



A unified approach to designing sequence-based personalized food recommendation systems: tackling dynamic user behaviors

Jieyu Zhang^{1,2} · Zidong Wang² · Weibo Liu² · Xiaohui Liu² · Qiusheng Zheng¹

Received: 23 November 2022 / Accepted: 16 February 2023 / Published online: 25 March 2023
© The Author(s) 2023

Abstract

The recommender system (RS) is a well-known practical application of the state-of-the-art information filtering and machine learning technologies. Traditional recommendation approaches, including collaborative and content-based filtering techniques, have been widely employed to provide suggestions in RSs, where the user-item interaction matrix is the primary data source. In many application domains, interactions between users and items are more likely to be dynamic rather than static, and thus dynamic user behaviors should be taken into account when solving recommendation tasks in order to provide more accurate suggestions. In this work, we consider the sequentially ordered information from user-item interactions in the RSs where a sequence-based recommendation model is put forward with applications to the food recommendation scenario. Furthermore, the long short-term memory (LSTM) network is employed as the building block to establish such a recommendation model, and a collaborative filtering unit is adopted to make personalized food recommendation. The proposed LSTM-based RS is successfully applied to a real-world food recommendation data set. Experimental results demonstrate that the developed method outperforms some currently popular RSs in terms of precision, recall, mean average precision and mean reciprocal rank in food recommendation.

Keywords Food recommendation · Sequential prediction · Long-short term memory network · Collaborative filtering

This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) of the UK, the Royal Society of the UK, and the Alexander von Humboldt Foundation of Germany.

✉ Zidong Wang
Zidong.Wang@brunel.ac.uk

Jieyu Zhang
jieyu.zhang@brunel.ac.uk

Weibo Liu
Weibo.Liu2@brunel.ac.uk

Xiaohui Liu
Xiaohui.Liu@brunel.ac.uk

Qiusheng Zheng
zqszut@163.com

¹ Frontier Information Technology Research Institute, Zhongyuan University of Technology, Zhengzhou 450007, China

² Department of Computer Science, Brunel University London, Uxbridge, Middlesex UB8 3PH, UK

1 Introduction

Food has always been at the heart of human life. In the past, people had to identify and store food to survive, while in nowadays, people have more concerns about dietary needs including essential nutrition, health, taste, calories, and social occasions [8, 16]. Due to the growing information overload of various food-related content on multimedia, food recommender systems (RSs) are becoming increasingly attractive for people worldwide. Clearly, long-term unhealthy eating habits would be harmful to people's health with potential risks such as the development of undesired chronic diseases. Taking into consideration of the importance of healthy eating habits, the RS is now used as an efficient tool by people to make informed decisions on food selection according to their health conditions, thereby helping people develop healthy eating habits and reduce unaware health risks [32, 54–56].

Generally speaking, RSs have the advantage of saving time and money by using a series of algorithms to analyze users' food behaviors and ratings so as to recommend the most relevant and appealing foods to users [31]. Note that

there are still several challenges (e.g., diversity, adaptation and fluctuation) that hinder the further development and application of RSs. The diversity challenge lies in the fact that food RSs are required to be able to handle diverse types of food preferences of all individuals, e.g., different taste preferences, perceptual abilities, cognitive restrictions, cultural backgrounds, and even genetic influences [40, 41, 58]. The adaptation challenge means that, accounting for the fast change of food trends among people, food RSs are expected to constantly adapt to the latest food fashions in order to provide up-to-date food suggestions [9]. The fluctuation challenge implies that it is unreasonable to supply the users with a one-size-fits-all food recommendation taking into consideration of the dramatic fluctuation of users' food preferences [44]. Faced by the three challenges, there is a practical need to develop a novel recommendation technique to help people select food plans that are reasonable and personalized based on individually diverse, rapidly changing and dramatically fluctuated food preferences.

The collaborative filtering (CF) as well as the content-based filtering (CBF) are two widely used recommendation techniques for food preference learning. Typically, CF works by taking into account the food preferences of users with similar tastes, while CBF focuses on the attributes (e.g., ingredients, nutrition, and reviews) of the food itself [11]. Although CF and CBF achieve reasonable performance for preference learning, their adaptation to the change of user preferences or food contents is poor. To overcome this problem, it is crucial to consider the dynamic pattern of user-item interactions into the recommendation process so as to help better predict future preferences (of users) and optimize recommendation results accordingly. In addition, the consideration of such a dynamic pattern can also provide valuable insights into the interaction between users and recommendation applications/systems, and therefore help improve the user experience [36].

Recently, artificial neural networks have been successfully applied to RSs owing to their strong feature extraction abilities [7, 22, 35, 39, 42, 47, 49]. For example, a convolutional sequence embedding (Caser) RS has been proposed in [42] for product recommendation, where a convolutional neural network (CNN) is employed to capture the sequential features by analyzing the embedding matrix. It is worth mentioning that the embedding matrix can be treated as the "image" of the items in the latent space. Experimental results demonstrate the effectiveness of the proposed Caser RS in extracting sequential patterns by taking both sequential patterns and general preferences into account. In [49], a convolutional attention network has been put forward to explore the user behaviors by unifying a general RS and a sequential RS. A user-based recurrent neural network (RNN) has been developed in [7] for sequence prediction by integrating user information so as to provide personalized

recommendation. In [22], a multi-period product RS has been introduced for online food recommendation, where an RNN-based recommendation model is developed to provide product recommendation in multiple time periods.

Serving as a popular RNN, the long short-term memory (LSTM) network has been widely adopted in RSs with hope to comprehensively investigate the dynamic features through user-item interactions [2, 18, 22]. It should be noted that the LSTM network has shown competitive performance in capturing both the long-term and short-term patterns, which contributes to a comprehensive investigation of user behavior in RSs. Motivated by above discussions, it becomes a seemingly natural idea to employ LSTM networks to study the user-item interaction sequences in order to carry out personalized food recommendation. In this paper, a sequence-based recommendation approach is proposed to lay an effective and systematic basis for establishing food RSs. A traditional LSTM network is adopted to reflect users' food preferences and generate accurate recommendation suggestions. Furthermore, the proposed LSTM-based RS is tested on a public data set, and the experiments verify the promising performance of our approach for food recommendation.

The main contributions of this paper are outlined in three-fold as follows: (1) a unified framework is proposed for food recommendation that leverages feedbacks from sequences as well as historical interactions to model users' long- and short-term preferences; (2) a traditional LSTM network is employed to extract the user representation by considering the user habits at a certain time period; and (3) a series of numerical experiments are conducted on a real-world data set to validate the effectiveness of the developed RS. In summary, the established RS is capable of effectively modeling users' long- and short-term preferences and providing more accurate and diverse food recommendation in comparison with existing food recommendation techniques.

The remainder of this paper is organized as follows. In Sect. 2, the related work is presented on existing solutions for food recommendation. A sequence-based model is developed in Sect. 3 for food recommendation. The experimental results are discussed in Sect. 4 and appropriate evaluation metrics are carefully selected for evaluating the algorithm performance. Section 5 concludes the paper while pointing out the future directions.

2 Related work

In the food domain, RSs play an important role in promoting healthy eating behaviors by approaches such as suggesting healthier food substitutes to users. Food RSs can be divided into three types based on the information used for food recommendation [43]. The first type adopts users' food preference for recommendation, e.g., the search terms

or ingredient inputs of users have been utilized in [5] to conduct recipe recommendation. The second type leverages the healthy and nutritional needs of users for recommendation, e.g., a food plan has been generated in [33] based on healthy ingredients instead of harmful ones. The third type finds a trade-off between user preferences and nutritional needs, e.g., a healthy and nutritional meal plan has been made for the elderly in [38] by taking advantage of information from both user preferences and food nutrition.

In comparison with the general RSs, the food RSs have the following differences. The first difference is that the food RSs have to consider more factors when conducting recommendation, e.g., users' nutritional needs, weight goals, and health problems which are primary factors for food recommendation. The second difference is that different domain knowledge and food databases (e.g., nutritional, medical, and dietary information) are required by the RSs to supply users with healthier food suggestions. The last difference is that the unique characteristics (e.g., cooking methods, preparation time, and ingredient combination effects) of various food have to be concerned when making recommendation. In summary, more factors and information should be taken into good consideration by the RSs in order to provide users effective and healthy food plans [12].

For the purpose of improving the accuracy of food RSs, it is crucial to consider both users' dynamic preferences and historical neighbor feedbacks. In this paper, sequence-based recommender systems (SRSs) are introduced to capture dynamic preferences of users. Modeling the sequential pattern of users' behaviors allows the RSs to understand the evolution of user tastes over time, thereby providing better recommendation [51]. It is worth mentioning that the SRSs are different from traditional RSs in that SRSs account for the order of items via the perspective of users' historical behaviors, and thus both timing and frequency of interactions are taken into account for recommendation [10]. So far, the SRSs have been successfully applied in many applications such as e-commerce, music, and news recommendation [3, 14, 19–21, 34].

Incorporating sequence-based RSs into food recommendation has several advantages. First, food recommendation is often time-sensitive where suggestions are expected to be interactive. For example, if a customer has ordered a steak, it is reasonable for RSs to recommend a salad as a starter and a glass of red wine as an accompaniment. Second, SRSs make it possible to model the rather complicated couplings/interactions among different foods that are consumed together. For example, SRSs can capture the fact that consuming bread increases the likelihood of subsequently consuming milk. Third, SRSs are effective at handling implicit feedbacks that are more reliable than the explicit feedbacks (ratings) which are not always available [46].

Existing solutions for SRSs mainly fall into two categories which are the Markov chain models and deep learning techniques. The Markov chain model treats the users' behavior as a sequence of states and item recommendation is provided based on the state transition probabilities [1, 17, 37]. As for the deep learning techniques, typical examples are CNNs, RNNs, and graph neural networks, which have been widely used in a variety of sequence-based RSs [6, 7, 18, 42, 49]. Although the SRS has been implemented in a variety of domains, it has been rarely considered in the food domain due to the fact that food recommendation is a highly contextualized and personalized task which unavoidably leads to significant difficulties to the satisfactory design of SRSs. As such, we are motivated to investigate a specialized food recommendation approach that effectively integrates the SRSs with other types of RSs to provide a comprehensive and personalized solution for food recommendation.

3 Sequence-based deep learning model for food recommendation

In this section, the recommendation task is described with elaborated descriptions of the proposed model. We start by introducing some key concepts required for the model.

3.1 Problem formulation

Consider a set of m users $\mathcal{U} = \{u_1, u_2, \dots, u_m\}$ and a set of n items $\mathcal{I} = \{i_1, i_2, \dots, i_n\}$, where m and n are the sizes of the user set and item set, respectively. For user u_i , it has an ordered list of items \mathcal{S}^{u_i} according to the action sequences. For each user u_i , the prediction task can be written as:

$$\mathcal{S}_{t-L}^{u_i}, \dots, \mathcal{S}_{t-2}^{u_i}, \mathcal{S}_{t-1}^{u_i} \rightarrow \mathcal{S}_t^{u_i} \quad (1)$$

where t for \mathcal{S}^{u_i} denotes the temporal order in which actions happen. Given sequence $\mathcal{S}_{t-L}^{u_i}, \dots, \mathcal{S}_{t-2}^{u_i}, \mathcal{S}_{t-1}^{u_i}$, the model tries to predict $\mathcal{S}_t^{u_i}$ as the next item with which the user will interact. Two sequential patterns are considered in this paper, i.e., the point-level sequential pattern (PSP) and the union-level sequential pattern (USP).

3.1.1 Point-level sequential pattern

The point-level sequential pattern (PSP) is a type of sequence-to-point learning model, where predictions are made based on all the actions that have occurred up to a certain time point [57]. As shown in Fig. 1a, the output $\mathcal{S}_t^{u_i}$ is a predicted item for the next action. All of the previous points influence the target independently. For example, if we have a list of items ingested by a person over the course of a day, we may be interested in finding two possible patterns,

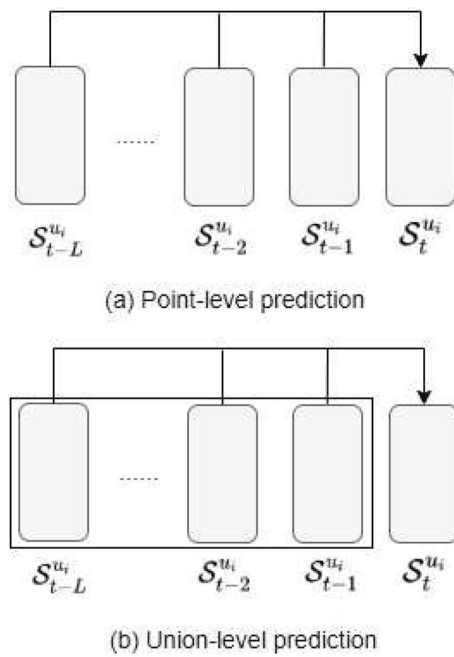


Fig. 1 Point and union-level sequential patterns

e.g., “coffee is usually consumed after dessert” and “chips are usually consumed after fish.”

3.1.2 Union-level sequential pattern

The USP tries to predict the behavior of users based on aggregating multiple interactions [23]. The pattern is based on the assumption that the union of all the event is a reasonable predictor for the next event in the sequence. Different from the PSP, the USP is able to identify items that are frequently co-consumed, such as the combination of breakfast and lunch. Figure 1b shows an illustration of the PSP, where several previous actions jointly influence the target action. The LSTM network is employed in our proposed approach to mine both PSP and USP that exist in users’ behaviors.

3.2 Modeling and learning

The proposed model mainly consists of two units, i.e., the LSTM unit and the CF unit, where the LSTM network attempts to discover long- and short-term preferences that exist in users’ interaction sequences and to determine the latent representation of the user in the embedding layer. The sequential learning process allows the model to learn behavior patterns of users’ preferences at both point- and union-level, which enables the model to make more accurate predictions about their future behaviors.

Specifically, the user-item interaction at each time step is transformed into one-hot encoded vector as the network

input. Then, these vectors are mapped to low-dimensional dense vectors through the embedding layer and passed to the LSTM network to capture the behavior pattern of each user. Afterward, the user embedding vector is calculated by averaging the trained embedding vectors. The results are fed as input vectors for the CF unit.

The CF unit makes recommendation by suggesting items (liked by other users with similar tastes) to target users. Cosine similarity is used to quantify the correlation between two users. After obtaining the similarity matrix, the most similar group of users (to the target users) is identified with the most liked item selected and forwarded to the target users as recommendation.

3.2.1 LSTM

Recurrent neural networks (RNNs) use data patterns to predict the probability of future events based on the sequential characteristics of the data [30]. Various ordinal or temporal problems can be solved using this method, such as language translation, natural language processing, speech recognition, and image captioning. In contrast to traditional deep neural networks, which assume that inputs and outputs are independent of each other, RNNs incorporate input and output information from previous inputs to influence the current input and output. The LSTM network, as a variation of the traditional RNN, is designed to better retain information over a long period of time.

In addition to learning the non-linear and non-stationary nature of sequential data, the LSTM network has the advantage of preserving information in memory for a long period of time, which is in line with the goal of capturing the union-level pattern. The LSTM network controls the flow of information using three gates: the forget gate, the input gate, and the output gate, where the forget gate determines which information requires attention and which may be ignored by using the update function given as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

where a sigmoid layer is applied on the input of the unit at time t and the last cell state, denoted by x_t and h_{t-1} , respectively. The next step is to determine what information should be stored in the current cell state. First, the input gate layer determines which values to update. Then, a tanh layer formulates a vector consisting of the values of new candidates, denoted as \tilde{C}_t , which can be added to the state. Such two layers combine to produce an update to the current state, which is defined as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

The new cell state C_t is decided by the old state $f_t * C_{t-1}$ and the new candidate value $i_t * \tilde{C}_t$. Finally, the output is generated from the current internal cell state C_t .

$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = o_t * \tanh(C_t) \tag{7}$$

where the values of the current state x_t and the previous hidden state h_{t-1} are passed into the sigmoid function to decide which parts of the cell state are to be updated. Then, the new cell state passes through the tanh function. Both of these outputs are multiplied point by point. The final hidden state h_t is used for prediction.

3.2.2 Customized LSTM network

The LSTM network is used in this paper to learn user representation. The input of the LSTM network is the item of the actual interaction, while the output is the predicted item which a user tends to interact with at next time step. The item is first converted to a one-hot encoding vector, where the length of the vector equals the number of items. Here, only the coordinate corresponding to the active item is one, and the rest coordinate are zeros. Then, the one-hot encoding is mapped to a learnable, low-dimensional vector through the embedding layer. After retrieving the pre-trained item embeddings, the user embedding can be calculated by averaging item embeddings. Note that the pre-trained process is independent for each user, and therefore the averaging embedding can be used as the reasonable representation for each user. Figure 2 depicts the structure of the LSTM unit. Additional embedding layers are added between the input and the LSTM layer, and the output is the predicted preference of the items.

3.2.3 CF unit

The CF unit starts with user embedding that represents the individual interest of each user. To find the user group most similar to the target user, the similarity between each pair of users is calculated using the cosine similarity measurement. The cosine similarity is defined as:

$$\text{sim}(x, y) = \frac{x \cdot y}{\|x\| \|y\|} \tag{8}$$

where $\| \cdot \|$ is the Euclidean norm of vector “.”. Conceptually, $\| \cdot \|$ is the length of the vector. The measure computes the cosine of the angle between vectors x and y . The greater

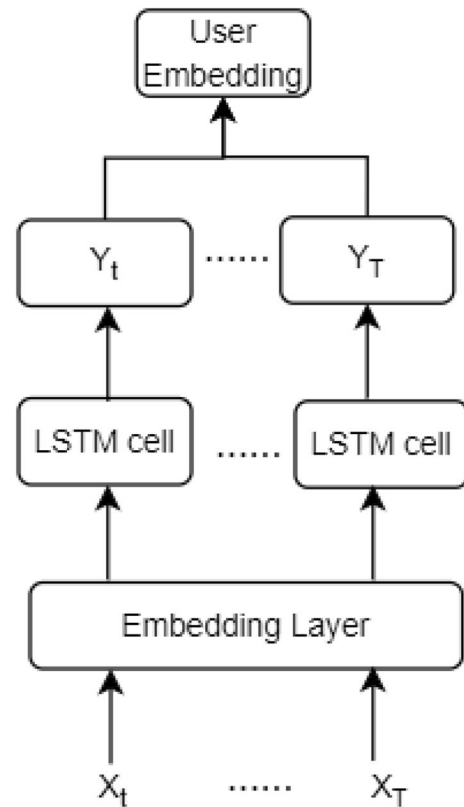


Fig. 2 General structure of the network

the cosine value is, the more similar the tastes of the two users are. The next step is to generate the recommendation. The top N most liked items have been retrieved from the target users’ neighborhood based on their popularity, and the recommendation lists are ranked according to their relevance and popularity. Table 1 provides the similarity matrix acquired from the CF unit.

4 Experiments and results

In this section, the proposed model is evaluated against popular baselines on one of the most popular food data sets, i.e., the Food.com data set which is previously the GeniusKitchen.com data set.

4.1 Data set

The website Food.com is arguably the largest food-oriented website that attracts 1.5 billion visits every year, and the adopted data set is comprised of 180K+ recipes as well as 700K+ reviews that cover user interactions for 18 years. Each interaction in the data set consists of a user identifier, a recipe identifier, and the corresponding rating and date. For

Table 1 Similarity matrix

user_id	1	2	3	4	5	6	7	8
1	1	-0.79	0.60	-0.76	0.84	-0.78	0.79	0.57
2	-0.79	1	-0.57	0.88	-0.93	0.93	-0.91	-0.60
3	0.60	-0.57	1	-0.54	0.63	-0.56	0.64	0.68
4	-0.76	0.88	-0.54	1	-0.90	0.93	-0.92	-0.54
5	0.84	-0.93	0.63	-0.90	1	-0.96	0.95	0.66
6	-0.78	0.93	-0.56	0.93	-0.96	1	-0.95	-0.61

Table 2 Food.com data set

user_id	recipe_id	Date	Interaction
1	122140	2011.01.04	1
1	77036	2011.01.05	1
1	156817	2011.01.06	1
1	76957	2011.01.07	1
1	68818	2011.01.08	1

better model performance, the explicit feedbacks have been converted to the implicit feedback.

Table 2 shows some examples of the Food.com data set. The user_id and recipe_id represent the user identifier and the recipe identifier, respectively. The date indicates the record time of this entry, and the interactions identify that the user has consumed the item.

4.2 Evaluation metrics

The goal of the experiments is to evaluate the quality and performance of the proposed approach against various baselines. For each user, the last 20% interactions are held as the test set and the remaining data are utilized for training. The performance of the utilized RSs is measured by precision@N, recall@N, mean average precision (MAP), as well as mean reciprocal rank (MRR). Precision refers to the number of retrieved items that are relevant, while recall indicates the number of relevant items that are retrieved. Precision@N and Recall@N are defined as:

$$\text{Prec@N} = \frac{|R \cap \hat{R}_{1:N}|}{N} \quad (9)$$

$$\text{Recall@N} = \frac{|R \cap \hat{R}_{1:N}|}{|R|} \quad (10)$$

where $\hat{R}_{1:N}$ denotes a list of top-N predicted items for a user and R denotes the last 20% of actions in the test set. To evaluate the overall performance of the approach, the MAP and MRR are used. The MAP is widely used in the RS for its ability to provide general estimation of model

performance. The MAP is the average of the average precision (AP) defined by:

$$\text{AP} = \frac{\sum_{N=1}^{|\hat{R}|} \text{Prec@N} \times \text{rel}(N)}{|\hat{R}|} \quad (11)$$

where $\text{rel}(N) = 1$ if the N th items are in the same ranking order in both prediction and test sets. The MRR is used to assess the performance of a CF unit and calculated as the mean of the reciprocal ranks of the items retrieved by the approach. The MRR is defined as:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \quad (12)$$

4.3 Experiment setting

In this paper, three widely used baselines (including the Item k nearest neighbor (Item-KNN) algorithm [25], the Meta-Prod2vec collaborative filtering (MPCF) algorithm [45], and the convolutional sequence embedding recommendation (Caser) algorithm [42]) are selected as the benchmark.

- Item-KNN: Item-KNN recommends items similar to the target item, and similarity is defined as the cosine similarity between the vectors of the user interaction history.
- MPCF: The Meta-Prod2vec method computes low-dimensional embeddings of items based on previous interactions with the items. The representation of a user is calculated as the mean of the products consumed by the user.
- Caser: A personalized top-N sequential recommendation framework which uses CNNs for sequence modelling.

The recommended item numbers are set to be 1, 5, and 10 in the experiment to evaluate the performance of the utilized RSs. The learning rate, the minimum epoch and the mini-batch size of the MPCF algorithm, the Caser algorithm and the proposed approach are set to be 0.001, 50 and 128, respectively. The numbers of horizontal filters and the vertical filters of the Caser algorithm are set to be 16 and 4, respectively. In the proposed approach, the number of

layer and the size of hidden neurons are set to be 1 and 30, respectively.

4.4 Performance comparison

The evaluation results of the three baselines and the proposed approach are presented in Table 3, where the best performer in each row is highlighted in bold, and the last column also included the improvement of the proposed approach over the best baseline in percentages. As shown in Table 3, the proposed method outperforms the Item-KNN method, the MPCF method, and the Caser method in terms of Prec@5, Prec@10, Recall@5, MAP and MRR. In addition, the proposed method obtains the second-best results in terms of Prec@1, Recall@1 and Recall@10 comparing to the other three baseline methods. In general, we can draw the conclusion that the proposed method outperforms the baseline methods with respect to the four chosen evaluation

metrics. It should also be noted that sequential RSs (e.g., MPCF and Caser) outperform the Item-KNN method (which is the traditional RS), suggesting that the considered sequential patterns in user behaviors lead to higher accuracy.

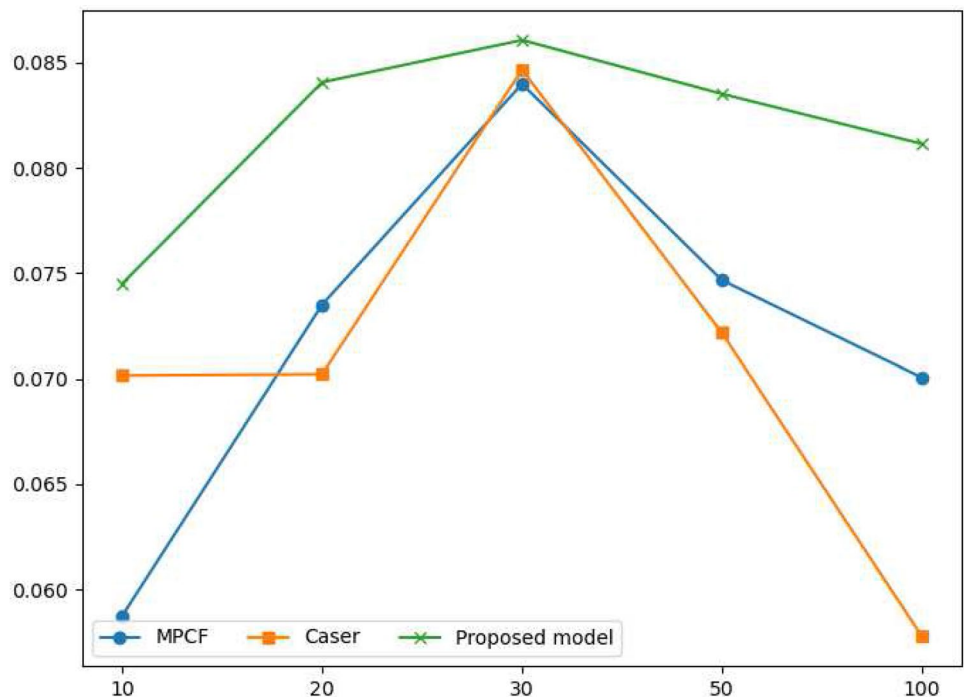
In our experiment, the embedding dimension is a key hyper-parameter which is optimized through the model selection process. To obtain an optimal solution of the embedding dimension, we adopt the embedding dimension from 10 to 100, and compare the MAP of two baselines with that of proposed model on different embedding dimensions, as shown in Fig. 3.

Figure 3 shows the MAP of two baseline plus the proposed model based on different embedding dimensions. Among these baselines, the MPCF, Caser, and the proposed model achieve their best performance with the embedding dimension of 30. It should be noted that performance does not improve with the increase of the dimension. Overall, the proposed model beats the strongest baseline based on the

Table 3 Performance comparison

Metric	Item-KNN	MPCF	Caser	Proposed model	Improvement (%)
Prec@1	0.0214	0.0257	0.0284	0.0281	- 1.05
Prec@5	0.0188	0.023	0.0244	0.0249	+2.04
Prec@10	0.0199	0.0203	0.0209	0.0226	+8.13
Recall@1	0.0428	0.0514	0.0569	0.0562	- 1.23
Recall@5	0.0564	0.0692	0.0732	0.0748	+2.18
Recall@10	0.0796	0.0814	0.0838	0.0815	- 2.74
MAP	0.06385	0.08398	0.08465	0.08606	+1.66
MRR	0.06102	0.06962	0.07403	0.08003	+7.49

Fig. 3 MAP (y-axis) vs. the number of the latent dimension d (x-axis)



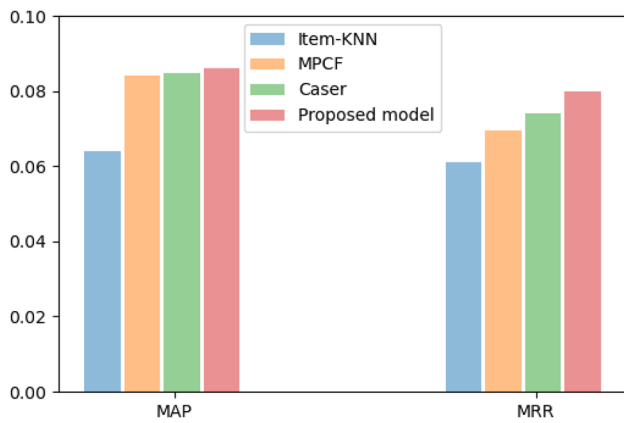


Fig. 4 Comparing Prec@10 and Recall@10 of the proposed solution against three baselines

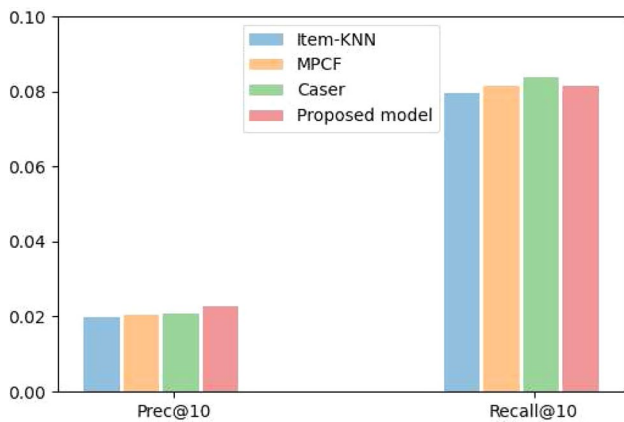


Fig. 5 Comparing MAP and MRR of the proposed solution against three baselines

selected range and shows a rather steady trend compared to other baselines, which verifies the stability of the approach. Figures 4 and 5 compare the proposed solution against Item-KNN, MPCF and Caser on four metrics in the form of bar charts.

5 Conclusion

In this paper, a novel sequence-based recommendation approach has been developed to solve food recommendation tasks. Specifically, LSTM networks are used to approximate user-item interactions where CF techniques are adopted to make recommendation. Experimental results show reasonable performance gains over the popular baseline of the sequence-based RSs. Some future research directions include (1) the adoption of additional information (e.g., images, reviews and browsing history); (2) the proposal of explainable and personalized food recommendation; and (3)

the introduction of more advanced machine learning techniques for cross-domain recommendations, see e.g. [4, 13, 15, 24, 26–29, 48, 50, 52, 53, 59].

Data availability statement The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Cai C, He R, McAuley J (2017) SPMC: Socially-aware personalized Markov chains for sparse sequential recommendation. ArXiv preprint, [arXiv:1708.04497](https://arxiv.org/abs/1708.04497)
2. Cao J, Wang Y, Tao H, Guo X (2022) Sensor-based human activity recognition using graph LSTM and multi-task classification model. *ACM Trans Multimed Comput Commun Appl* 18(139):1–19
3. Chen S, Moore JL, Turnbull D, Joachims T (2012) Playlist prediction via metric embedding. In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. Beijing, China, pp 714–722
4. Chen Y, Song Q, Zhao Z, Liu Y, Alsaadi FE (2022) Global Mittag-Leffler stability for fractional-order quaternion-valued neural networks with piecewise constant arguments and impulses. *Int J Syst Sci* 53(8):1756–1768
5. Cruz ZMC, Alpay JJR, Depeno JDD, Altabirano MJC, Bringula R (2017) Usability of “Fatchum”: a mobile application recipe recommender system. In: *Proceedings of the 6th annual conference on research in information technology*, Rochester, New York, USA, Oct, pp 11–16
6. Dong X, Jin B, Zhuo W, Li B, Xue T (2021) Improving sequential recommendation with attribute-augmented graph neural networks. ArXiv preprint, [arXiv:2103.05923](https://arxiv.org/abs/2103.05923)
7. Donkers T, Loepp B, Ziegler J (2017) Sequential user-based recurrent neural network recommendations. In: *Proceedings of the 11th ACM conference on recommender systems*, Como, Italy, pp 152–160
8. Elswailer D, Harvey M (2015) Towards automatic meal plan recommendations for balanced nutrition. In: *Proceedings of the 9th ACM conference on recommender systems*, Vienna, Austria, pp 313–316
9. Elswailer D, Trattner C, Harvey M (2017) Exploiting food choice biases for healthier recipe recommendation. In: *Proceedings of the 40th international ACM sigir conference on research and development in information retrieval*, Shinjuku, Japan, pp 575–584
10. Fang H, Guo G, Zhang D, Shu Y (2019) Deep learning-based sequential recommender systems: Concepts, algorithms, and

- evaluations. In: International conference on web engineering, Daejeon, South Korea, pp 574–577
11. Forbes P, Zhu M (2011) Content-boosted matrix factorization for recommender systems: experiments with recipe recommendation. In: Proceedings of the 5th ACM conference on recommender systems, Chicago, Illinois, USA, pp 261–264
 12. Gaikwad DS, Deshpande AV, Nalwar NN, Salave AV, Katkar MV (2017) Food recommendation system. *Int Res J Eng Technol* 4(1):535–536
 13. Gao C, He X, Dong H, Liu H, Lyu G (2022) A survey on fault-tolerant consensus control of multi-agent systems: trends, methodologies and prospects. *Int J Syst Sci* 53(13):2800–2813
 14. Garcin F, Dimitrakakis C, Faltings B (2013) Personalized news recommendation with context trees. In: Proceedings of the 7th ACM conference on recommender systems, Hong Kong, China, pp 105–112
 15. Han X, Liu YH, Zhang X, Yang H (2022) Study on egg sorting model based on visible-near infrared spectroscopy. *Syst Sci Control Eng* 10(1):733–741
 16. Harvey M, Ludwig B, Elswailer D (2013) You are what you eat: learning user tastes for rating prediction. International symposium on string processing and information retrieval, Jerusalem, Israel, pp 153–164
 17. He R, McAuley J (2016) Fusing similarity models with Markov chains for sparse sequential recommendation. In: Proceedings of the 16th international conference on data mining (ICDM), Barcelona, Brazil, pp 191–200
 18. Hidasi B, Karatzoglou A, Baltrunas L, Tikk D (2015) Session-based recommendations with recurrent neural networks. *ArXiv preprint, arXiv:1511.06939*
 19. Hidasi B, Quadrana M, Karatzoglou A, Tikk D (2016) Parallel recurrent neural network architectures for feature-rich session-based recommendations. In: Proceedings of the 10th ACM conference on recommender systems, Boston, Massachusetts, USA, pp 241–248
 20. Jannach D, Lerche L, Jugovac M (2015) Adaptation and evaluation of recommendations for short-term shopping goals. In: Proceedings of the 9th ACM conference on recommender systems, Vienna, Austria, pp 211–218
 21. Jannach D, Lerche L, Kamehkhosh I (2015) Beyond “hitting the hits”: generating coherent music playlist continuations with the right tracks. In: Proceedings of the 9th ACM conference on recommender systems, New York, USA, pp 187–194
 22. Lee HI, Choi IY, Moon HS, Kim JK (2020) A multi-period product recommender system in online food market based on recurrent neural networks. *Sustainability* 12(3):969
 23. Li C, Niu X, Luo X, Chen Z, Quan C (2019) A review-driven neural model for sequential recommendation. *ArXiv preprint, arXiv:1907.00590*
 24. Li H, Wu P, Zeng N, Liu Y, Alsaadi FE (2022) A survey on parameter identification, state estimation and data analytics for lateral flow immunoassay: from systems science perspective. *Int J Syst Sci* 53(16):3556–3576
 25. Linden G, Smith B, York J (2003) Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Comput* 7(1):76–80
 26. Luo X, Wu H, Wang Z, Wang J, Meng D (2022) A novel approach to large-scale dynamically weighted directed network representation. *IEEE Trans Pattern Anal Mach Intell* 44(12):9756–9773
 27. Luo X, Wu H, Li Z (2022) NeuLFT: a novel approach to nonlinear canonical polyadic decomposition on high-dimensional incomplete tensors. *IEEE Trans Knowl Data Eng. <https://doi.org/10.1109/TKDE.2022.3176466>*
 28. Luo X, Zhou Y, Liu Z, Zhou MC (2021) Fast and accurate non-negative latent factor analysis on high-dimensional and sparse matrices in recommender systems. *IEEE Trans Knowl Data Eng. <https://doi.org/10.1109/TKDE.2021.3125252>*
 29. Luo X, Zhong Y, Wang Z, Li M (2021) An alternating-direction-method of multipliers-incorporated approach to symmetric non-negative latent factor analysis. *IEEE Trans Neural Netw Learn Syst. <https://doi.org/10.1109/TNNLS.2021.3125774>*
 30. Medsker LR, Jain LC (2001) Recurrent neural networks. *Des Appl* 5:64–67
 31. Min W, Jiang S, Jain R (2020) Food recommendation: framework, existing solutions, and challenges. *IEEE Trans Multimed* 22(10):2659–2671
 32. Nag N, Pandey V, Jain R (2017) Health multimedia: Lifestyle recommendations based on diverse observations. In: Proceedings of the 2017 ACM on international conference on multimedia retrieval, Bucharest, Romania, pp 99–106
 33. Nag N, Pandey V, Jain R (2017) Live personalized nutrition recommendation engine. In: Proceedings of the 2nd international workshop on multimedia for personal health and health care, New York, USA, pp 61–68
 34. Pauws S, Verhaegh W, Vossen M (2006) Fast generation of optimal music playlists using local search. *ISMIR*, pp 138–143
 35. Pinto ER, Nepomuceno EG, Campanharo ASLO (2022) Individual-based modelling of animal brucellosis spread with the use of complex networks. *Int J Netw Dyn Intell* 1(1):120–129
 36. Quadrana M, Cremonesi P, Jannach D (2018) Sequence-aware recommender systems. *ACM computing surveys (CSUR)*, pp 1–36
 37. Rendle S, Freudenthaler C, Schmidt-Thieme L (2010) Factorizing personalized Markov chains for next-basket recommendation. In: Proceedings of the 19th international conference on world wide web-WWW’10, Raleigh, North Carolina, USA, pp 811–820
 38. Ribeiro D, Ribeiro J, Vasconcelos MJM, Vieira EF, Barros ACD (2017) Souschef: improved meal recommender system for portuguese older adults. In: Proceedings of the international conference on information and communication technologies for ageing well and e-Health, Porto, Portugal, pp 107–126
 39. Shakiba FM, Shojaei M, Azizi SM, Zhou M (2022) Real-time sensing and fault diagnosis for transmission lines. *Int J Netw Dyn Intell* 1(1):36–47
 40. Schfer H, Elahi M, Elswailer D, Groh G, Harvey M, Ludwig B (2017) User nutrition modelling and recommendation: balancing simplicity and complexity. In: Adjunct publication of the 25th conference on user modeling, adaptation and personalization, New York, USA, pp 93–96
 41. Szankin M, Kwasniewska A (2022) Can AI see bias in X-ray images? *Int J Netw Dyn Intell* 1(1):48–64
 42. Tang J, Wang K (2018) Personalized top-n sequential recommendation via convolutional sequence embedding. In: Proceedings of the 11th ACM international conference on web search and data mining, New York, USA, pp 565–573
 43. Trang TTN, Atas M, Felfernig A, Stettinger M (2018) An overview of recommender systems in the healthy food domain. *J Intell Inf Syst* 50(3):501–526
 44. Trattner C, Elswailer D (2017) Food recommender systems: important contributions, challenges and future research directions. *ArXiv preprint, arXiv:1711.02760*
 45. Vasile F, Smirnova E, Conneau A (2016) Meta-Prod2Vec-product embeddings using side-information for recommendation. In: Proceedings of the 10th ACM conference on recommender systems, New York, USA, pp 225–232
 46. Wang M, Gong M, Zheng X, Zhang K (2018) Modeling dynamic missingness of implicit feedback for recommendation. *Adv Neural Inf Process Syst* 31:6669–6678
 47. Wang X, Sun Y, Ding D (2022) Adaptive dynamic programming for networked control systems under communication constraints: a survey of trends and techniques. *Int J Netw Dyn Intell* 1(1):85–98

48. Wen P, Li X, Hou N, Mu S (2022) Distributed recursive fault estimation with binary encoding schemes over sensor networks. *Syst Sci Control Eng* 10(1):417–427
49. Yakhchi S, Behehti A, Ghafari S, Razzak I (2022) A convolutional attention network for unifying general and sequential recommenders. *Inf Process Manag* 59(1):102755
50. Yang F, Li J, Dong H, Shen Y (2022) Proportional-integral-type estimator design for delayed recurrent neural networks under encoding-decoding mechanism. *Int J Syst Sci* 53(13):2729–2741
51. Yap G-E, Li X-L, Yu PS (2012) Effective next-items recommendation via personalized sequential pattern mining. In: *Proceedings of the international conference on database systems for advanced applications*, Busan, South Korea, pp 48–64
52. Yuan Y, Ma G, Cheng C, Zhou B, Zhao H, Zhang H-T, Ding H (2020) A general end-to-end diagnosis framework for manufacturing systems. *Natl Sci Rev* 7(2):418–429
53. Yuan Y, Tang X, Zhou W, Pan W, Li X, Zhang H-T, Ding H, Goncalves J (2019) Data driven discovery of cyber physical systems. *Nat Commun* 10(1):1–9
54. Yue W, Wang Z, Tian B, Pook M, Liu X (2021) A hybrid model- and memory-based collaborative filtering algorithm for baseline data prediction of Friedreich's Ataxia patients. *IEEE Trans Ind Inform* 17(2):1428–1437
55. Yue W, Wang Z, Liu W, Tian B, Lauria S, Liu X (2021) An optimally weighted user- and item-based collaborative filtering approach to predicting baseline data for Friedreich's Ataxia patients. *Neurocomputing* 419:287–294
56. Yue W, Wang Z, Zhang J, Liu X (2021) An overview of recommendation techniques and their applications in healthcare. *IEEE/CAA J Autom Sin* 8(4):701–717
57. Zhang C, Zhong M, Wang Z, Goddard N, Sutton C (2018) Sequence-to-point learning with neural networks for non-intrusive load monitoring. In: *Proceedings of the AAAI conference on artificial intelligence*, New Orleans, Louisiana, USA, pp 2604–2611
58. Zhao G, Li Y, Xu Q (2022) From emotion AI to cognitive AI. *Int J Netw Dyn Intell* 1(1):65–72
59. Zhao Y, He X, Ma L, Liu H (2022) Unbiasedness-constrained least squares state estimation for time-varying systems with missing measurements under round-robin protocol. *Int J Syst Sci* 53(9):1925–1941

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.