [7, 33], and [42]. In [16], it is essential to consider both items and attribute session information to build a recommendation model. Further, they presented a Feature weighted session-based recommendation model that combines multiple attribute sessions using different feature weighting mechanisms. However, the model only considers a single session, and it is better to consider various sequences together [22] as it will give a better understanding of user preferences. Furthermore, deep learning (DL) approaches can be used to improve their conversion probability model. The RS method modelled in recent times [13, 14], and [38] show that the RNN-based RS model can outclass state-of-art other well-known options in some session-based forecasting undertakings.

The research aims to utilize both inter-session information of long-term context and intra-session of short-term context to build superior personalized time-centric RS. The following concerns and challenges must be addressed in order to develop such a model. First, a state-of-art Recurrent Neural Network (RNN) can't learn efficiently using extensive sequential information. Thus, it induces high memory usage and training latency. Then, the communication exchange information is composed of noise: as few clicks are done accidentally, few clicks are useful. Lastly, data from previous historical sequential sessions should assume various norms as the current session, yet there is no particular norm for combining session-based long-term information with session-centric short-term information. In this way, a progressively efficient method to deal with time-centric information with varied recommendation significance needs. This work considered the abovementioned challenge and issues in building a hybrid time-centric prediction model (HTCP) for the e-Commerce recommendation system.

The research contribution is as follows.

- This work modelled time-centric behaviour modelling for recommending items to its customers.
- We are designing a hybrid time-centric prediction model for modelling both short-term and long-term sequence information.
- The HTCP model outperforms the existing hybrid recommendation model in terms of HitRate and Mean Reciprocal Rank.

The rest of the paper is laid out as follows. Section II reviews the studies that have been used hybrid techniques in recommendations exclusively, which was the critical approach used in the existing systems. Section III then provides observations and discussions about the implemented HTTP model using the customer framework's dynamic behaviour trait. Conventional model performance metrics is presented. An experimental discussion is carried out in the penultimate segment. In the final section, we can see the conclusion and future work.

1.1 Literature survey

This section discusses the work done so far building efficient RS for the online shopping portal environment. Further, it discusses the techniques used, benefits, drawbacks, and limitations of these state-of-art RS models. Rating Matrix was used in RS for exploiting and extracting relationships among subscribers and the items for obtaining the subscriber's preferences in the current session (RM). Thus, a single outcome of RM depicts a user's preference outcome for particular item [14, 23, 26, 39], and [35]. For maintaining RM, the RS model can request the subscriber to rate an item explicitly.

Similarly, even the model can rate an item based on a subscriber's behaviour towards an item. In this way, RS, the matrix-factorization (MF) method, is used to deal with subscribers' favourite list or preferences within the RM [14, 23, 26, 39], and [35]. In [11], to build people's sequential actions, the author created a new LSTM (Long Short-Term Memory) updated version implemented in Time-LSTM. Time-LSTM trains LSTM with time gates to design time intervals. These time gates are deliberately built in such a way that Time-LSTM combines customers' short- and long-term wishes to improve the reliability of the recommendation compared to traditional RNN approaches.

Matrix factorization-based collaborative filtering strategies utilizes the rating matrix for learning subscriber's preferences. However, these techniques suffer from problems of CS and data sparsity issues [39]. If an item is not rated by any subscribers, it is difficult to recommend it to the users. This sort of issue is known as the cold start issue. Correspondingly, if barely any clients have given a rating to an item, or a client has given a rating to a few things, at that point, there is less information for carrying out forecasting. This sort of issue is described as a data sparsity issue. Subsequently, for overcoming the research issues, some work has been modelled using content-based strategies that utilize the content of the items or the metadata alongside the RM for item forecasting [39]. The author [8] presented RS using collaborative filtering and gated recurrent network [5] to learn subscriber favourite lists. The paper [26] proposed an RS using collaborative filtering alongside a gated recurrent network [5] for distinguishing subscriber behaviour from the current sessions. While [5] utilized the CF with gated recurrent unit learning user behaviour in recent sessions. The author[26] focused on a session-based recommendation and proposed a recurrent neural hierarchical network to transfer cross-session knowledge. A model-based hierarchical RNN (HRNN) that extends previous RNN-based session modelling by an additional GRU stage, predicting user behaviour through sessions and time transformation to predict personalized session-based recommendations. HRNNs offer a streamlined means of transferring information on long-term customer interest patterns to the session level and thereby delivering personalized user-oriented session-based suggestions.

In [26] and [5], authors have demonstrated that utilizing the gated recurrent network method handled sequential information from the current session superior to the other deep learning-based approaches. Existing session-based recommendation techniques use the metadata of the items click of the subscriber for forecasting items in the current session; consequently, these strategies don't consider the historical information of the client's traits from the recent sessions [5, 26]. The work states that item forecasting accuracy can be enhanced by efficiently learning the short-term feature sets (i.e., hidden items feature sets) from the client's behavioural information in the current sessions. This work presents a hybrid time-centric recommendation model for extracting the essential features from the customer's behaviour to overcome the research problem.

In[42], a multi-task learning system was suggested to predict consumer return times and jointly recommend products. A survival analysis model is used to assess patients' probability of survival to determine the return time. The authors have extended this model to predict customer return times using LSTM. LSTM also makes product recommendations from the previous customer session activities. This work aimed at presenting intersession proposals instead of the aforementioned session-based suggestions that concentrated on the advice from the same session.

2 Summary of the literature

Most techniques proposed for session recommendations are based on some sequence learning in the literature; A recent survey of the broader class of sequence-aware recommenders was found in [44]. Early techniques placed more importance on identifying recurrent sequential patterns to forecast a user's future behaviour at the time of the recommendation. These early algorithms were used in the context of forecasting user-online navigation behavior. Pattern mining approaches are then used to next-item suggestion problems in e-Commerce and the music industry [45]. While frequent pattern approaches are simple to use and result in comprehensible models, the mining process can be computationally intensive. Finding acceptable algorithm parameters, especially a reasonable minimum support threshold, might be difficult at the same time. Finally, it appears that in some application domains, employing frequent item sequences does not produce better recommendations than employing simpler item co-occurrence patterns [45].

More advanced sequence learning algorithms that incorporate some type of sequence modeling have been presented in a number of recent papers. Markov Chain (MC) models are commonly used in such sequence modeling approaches[46].

Over the last few years, the quantity of research papers on deep learning-based recommendation systems has exploded. Since 2016, RecSys, the world's leading international conference on recommendation systems, has hosted monthly deep learning seminars. Deep learning can characterize non-linear correlations in data by employing non-linear activations such as ReLU, Sigmoid, and Tanh. This characteristic enables the capture of sophisticated and complex user-item interaction patterns. Factorization machines and matrix factorization, for example, are inherently linear models.

RNNs are used in the most current efforts on sequence modeling. For example, Zhang et al.[37] employed them to forecast user actions in an advertisement situation. Hidasi et al. [14] were among the first to investigate GatedRecurrentUnits (GRUs), a type of RNN that may predict the next user action in a session. In [13][44]and [47] all enhanced their gru4rec approach in different ways. Approaches based on recurrent neural networks (RNNs) are the most recently studied group of approaches for session-based recommendation challenges. gru4rec is a brand-new deep learning algorithm explicitly designed for session-based recommendation applications [14][47].gru4rec uses an

RNN with GatedRecurrentUnits to model user sessions to forecast the likelihood of succeeding events (e.g., item clicks) given a session starting point.

2.1 Memory usage and training latency

The neural networks must store input data, weight parameters, and activations in their memory to transport data across the network. The activations from the forward pass must be retained during training before the error gradients in the reverse pass can be computed. The 50-layer ResNet network, for example, has 26 million weight parameters and calculates 16 million forward activations. If you have a floating-point value of 32 bits to store every weight and activation, that will provide 168 MB of storage. We may reduce this storage requirement by half or a quarter by adding the lower precision value to store these weights and activations.

It is impossible to hold the GPU processor's data with such large quantities of storage state required. In reality, many high-performance GPU processors only have 1 KB of memory per core processor that can be read quickly enough to saturate the floating-point datapath. This means that it must save the state in an external DRAM at each layer of the DNN, load up the next network layer, and load the data back into the device. Therefore, the already limited off-chip memory interface bandwidth and latency suffer from the added pressure of continuous recharge and re-activation. This decreases training time substantially and increases power demand considerably.

Although parallelism in large mini-batches improves computational efficiency, research demonstrates that large mini-batches result in generalizing networks that take longer to train. Furthermore, machine learning model graphs already reveal a great deal of parallelism.

During decades of compilation work for sequential languages, many techniques are available to minimize the memory further. First, there should be "in-place" operations, such as activation functions, to transcribe input data directly from the output. This allows for the reuse of the memory state. Second, the data dependencies between network activities can be reused by evaluating and assigning the same memory to operations that fail to use it simultaneously.

Researchers from Google DeepMind with recurring neural networks have established a similar memory-reuse approach (RNNs). RNNs are a special form of DNN that has a structure that encodes behavior cycles over input sequences. Re-computing for RNN's has shown that the memory is reduced by 20 for 1000-long sequences with only 30 percent overhead efficiency.

2.2 Proposed work

The goal of phase-1 of the research aims to create a recommendation model based on the client's geographical location. Customers' dynamic behavior traits were taken into account when developing the hybrid location-centric e-Commerce recommendation model. The work considers the dynamic behavior of the user and learns both short-term and long-term context. The RNN model is described in detail in HLCP and how RNN is used to model the long-term context. When people/users are exposed to a new environment, their behavior changes quickly (short-term). The priority of items may vary when a user goes to a different country or place.

Similarly, when certain advertising is posted to a specific user interface, user preferences change quickly (AUI). As a result, the RNN model can't be used to describe such a short-term scenario. The paper provides the operational structure of the feed-forward neural network for modeling short-term context in dynamic behavior features of the user in order to address the challenges.

In addition, the research includes static behavior modeling, such as a customer viewing a product list and spending some time on an e-Commerce website. On an e-Commerce website, more dynamic behavior was also considered, such as user clicks, views, purchases, spending time, and so on. The dynamic behavior model is then exhibited in the HLCP model. Finally, the Bayesian personalized ranking is utilized to train a recurrent neural network to predict e-Commerce customers' behavior traits.

In phase-2, the research work provides a hybrid time-centric e-Commerce recommendation model based on the dynamic behavioral traits of the customer. The architecture of the HTCP model is shown in Fig. 3. The HTCP model is designed for recommending items based on users' dynamic behavior and session-centric information on the ongoing session. For obtaining a relationship between user current sessions and past behavior for recommending items, it is essential to model both short and long-term behavior contexts (i.e., time-centric HLCP model). The hybrid time-centric prediction (HTCP) model is designed by substituting location-centric (LC) transition matrices (TM) with session-centric TM's.

3 Hybrid time centric recommendation model for online shopping environment

The HTCP model recommends items based on the user's dynamic behaviour and session-centric information on the ongoing session. For obtaining a relationship between user current session and past behaviour for recommending items, it is essential to consider both short-term and long-term behaviour context (i.e., Location centric HLCP model [36]). Let's consider a set of items like clothes, furniture, computer accessories. as follow

$$\mathcal{U}\{u_1, u_2, \ldots, u_n\}, \tag{1}$$

where u_n. No._of_items and set_of_users like a male, female, chilen, etc. as follows

$$\mathcal{V} = \{u_1, u_2, \ldots, u_n\}. \tag{2}$$

where u_n. total number of users. The proposed work considers e-Commerce application, which includes different behaviour such as

$$\mathcal{C} = \{c_1, c_2, c_3, c_4\}. \tag{3}$$

where c_1 clickstream, c_2 items_added to cart list, c_3 adding to favourite_list, and c_4 purchased items. The job is to predict what clients will buy in the current session using the hybrid time-centric prediction model.

3.1 Long-term and short-term behaviour modelling

For modelling the long-term behaviour context, this work employs RNN. The architecture of RNN is shown in Fig. 1

The RNN architecture comprises an input layer, numerous hidden layers, output layers, and inner weight matrices. The activation parameter of the hidden layers is obtained as follows

$$i^v_\ell = f\left(\mathcal{X}i^v_\ell + \mathcal{D}s_{w^v_\ell}\right), \tag{4}$$

where $i^v_\ell \in \mathbb{S}^e$ illustration of user v at ℓ. in a trait, $\mathcal{D}_{j^v_\ell} \in \mathbb{S}^e$. ℓ^{th}. input item of user v.. The activation parameter is represented by $f(i)$ and the present item transition matrix is represented as follows

$$\mathcal{D} \in \mathbb{S}^e \tag{5}$$

$\mathcal{D}$. can obtain the user's present behaviour, and $\mathcal{X}$ can propagate traits signals. Equation (4) is executed iteratively to get the status of each location in traits. The hidden layer information of the RNN is dynamic for behavioural characteristics, wherein the pattern is repetitive. Thus, RNN faces issues in adapting short-term contexts in behavioural traits.or learning short-term behavioural context productively, this work utilizes a feed-forward neural network (FFNN) with a solitary linear hidden layer[15, 20]. The usage of FFNN aid in learning recent behaviour more efficiently and improves behaviour classification accuracy [19]. The author's prior work [36] has additional details on the RNN and FFNN models depicted in Fig. 1 and Fig. 2. The architecture of FFNN is shown in Fig. 2.

Thus, behaviour traits in the ongoing session window are extracted using items clicked and their transition matrices in each session. The next position can be estimated using a linear prediction model using the following equation

$$i^u_\ell = \sum_{j=0}^{\alpha-1} \mathcal{D}_j s_{w^u_j} \tag{6}$$

where $\mathcal{D}_j \in \mathbb{S}^{e*e}$ transition-matrix for each location in a behaviour trait, and a number of components modelled in a trait.

To capture the consumer's complex behaviour, the proposed study considers behavior-based matrices to obtain characteristics of different forms of behaviour. Then, the illustration of subscriber $\sqsubseteq$ at $\updownarrow$ is estimated using the following equation

$$i^v_\ell = \mathcal{X}i^v_\ell + \sum_{i=0}^{\alpha-1} \mathcal{D}_j \mathcal{N}_{c^v_\ell - jr^v_\ell - j'} \tag{7}$$

where $\mathcal{N}_{c^v_{\ell-j}} \in \mathbb{S}^{e*e}$ behaviour-based transition matrix on the j^{th}. product of subscribe u. The cold-start problem can be addressed by considering $i^v_0 = v_0$.ething etial to be outwardly seen here is that the behaviour-based matrices can be eliminated if there is one kind of behaviour.LCP can get the underlying features of whether various types of behaviour in past traits.

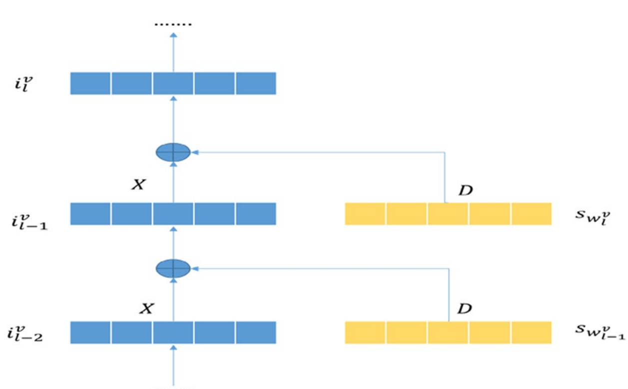

Fig. 1 The architecture of the RNN model

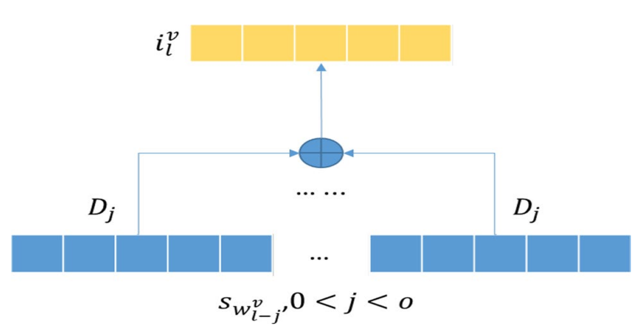

Fig. 2 The architecture of the FFNN model

Further, by computing whether user v. would perform behaviour c. on product w at session sequence $\ell + 1$ can be obtained as

$$z_{v,\ell+1,c,w} = \left(t_\ell^v\right)^{\mathcal{U}} \mathcal{N}_c s_w = \left(i_\ell^v + v_v\right)^{\mathcal{U}} \mathcal{N}_c s_w \tag{8}$$

where t_ℓ^v current_position of user v at the traits position ℓ, static hidden representation $v_v \in \mathbb{S}^e$ and containing a dynamic representation i_ℓ^v.

3.2 Time-centric behaviour modelling

The sequence-based model generally neglect continuous session variance (SV) among input feature sets. The session variance feature is handy in forecasting as short-term session window variances have typically higher effects on future buying than using long-term session variances. For example, let consider two product, w_a. and w_b., in a customer's procuring log. The customer purchased the product w_a the previous night or a few hours back and product w_b a few weeks back. There is a high chance that a customer's choice of product to purchase in the future may be impacted by product w_a. On the way around, if the product w_b are purchased the previous day, then there is a high probability that that product w_a. and w_b have an identical effect on customer's selection due to similar interest in a short-term instance. Besides, the behaviour of purchasing individual items is periodic. For instance, if you are buying soaps and shampoos on a monthly basis, the influence of session variances results in a more dynamic environment. Considering these conditions, we improve the HLCP model [36] by incorporating session variances knowledge and model time-centric prediction models.. 3. The architecture of HTCP: HTCP is modelled to capture both short-term and long-term context in behavioural traits.

3.3 Hybrid time-centric prediction-based recommendation system

In the above, this work discussed how user preference changes when they move to different regions/locations. For learning the user's preference HLCP model is presented in [36]. However, it is still essential to consider and incorporate time/session variance data into HLCP. Thus, this work presents a hybrid time-centric predictionHTCP) model by substituting location-centric (LC) transition matrices (TM) with session-centric TM's. The HTCP model is shown in Fig. 3. From Fig. 3, for a given customer v., the location l is computed as follows

$$i_l^v = X i_{l-o}^v + \sum_{j=0}^{o-1} U_{u_l^v - u_{l-1}^v} s_{w_{l-j}^v}, \tag{9}$$

where u_l^v. present time,u_{l-1}^v. time of every producof eacler oHTCP, and $U_{u_l^v - u_{l-1}^v}$. time-centric TM of time variance $u_{l-1}^v - u_l^v$ among time u_{l-1}^v. and u_l^v. The time-centric TM helps in capturing session-specific behaviour based on the most recent activity log. Further, Eq. (9) is rewritten similarly with the HLCP model as follow

$$i_l^v = X i_0^v + \sum_{j=0}^{l-1} U_{u_l^v - u_{l-1}^v} s_{w_{l-j}^v} \tag{10}$$

where $i_l^v = v_0$., depicting the initial condition of customers. For modelling dynamic behavioural traits, behavioural-centric TM's is used in the HTCP model as follows

$$i_l^v = X i_{l-o}^v + \sum_{j=0}^{l-1} U_{u_l^v - u_{l-1}^v} N_{c_{l-m}^v} s_{w_{l-j}^v}. \tag{11}$$

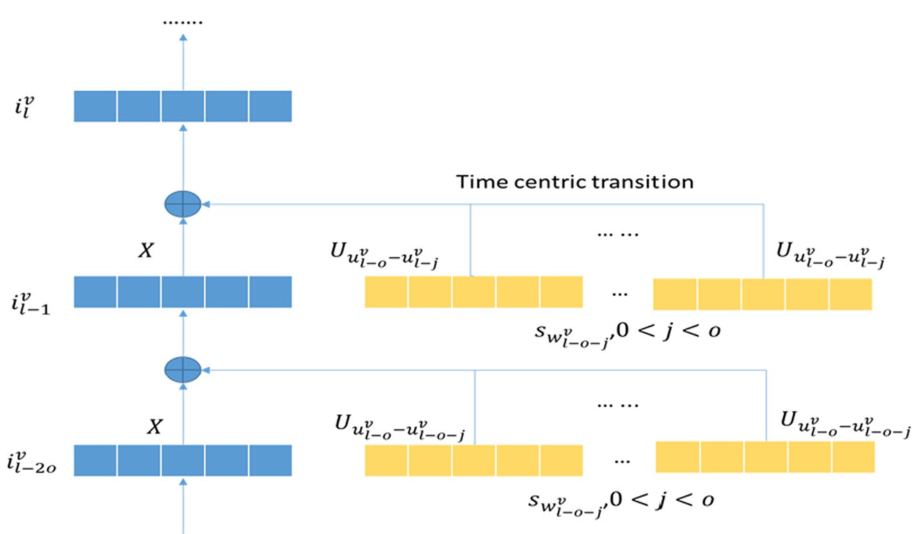

Fig. 3 The architecture of HTCP: HTCP is modelled to capture both short-term and long-term context in behavioural traits

Then, carry out forecasting operation whether a customer v. will carry out certain behaviour c. on particular product w. at sequential location $l+1$ is estimated similarly to HLCP using the following equation

$$z_{v,l+1,c,w} = \left(t_l^v\right)^U N_c s_w = \left(i_l^v + v_v\right)^U N_c \tag{12}$$

3.4 Transition matrix learning

Suppose we have to continuously learn unique conceivable session variance, then we must compute a higher number of session-centric TM's and result inverfitting problems. Thus, it is essential to segment all probable session variance parameters equally into a separate window for addressing the issue. This work considers estimating only TM's of the lower and upper limits of the session window (SW). Using linear interpolation (LI) for all session variance in SW, their TM's are computed. The time-centric TM U_{ue}. for session variance parameter u_e is mathematically expressed as follows

$$U_{ue} = \frac{\left[U_{M(u_e)}\left(V\left(u_e\right) - u_e\right) + U_{V(u_e)}\left(u_e - M\left(u_e\right)\right)\right]}{\left[\left(V\left(u_e\right) - u_e\right) + \left(u_e - M\left(u_e\right)\right)\right]}, \tag{13}$$

where $M\left(u_e\right)$. and $V\left(u_e\right)$. depicts the lower and upper limit of session variance u_e., $U_{V(u_e)}$. and $U_{M(u_e)}$. depict the session-centric TM's for $V\left(u_e\right)$. and $M\left(u_e\right)$, respectively. Using LI, we can address the problem in learning time-centric TM for continuous session variances. Note that optimizing time-centric TM's in every separate session window is linear. However, the globptimization ithe comprehensive range of every probable session variance is non-linear. The HLCP model is efficient in modelling the sequence of behavioural traits of the user.

Fthermore, when the user's location changes, the HLCP model is particularly efficient. This is because the HLCP model stores the entirety of the user's behaviour in a single session window. However, for modelling session-centric data, the HTCP model outperforms HLCP since the session is partitioned into multiple-session windows in HTCP. The session windows are then compared to one another. As a result, the HTCP model is useful when time-centric information is provided; otherwise, HLCP is more efficient.

3.5 Learning of TC-LCP model

The Bayesian Personalized Ranking (BPR) model [30] is a pairwise ranking method used for implicit feedback. It is an objective parameter widely used to learn RNN and for predicting customer behaviour traits. In general Bayesian personalized ranking considers that a customer desires a chosen set than a negative one. That is, it aims to maximize the probability using the below equation

$$p\left(v, l+1, c, w \succ w'\right) = h\left(z_{v,l+1,c,w} - z_{v,l+1,c,w'}\right) \tag{14}$$

where w' negative features, $h(y)$. is a non-linear function selected using the below equation

$$h(y) = \frac{1}{1 + e^{-y}}. \tag{15}$$

Thus, by considering negative log-likelihood, the fourth coming objective function can be minimized equivalently as follows

$$K_1 = \sum \log\left(1 + e^{-\left(z_{v,l+1,c,w} - z_{v,l+1,c,w'}\right)}\right) + \frac{\mu}{2}\|\ominus\|_1^2, \tag{16}$$

where μ. regularization power control parameter and $\ominus_1 = \{V, S, X, U, N\}$ depicts parameter to be computed. The experiment is conducted to evaluate the performance outcome of HTCP in terms of Hit Rate (HR) and Mean Reciprocal Rank (MRR) performance.

4 Inter and Intra-session RNN

Intra-session RNN generates suggestions by analyzing the order in which items are presented in a session. A large amount of memory can be used to create a mini-batch of one hot-vectors when the collection of items is extensive, which may be a problem and dropout is applied to these levels once the embedded item representation is transmitted through one or more levels of GRU. The vector is then scaled up to R|N| using a feed-forward layer. The result is [ov1,ov2…ov|N|], where ov_i is the score for item vi € N. Rj_is then developed a list of recommendations by sorting things belonging to the k highest scores by their score. Although the RNN intra-session can attain high efficiency, it begins at any session without any user knowledge. In the session, the model will learn about the user's needs, but all data will once again be discarded at the end of the session.

Long-term user profiles are included for personalized session-based RS to learn the intersession pattern in a smooth manner by using a temporal dynamics model. User embedding, short-term interest, user taste evolution, user survival time, and local negative sampling are integrated into the temporal changes in session RNN. The inter_session RNN will boost the intra-session RNN because it considers the user's previous sessions, and at the beginning of each session, it provides information to the intra-session feature.

We differentiate the above two kinds of session contexts because they require different types of relationships for the recommendation task: the context of intra-session embeds the dependency of the intra-session. At the same time, inter-session primarily conveys the inter-session dependency.

5 Result and discussion

This section presents the experimental analysis of the HTCP model over the existing hybrid recommendation model [18]. The existing hybrid recommendations [18] are designed by combining both RNN and k-nearest neighbour algorithms using static customer behavioural information [41], and also GRU-based methods are used for comparison. On the other hand, the hybrid recommendation models combine both FFNN and RNN using dynamic customer behaviour. The RNN model is then learned by maximizing objective function using BPR [30] [12].

Inter-session and intra-session approaches are used in this study to improve the accuracy of the proposed work in comparison to existing methodologies. The user's actions in the time-centric prediction model may be influenced by all past activities in the session, not just the earlier one. Session-based recommendations benefit from RNNs' ability to process user activity sequences and internally represent the user's interests. It also doesn't imply that all actions indicate a willingness to learn more about anything. It can learn to recognize other people's indifferent behaviour. Short-term dependencies between actions within a session, as well as long-term dependencies between actions from distinct sessions, are prevalent. For example, a user who was interested in research papers in their prior session(s) is likely to be interested in them again in their present session. Similarly, a user who purchased a new gadget in a previous session is unlikely to purchase another in the present session, although they may be interested in accessories for the gadgets they purchased. For instance, a user who has shown an interest in research articles during his/her previous session(s) is likely to express that interest again during his/her current session. Likewise, a user who acquired a new gadget during a prior session is unlikely to purchase another during this session, albeit he/she may be interested in accessories for the gadgets he/she purchased. Thus, if the user's behaviour does not change in response to the location, the model employs RNN and operates efficiently through the use of historical data.

However, humans have a tendency to shift their behavioural patterns when exposed to different environments such as location, climate, and even in this pandemic circumstance, the user's surfing patterns change dramatically. Existing RNNs struggle to learn unexpected changes in the user's behavioural patterns since the RNN's hidden layer data is complex in terms of behavioural features, especially when the pattern is recurrent.

Table 1 Characteristics of the Yoochoose datasets

Dataset Behaviours	Yoochoose Dataset
Clicks Streams	31,637,239
Sessions	7,996,257
Items	37,483
Timespan in days	182
Actions per sessions	3.97
Action per Day	174,222
Session per Day	43,854

Table 2 Characteristics of the TMall datasets

Dataset Behaviours	TMall Dataset
Clicks Streams	13,421,239
Sessions	1,776,154
Items	25,348
Timespan in days	91
Actions per sessions	7.56
Action per Day	149,096
Session per Day	19,719

To efficiently learn short-term behavioural contexts, the work employs a Feed-Forward Neural Network (FFNN) with a single linear hidden layer. For instance, if a person is always engaged in browsing, his or her smartphone's browsing habits may change as the person changes its locations. In this case, recommendations would be generated solely on the basis of current intrasession browsing patterns, as advising on the basis of previous sessions' data would diminish the model's accuracy. In this work, the Feed Farword neural network is employed to successfully interpret the user's short-term behavioural context.Then the intrasession information will be stored in the RNNs memeory for better recommendations.

5.1 Dataset characteristics

First, the experiment is conducted on the Yoochoose dataset used in the Recsys Challenge 2015 [2], as shown in Table1. The Yoochoose dataset contains a collection of a sequence of clickstreams (click sessions) from yoochoose.com. Given a sequence of click streams, the objective is to foresee what things the client will purchase, if any. Such data is profoundly significant to e-businesses since it can indicate what stuff to recommend to the client and urge them to turn into a purchaser. The trial information generally contains 31,637,239 clicks on 37,483 items and involving 7,996,257 sessions gathered from yoochoose.com. This dataset includes the clicks of a customer (such as session id, timestamp, product id, and category) and the date of purchase of a customer (session-id, timestamp, product id, price, and quantity).

Second, the experiment is conducted on the TMall data set [10], and its characteristics are shown in Table2. The recommendation model's preliminary job is to forecast what the customer will buy in the current ongoing session using time-centric prediction and the existing recommendation model. The TMall dataset comprises two lakhs of shopping records composed of thousands of users considering twenty-five thousand products. It is one of china's biggest e-commerce web portals. The HTCP utilizes both the dataset to predict the items a client will purchase in the progressing session. These datasets contain redundant data. Thus in preprocessing steps, redundant data are removed for experiment analysis. Then, we establish the session information using clickstreams. Also, utilize the data from each customer's previous session as the testing data, and the remaining session information is used to train the model. The experiment is conducted on the Yoochoose and TMall dataset to evaluate the performance of HTCP over the existing hybrid forecasting model in terms of Precision/Mean reciprocal rate (MRR) and Recall/Hit rate (HR). (Tables 3 and 4).

5.2 Baselines methods used

5.2.1 Gated Recurrent Unit(GRU4Rec)

gru4rec is a new deep learning methodology that was created with session-based recommendation scenarios in mind [13][14]. gru4rec uses an RNN with GateRecurrentUnits to model user sessions to forecast the probability of subsequent events (e.g., item clicks) given a session start. To learn the model, it employs a method of session-parallel mini-batch training and frequently employs ranking-based loss functions.

5.3 Sequential Rules (SR)

In [15] suggested a sequential rule method that is a version of Markov Chains (MC) or Association Rules (AR). It considers the order of events as well, but in a less restrictive way. We generate a rule when an item-M appears after an item-N in a session, even if additional events occur between M and N, unlike the MC method. We examine the number of elements appearing between p and q in the session when allocating weights to the rules. We employ the weight function $wsr(x) = 1/(x)$, where x denotes the number of steps between the two objects.

5.3.1 Session-based Matrix Factorization (SMF)

Session-Based Matrix Factorization is a novel factorization-based method explicitly built for the purpose of session-based recommendation.

It combines Factorised Markov Chains with classical matrix factorization, similar to Factorised Sequential Prediction with Item Similarity Models (FOSSIL)[18]. In the smf technique, we replace the latent user vector M_u with a session preference vector s_e,in contrast to the conventional factorization-based prediction model $r_{u,i} = M_u N_i^T$.

5.3.2 Session-based kNN (SKNN)

Rather than just looking at the last event in the current session, the sknn methodology compares the current session to past sessions in the training data to identify the items to

Table 3 Results comparison with proposed work and existing works in terms of HR and MRR for Yoochoose dataset

Algorithm	Metrics MRR@20	Metrics MRR@10	Metrics HR@20	Metrics HR@10
HTCP	0.394	0.308	0.783	0.614
gru4rec (ER)	0.308	0.301	0.683	0.591
Sequential Rule (SR)	0.304	0.298	0.653	0.569
Session Based Matrix Factorization (SMF)	0.302	0.295	0.666	0.575
Sequential session based KNN(S-SKNN)	0.272	0.267	0.602	0.531

Table 4 Results comparison with proposed work and existing works in terms of HR and MRR for TMall dataset

Algorithm	Metrics MRR@20	Metrics MRR@10	Metrics HR@20	Metrics HR@10
HTCP	0.298	0.197	0.534	0.457
gru4rec (ER)	0.129	0.125	0.277	0.225
Sequential Rule (SR)	0.128	0.125	0.242	0.206
Session Based Matrix Factorization (SMF)	0.121	0.118	0.261	0.213
Sequential session based KNN(S-SKNN)	0.185	0.181	0.387	0.331

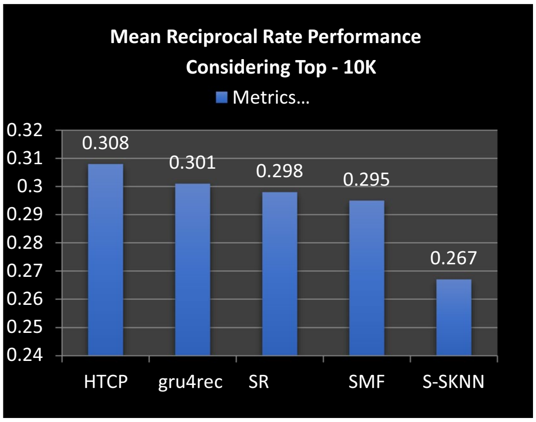

Fig. 4 Average MRR performance attained by the HTCP considering the top-10 k recommendation for Yoochoose Dataset

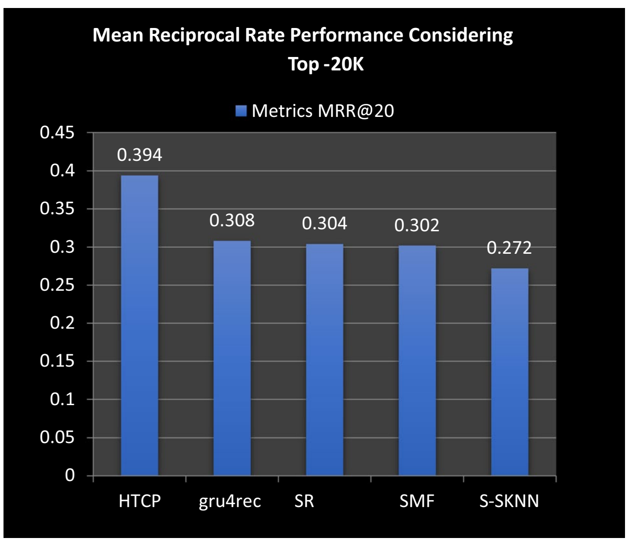

Fig.5 Average MRR performance attained by the HTCP considering the top-20 k recommendation for Yoochoose Dataset

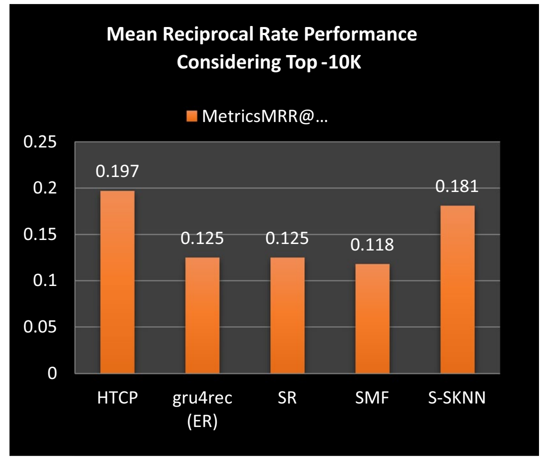

Fig.6 Mean reciprocal rate performance considering the top-10 k recommendation for TMall Dataset

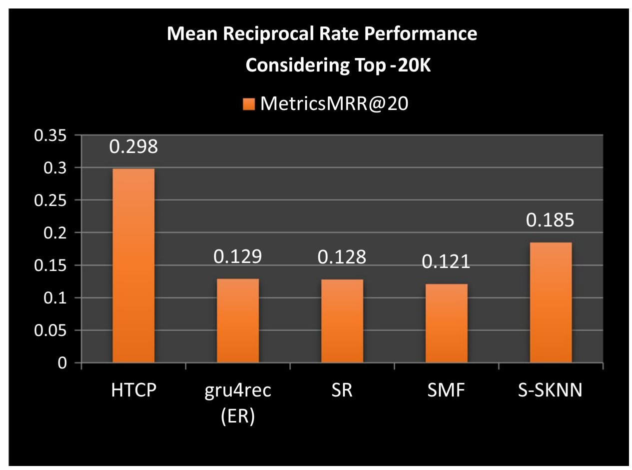

Fig.7 Mean reciprocal rate performance considering the top-20 k recommendation for the TMall Dataset

recommend(see also [45] and [18]). Technically, given a session s, we find the k most similar previous sessions (neighbors) Ns by smearing a suitable session similarity measure to binary vectors over the item space, such as the Jaccard index or cosine similarity [45].

5.3.3 Mean Reciprocal Rate/Precision performance

The accuracy performance of both the HTCP model and the existing model is evaluated on the Yoochoose dataset in MRR for a top-10 k recommendation top-20 k recommendation considering the Yoochoose data set. Figure 4 shows that the HTCP achieves an MRR outcome of 0.308, and the ER method achieves an MRR outcome of 0.301, considering the top-10 k recommendation. Similarly, the HTCP achieves an MRR outcome of 0.3945, and the existing method achieves an MRR outcome of 0.308, considering the top-20 k recommendations. The result attained in Figs. 4 and 5 shows that the HTCP model attains significant improvement in the performance over the existing baseline model in terms of MRR considering the Yoochoose dataset. Further, the experiment is conducted using the TMall dataset to evaluate the MRR performance of HTCP and the existing recommendation (ER) model. The accuracy outcome is evaluated considering the top-10 k recommendations, as shown in Fig. 6. The HTCP achieves an MRR outcome of 0.1978, and the existing research method, an MRR achieves an outcome of 0.0986. The result attained shows the HTCP method achieves a much superior MRR outcome than existing approaches, considering the top-10 k recommendation.

Further, the accuracy outcome is evaluated considering top-20 k recommendations, as shown in Fig. 7. The HTCP achieves an MRR outcome of 0.2984, and the ER method achieves an MRR outcome of 0.1852. The result attained shows the HTCP method achieves a much superior MRR outcome than ER methodologies, considering the top-20 k recommendation. The overall result attained considering the top-10 k recommendation and top-20 k recommendation considering both Yoochoose and TMall dataset shows HTCP achieves superior MRR outcomes. HTCP achieving better MRR will aid in improving better profitability for the e-Commerce environment.

5.3.4 Hit Rate performance

The performance evaluation based on both the HTCP model's accuracy and the existing hybrid recommendation model is evaluated considering the Yoochoose dataset in HR for a top-10 k recommendation and top-20 k recommendation shown in Figs. 8 and 9. Figure 8 indicates that the HTCP achieves a HitRate outcome of 0.6145, and the ER method achieves a HitRate outcome of 0.155, considering the top-10 k recommendation. Similarly, the HTCP achieves a HitRate outcome of 0.7832, and the ER method achieves a HitRate outcome of 0.6827, considering the top-20 k recommendations. The result attained in Figs. 8 and 9 shows that the HTCP model attains significant improvement in the performance over the existing baseline model in terms of HR considering the Yoochoose dataset. Further, the experiment is conducted using the TMall dataset to evaluate the MRR performance of the HTCP and the existing recommendation (ER) model. The accuracy outcome (i..e, HitRate) of HTCP and existing recommendation methodologies are evaluated in this section. The accuracy outcome is evaluated considering top-10 k recommendations, as shown in Fig. 10. The HTCP achieves a HitRate outcome of 0.4571, and ER (i..e, ES (existing system)) method achieves an HitRate outcome of 0.2119. The result shows that the HTCP method achieves a much superior HitRate outcome than ER methodologies, considering the top-10 k recommendation.

Further, the accuracy outcome is evaluated considering top-20 k recommendations, as shown in Fig. 11. The HTCP achieves a HitRate outcome of 0.5347, and the ER method achieves a HitRate outcome of 0.4038. The result shows that the HTCP method achieves a much superior HitRate outcome than ER methodologies, considering the top-20 k recommendation. The overall result attained considering the top-10 k recommendation, and top-20 k recommendation shows HTCP achieves superior HitRate outcomes considering two dynamic behaviour datasets such as Yoochoose and TMall dataset. Higher HitRate outcome of the HTCP model will help achieve better recommendations for the user and

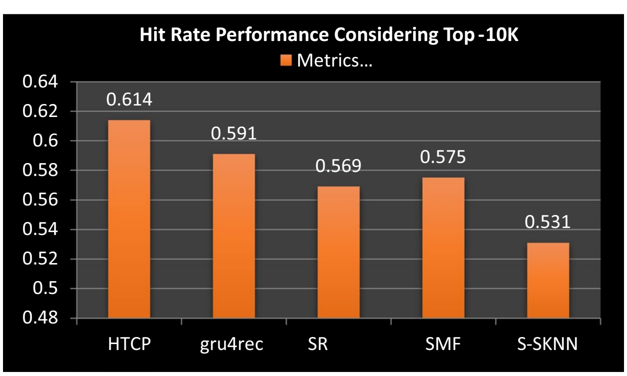

Fig. 8 Average HR performance attained by the HTCP considering the top-10 k recommendation for Yoochoose Dataset

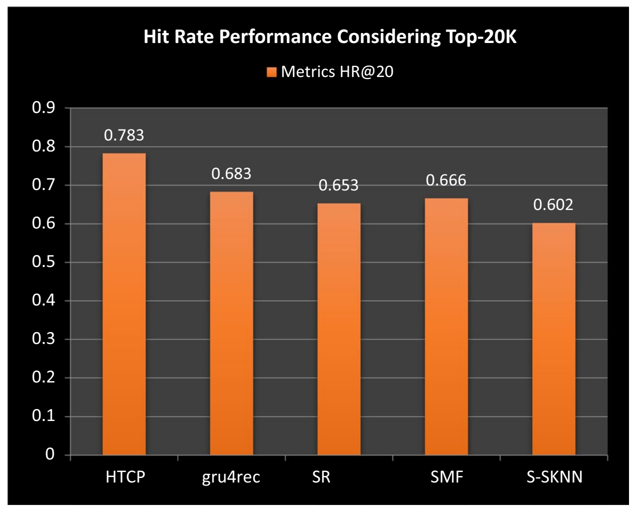

Fig. 9 Average HR performance attained by the HTCP considering the top-20 k recommendation for Yoochoose Dataset

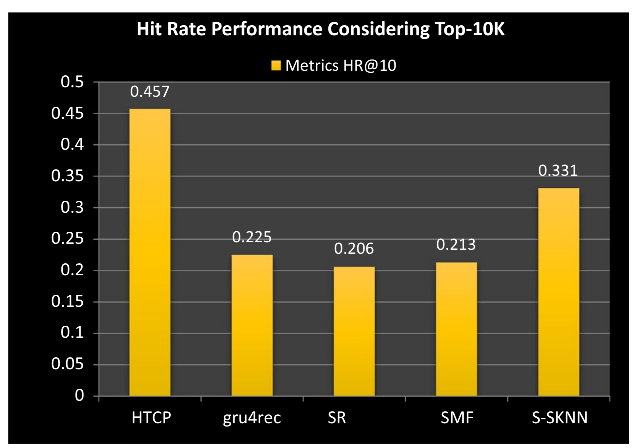

Fig. 10 Hit rate performance considering the top-10 k recommendation for the TMall Dataset

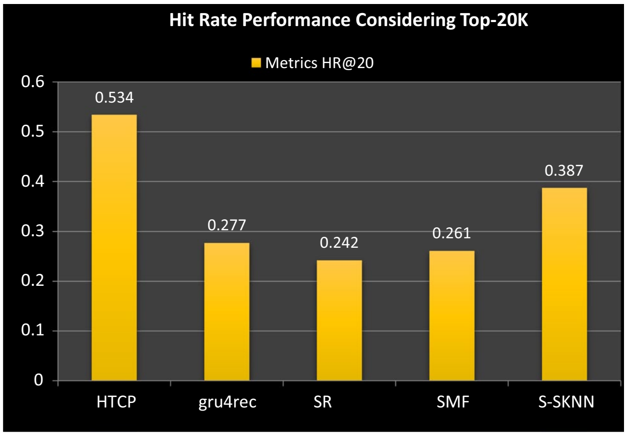

Fig. 11 Hit rate performance considering the top-20 k recommendation for the TMall Dataset

increase the quality of subscribers' experience aiding better profitability of the online shopping portal.

6 Conclusion

The research first explores how creating an effective session-based recommendation framework increases the online shopping portal's conversion rate. The comprehensive study shows that RNN is commonly used in numerous current session-based recommendation models with good performance. However, they are not effective for modelling the subscriber's short-term behaviour sequences. It has been presented to boost the efficiency of the hybrid model using RNN. Using FFNN as an attenuation layer for weight optimization in RNN has helped build a better recommendation model. However, the existing hybrid model is not productive for modelling the present ongoing session's suggestions that end in purchases.

To address research issues, the research work demonstrates the usefulness of using hybrid forecasting to combine RNN and FFNN to predict user choices in the current session. It is shown by combing RNN and FFNN, and the hybrid model is sufficient to reduce the issue of cold start and data sparseness in the time-centric recommendation method, which is experimentally proved by experimenting with a better outcome than the existing recommendation models. HTCP has helped solve data sparsity issues using the session window where the model will suggest items based on previous session sequences even though a specific sequence is missing—further, modelling the dynamic behaviour of customers aid in improving the learning efficiency of the time-biased feature of the user's interest in the particular item for certain session intervals. This aids in extracting significant item feature sets for a user in the present session.

The experiment is conducted on Yoochoose and the TMall dataset to evaluate the performance accuracy of implemented HTCP over the existing hybrid recommendation model. The results show that, respectively, HTCP and existing hybrid models for the yoochoose dataset achieve the hit-rate efficiency of 0,6145 and 0,595 for top-10 K recommending and 0.7832 and 0.6827 for a Top-20 k recommendation. Also, an average mean reciprocal rate performance of 0.308and 0.301 for top-10 k recommendations and 0.3945 and 0.308 for top-20 k recommendations is attained by HTCP and the existing hybrid model.

The result shows that the Hit Rate performance of 0.4571and 0.3312 for top-10 k recommendation is attained by the HTCP and existing hybrid model for the TMall dataset. Similarly, for a top-20 k recommendation, an average hit rate performance of 0.5347 and 0.3871is attained by HTCP and existing hybrid models for the TMall dataset, respectively. Furthermore, an average mean reciprocal rate performance of 0.1978 and 0.185 for top-10 k recommendations and 0.2984 and 0.1852 for top-20 k recommendations is attained by HTCP and the existing hybrid model for the TMall dataset, respectively.

The overall result shows that significant performance achieved by HTCP over the existing hybrid recommendation model in terms of MMR and HR. Future work would consider performance evaluation considering varied datasets such as Amazon, Movie Lens, etc. Along with would carry out comparative analysis over various state-of-art methods.

Declarations

Conflict of interest I declare that I am the sole author and exclusive owner of my work and declares that the following contribution has No conflict of interest with any commercial product or service related to the submitted article.

References

1. Aaron Smith MA (2016). Online Shopping and E-Commerce. In: Pew Research Center, December, 2016, "Online Shopping and E-Commerce.
2. Ben-Shimon D, Shapira B, Tsikinovsky A, Rokach L, Friedmann M, Hoerle J (2015) RecSys Challenge 2015 and the YOOCHOOSE Dataset. In: RecSys 2015 - Proceedings of the 9th ACM Conference on Recommender Systems (pp 357–358). Association for Computing Machinery, Inc. https://doi.org/10.1145/2792838.2798723
3. Campos PG, Díez F, Cantador I (2014) Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. User Modell User-Adapt Interact 24:67–119. https://doi.org/10.1007/s11257-012-9136-x

4. Cheng HT, Koc L, Harmsen J, Shaked T, Chandra T, Aradhye H et al. (2016) Wide & Deep Learning for Recommender Systems. In: DLRS 2016: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (Vol. 15-Septemb, pp 7–10). https://doi.org/10.1145/2988450.2988454
5. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modelling. In: Neural and Evolutionary Computing (pp 1–9). http://arxiv.org/abs/1412.3555
6. Covington P, Adams J, Sargin E (2016) Deep neural networks for youtube recommendations. In: RecSys 2016 - Proceedings of the 10th ACM Conference on Recommender Systems (pp 191–198). Association for Computing Machinery, Inc. https://doi.org/10.1145/2959100.2959190
7. Dai H, Wang Y, Trivedi R, Song L (2016) Recurrent coevolutionary latent feature processes for continuous-time recommendation. In: DLRS 2016: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (pp 29–34). Association for Computing Machinery. https://doi.org/10.1145/2988450.2988451
8. Donkers T, Loepp B, Ziegler J (2017) Sequential user-based recurrent neural network recommendations. In: RecSys 2017 - Proceedings of the 11th ACM Conference on Recommender Systems (pp 152–160). https://doi.org/10.1145/3109859.3109877
9. Fu M, Qu H, Yi Z, Lu L, Liu Y (2019) A novel deep learning-based collaborative filtering model for recommendation system. IEEE Transa Cybernet 49(3):1084–1096. https://doi.org/10.1109/TCYB.2018.2795041
10. Gao C, Jin D, He X, Gan D, Chen X, Feng F et al (2019) Learning to recommend with multiple cascading behaviours. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/tkde.2019.2958808
11. Ge H, Caverlee J, Lu H (2016). TAPER: A contextual tensor-based approach for personalized expert recommendation hancheng. In: RecSys 2016 - Proceedings of the 10th ACM Conference on Recommender Systems (pp 261–268). https://doi.org/10.1145/2959100.2959151
12. He Y, Wang C, Jiang C (2019) Correlated matrix factorization for recommendation with implicit feedback. IEEE Trans Knowl Data Eng 31(3):451–464. https://doi.org/10.1109/TKDE.2018.2840993
13. Hidasi B. zs, Karatzoglou A (2018) Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In: CIKM '18: Proceedings of the 27th ACM International Conference on Information and Knowledge Management (pp 843–852). https://doi.org/10.1145/3269206.3271761
14. Hidasi B, Karatzoglou A, Baltrunas L, Tikk D (2016) Session-Based Recommendations with Recurrent Neural Networks. In: 4th International Conference on Learning Representations, ICLR 2016 (pp 1–10). http://arxiv.org/abs/1511.06939
15. Fang H, Guo G, ZhangS DY (2019) Deep learning-based sequential recommender systems: concepts, algorithms, and evaluations hui. Deep Learn-Based Sequential Recommend Syst 577(11496):574–577. https://doi.org/10.1007/978-3-030-19274-7
16. Hwangbo H, Kim Y (2019) Session-based recommender system for sustainable digital marketing. Sustain (Switzerland) 11(3336):1–19. https://doi.org/10.3390/SU11123336
17. Hwangbo H, Kim YS, Cha KJ (2018) Recommendation system development for fashion Retail e-commerce. Electron Commer Res Appl 28:94–101. https://doi.org/10.1016/j.elerap.2018.01.012
18. Jannach D, Ludewig M, Lerche L (2017) Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. User Modell User-Adapt Interact 27(3–5):351–392. https://doi.org/10.1007/s11257-017-9194-1
19. Kang WC, McAuley J (2018) Self-attentive sequential recommendation. In: Proceedings - IEEE International Conference on Data Mining, ICDM (Vol. 2018-Novem, pp 197–206). https://doi.org/10.1109/ICDM.2018.00035
20. Ko YJ, Maystre L, Grossglauser M (2016) Collaborative Recurrent Neural Networks for Dynamic Recommender Systems. In: Journal of Machine Learning Research (Vol. 63, pp 366–381).
21. Koren Y (2010). Collaborative Filtering with Temporal Dynamics. In: KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (pp 447–456). https://doi.org/10.1145/1721654.1721677
22. Lika B, Kolomvatsos K, Hadjiefthymiades S (2014) Facing the cold start problem in recommender systems. Exp Syst Appl 41:2065–2073. https://doi.org/10.1016/j.eswa.2013.09.005
23. Liu M, Pan W, Liu M, Chen Y, Peng X, Ming Z (2017) Mixed similarity learning for recommendation with implicit feedback. Knowl-Based Syst 119:178–185. https://doi.org/10.1016/j.knosys.2016.12.010
24. Mu D, Guo L, Cai X, Hao F (2017) Query-focused personalized citation recommendation with mutually reinforced ranking. IEEE Access 6:3107–3119. https://doi.org/10.1109/ACCESS.2017.2787179
25. Posts R (2020). Online Retailing : Britain , Europe , US. *The Property Chronicle*, 1–6. https://www.propertychronicle.com/online-retailing-britain-europe-us-canada-2017/TH
26. Quadrana M, Karatzoglou A, Hidasi B, Cremonesi P (2017) Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. In: RecSys 2017 - Proceedings of the 11th ACM Conference on Recommender Systems (pp 130–137). https://doi.org/10.1145/3109859.3109896
27. Gediminas Adomavicius and Alexander Tuzhilin (2011) Context-Aware Recommender Systems. In: Recommender Systems Handbook (pp 217–253). Springer US. https://doi.org/10.1007/978-0-387-85820-3
28. Reddy S, Mascia J (2006) Lifetrak: Music In Tune With Your Life. In: Proceedings of the ACM International Multimedia Conference and Exhibition (pp 25–34). https://doi.org/10.1145/1178745.1178754
29. Rendle S (2012) Factorization machines with libFM. ACM Trans Intell Syst Technol. https://doi.org/10.1145/2168752.2168771
30. Rendl e S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) BPR: Bayesian personalized ranking from implicit feedback. In: Information Retrieval (pp 452–459).
31. Sedhain S, Menony AK, Sannery S, Xie L (2015) AutoRec: Autoencoders Meet Collaborative Filtering. In: WWW 2015 Companion - Proceedings of the 24th International Conference on World Wide Web (pp 111–112). https://doi.org/10.1145/2740908.2742726
32. Shams B, Haratizadeh S (2017) Graph-based collaborative ranking. Exp Syst Appl 67:59–70. https://doi.org/10.1016/j.eswa.2016.09.013
33. Shen J, Karimzadehgan M, Bendersky M, Qin Z, Metzler D (2018) Multi-Task Learning for Email Search Ranking with Auxiliary Query Clustering. In: CIKM '18: Proceedings of the 27th ACM International Conference on Information and Knowledge-Management (pp 2127–2136). https://doi.org/10.1145/3269206.3272019
34. Shengyu L, Beizhan W, Hongji W, Q H (2018) A Hybrid Collaborative Filtering Algorithm Based On KNN and Gradient Boosting. In: The 13th International Conference on Computer Science & Education (ICCSE 2018*)* (pp 1–5). https://doi.org/10.1109/ICCSE.2018.8468751
35. Shi Y, Larson M, Hanjalic A (2014) collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges. ACM Comput Surv 47(1):1–45. https://doi.org/10.1145/2556270
36. Sreenivasa BR, Nirmala CR (2019) Hybrid location-centric e-commerce recommendation model using dynamic behavioural

traits of customer. Iran J Comput Sci 2(3):179–188. https://doi.org/10.1007/s42044-019-00040-3
37. Sun Y, Zhang Y (2018) Conversational Recommender System. In: 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018 (pp 235–244). https://doi.org/10.1145/3209978.3210002
38. Tan YK, Xu X, Liu Y (2016) Improved recurrent neural networks for session-based recommendations. In: ACM International Conference Proceeding Series (Vol. 15-September-2016, pp 17–22). Association for Computing Machinery. https://doi.org/10.1145/2988450.2988452
39. Wei J, He J, Chen K, Zhou Y, Tang Z (2017) Collaborative filtering and deep learning based recommendation system for cold start items. Exp Syst Appl 69:29–39. https://doi.org/10.1016/j.eswa.2016.09.040
40. Young J (2020) US ecommerce sales grow. Digital Commerce 360:1–9
41. Yu F, Liu Q, Wu S, Wang L, Tan T (2016) A Dynamic Recurrent Model for Next Basket Recommendation. In: SIGIR 2016 - 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
42. Zhu Y, Li H, Liao Y, Wang B, Guan Z, Liu H, Cai D (2017) What to Do Next: Modelling User Behaviours by Time-LSTM. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17) (pp 3602–3608).
43. Jing H, Smola AJ (2017) Neural survival recommender, WSDM 2017 - Proc. 10th ACM Int. Conf. Web Search Data Min., pp 515–524, doi: https://doi.org/10.1145/3018661.3018719.
44. Massimo Q, Paolo C, Dietmar J (2018) Sequence-Aware Recommender Systems. ACM Comput. Surv. 51, 4, Article 66 (September 2018), 36 pages. https://doi.org/10.1145/3190616.
45. Geoffray Bonnin and Dietmar Jannach. 2014. Automated Generation of Music Playlists: Survey and Experiments. ACM Comput. Surv. 47, 2, Article 26 (January 2015), 35 pages. https://doi.org/10.1145/2652481.
46. Mehdi HA, Negar H, Bamshad M, Robin B (2015) Adapting Recommendations to Contextual Changes Using Hierarchical Hidden Markov Models. In: Proceedings of the 9th ACM Conference on Recommender Systems. Association for Computing Machinery, New York, NY, USA, pp 241–244. https://doi.org/10.1145/2792838.2799684.
47. Hidasi B, Quadrana M, Karatzoglou A, Tikk D (2016) Parallel recurrent neural network architectures for feature-rich session-based recommendations, RecSys 2016 - Proc. 10th ACM Conf. Recomm. Syst., pp 241–248 https://doi.org/10.1145/2959100.2959167.