A Multi-Period Product Recommender System in Online Food Market based on Recurrent Neural Networks

Hea In Lee [1], Il Young Choi [2], Hyun Sil Moon [2] and Jae Kyeong Kim [3],*

[1]   Department of Social Network Science, Kyunghee University, Seoul 02447, Korea; hil106@khu.ac.kr
[2]   Graduate School of Business Administration & AI Research Management Center, Kyunghee University,
     Seoul 02447, Korea; choice102@khu.ac.kr (I.Y.C.); pahunter@khu.ac.kr (H.S.M.)
[3]   School of Management, Kyunghee University, Seoul 02447, Korea
*   Correspondence: jaek@khu.ac.kr; Tel.: +82–2–961–9355

Abstract: A recommender system supports customers to find information, products, or services (such as music, books, movies, web sites, and digital contents), so it could help customers to make rapid routine decisions and save their time and money. However, most existing recommender systems do not recommend items that are already purchased by the target customer, so are not suitable for considering customers' repetitive purchase behavior or purchasing order. In this research, we suggest a multi-period product recommender system, which can learn customers' purchasing order and customers' repetitive purchase pattern. For such a purpose we applied the Recurrent Neural Network (RNN), which is one of the artificial neural network structures specialized in time series data analysis, instead of collaborative filtering techniques. Recommendation periods are segmented as various time-steps, and the proposed RNN-based recommender system can recommend items by multiple periods in a time sequence. Several experiments with real online food market data show that the proposed system shows higher performance in accuracy and diversity in a multi-period perspective than the collaborative filtering-based system. From the experimental results, we conclude that the proposed system is suitable for multi-period product recommendation, which results in robust performance considering well customers' purchasing orders and customers' repetitive purchase patterns. Moreover, in terms of sustainability, we expect that our study contributes to the reduction of food wastes by inducing planned consumption, and the reduction of shopping time and effort.

Keywords: recommender systems; recurrent neural networks; multi-period prediction; sequential data analysis

---

1. Introduction

As the interest in personalization services increases in various fields, recommender systems applying various knowledge discovery techniques are being studied commercially and academically [1]. Especially, product recommendation systems, mostly developed based on online commerce, have been gradually becoming important in terms of sales and customer relationship as well as helping consumers to choose [2]. Collaborative filtering (CF) is a technique known as showing the best performance in product recommender systems [3,4]. The underlying assumption of collaborative filtering is that 'customers with similar preferences for particular items will show similar preferences for other items'. CF-based recommendation models predict preference based on the similarity between users or items, but scalability and sparsity problems may have occurred due to data increases as e-commerce grows [4]. However, as online business and technology advances, customers transaction data not only increases [5] but also consumers' consumption patterns are changing [6]. Consumers not

only use online purchases to buy cellular phone accessories and clothes, but they also buy almost all products they need for everyday life. The fresh food market is a typical example where preference changes over time and recurring purchases frequently occur. Its products are not expensive enough for frequent purchases so customers suffer repetitive decision-making problems. Therefore, there is a need for recommender systems that can process and analyze a huge amount of data and recommend the multi-period products needed in everyday life. Multi-period product recommender systems reflect purchasing patterns of customers, but traditional CF-based recommender systems do not consider well customers' purchasing orders [7].

In recent years, interest in artificial neural networks has been increasing, and it has been used in various fields such as pattern recognition, natural language processing, and image recognition [8]. Among various structures of artificial neural networks, Recurrent Neural Networks (RNN) show outstanding performance on sequential data, because RNN has a structure that is able to selectively pass information across sequence steps while processing sequential data one element at a time [9,10].

In this study, we propose a multi-period product recommender system using RNN that considers customers' purchase order. It can reflect the customers' preference changes and is expected to improve the recommendation quality compared with comparative existing recommender systems. For the evaluation of the multi-period product recommender systems, we segment recommendation period as various time-steps and the proposed system and other benchmark systems are comparatively evaluated by multiple periods in a time sequence. Also, the previously purchased products are also included from the recommendation list in this paper. As in the case of frequent repetitive purchasing products such as fresh food, recommender systems should consider repurchase behavior. It is important to consider items to be repurchased in advance because this can provide a positive effect to both customers and sellers, such as giving appropriate promotions, customer churn prevention, and increasing sales through them.

The real online transaction data is used for experiments and we compare the recommendation performance between the proposed RNN-based recommender system and the benchmark recommender system, item-based CF [11]. Experiment results indicate that the proposed system outperforms higher recommendation accuracy and diversity in a multi-period perspective. Based on the experiments, we claim that it is beneficial to consider purchase order information to recommend repurchasing items in multi-period recommender systems.

2. Literature Review

2.1. Recommender Systems

Recommender systems, also called recommendation systems, are kind of information filtering systems that analyzes user's past behavior data and seek to predict the user's preference to items [12]. They are mainly focused on individuals who lack the personal experience, not a segmentation group of customers. Since the 1990s, various recommender systems have been studied in various domains such as movies, music, books, articles, social media and products in general [4,13–15]. Burke [16] distinguishes recommendation approaches to five different classes; content-based, collaborative filtering (CF), demographic, knowledge-based, and hybrid recommender systems. Among them, CF is considered as a representative model of a recommender system, which was first introduced by Goldberg et al. [3]. Collaborative filtering builds a model to predict the preference based on the similarity between users or items, and the underlying assumption is that users with similar preferences for a particular item will have similar preferences for other items. Some studies have pointed that the main advantages of these neighborhood-based methods compared with other approaches are simplicity, justifiability, efficiency, and so on. Therefore, many studies have been conducted to improve their performance using some supplementary tools based on machine learning techniques such as clustering. However, collaborative filtering has often suffered from a problem of sparsity caused by a lack of data to predict users' preferences for items [17,18]. The number of items sold on an e-commerce

site is very large, but even the most active users would have rated only a small part of the entire items. Therefore, even the most popular items have no rating data. In addition, as online purchases have become popular recently, available item-user data sets have increased. As a result, the computational complexity for recommendations has increased, resulting in scalability problems [4]. As well as sparsity and scalability problems due to the increase of usable data in recommender systems, most of the CF approaches have the problem that almost same products are recommended because it does not reflect the change of customers' preference over time. Most of the CF approaches only use information about user's purchases over a specific period and do not utilize information about the order of purchase items. Song et al. [19] attempted to find changes in customers' behavior by creating association rules from two different points of view datasets and compared the association rules at two points to find out changes in customers' purchase behavior over time. However, they analyzed only a comparison between two points of purchasing using the association rules and did not find any long-term pattern changes.

On the other hand, research has been proposed to reflect temporal dynamics in order to take into account the changing preferences of customers [20–22]. The method that reflects the temporal dynamics includes a way of reducing weight by time flow based on the time when the item is purchased or using a moving window method which only a certain period of data is used when recommending. As seen in previous studies, the forgetting method was used to capture the preferences of customers. This forgetting method is done either using a fixed-size moving window over data and repeatedly training the model with data in the window size using a decay function in the similarity calculation making older items less relevant [22,23]. In this study, we use RNN, an artificial neural network model suitable for sequential data, in which input and output are both sequences, to capture preference changes in individual user-level. We propose an improved recommendation model that learns how previous purchase history affects the next purchase history.

2.2. RNN and Recommendation

RNN (Recurrent Neural Networks) is a structure appropriate for analyzing time series data since past information is stored in the hidden node and transferred to the next steps, which mean that previous input data can affect predictions of the current output [8]. The previous state is stored or memorized in the current state, that is, the data at time t can be considered when predicting the next state of time t+1. The hidden unit of the RNN serves as a memory block, and each memory block receives data at a specific time t. The memory block receiving the input at time t transfers the information to the connected memory block. However, the input sequence gets longer, errors may not be conveyed forward, which may lead to a problem that is called long-term dependency. In 1997, Hochreiter and Schmidhuber [24] first introduced the vanishing gradient problem and presented LSTM (Long Short-Term Memory) to address this problem. In 2014, Cho et al. [25]. presented GRU (Gated Recurrent Unit), a simple variant of LSTM. Many variants of RNN have appeared, but LSTM and GRU are the most commonly used RNN structures.

Recently, studies using RNN have been actively carried out in research on recommender systems. Zhang et al. [26] used RNN to predict online advertisement clicks and attempted to predict the user's next clicks. Hidasi et al. [27] viewed the recommendation problem as a session-based recommendation and then the use of Gated Recurrent Units (GRUs), a variant of RNN, for modeling user behavior in a session-based scenario in internet sites. Based on these studies, the session-based recommendation, various follow-up studies have been conducted. Tan et al. [28] presented various training methods to improve the performance of the model presented in Hidasi et al. [27]. I.e., data augmentation and a method to account for shifts in input data distribution. However, session-based recommendation is only justified by the fact that most e-commerce sites do not track user behavior beyond the session level. On the other hand, Song et al. [29] modeled both static and temporal user characteristics, assuming that user interest changes over time. The authors saw static features as the entire dataset and temporal features as the most recent dataset. Yu et al. [30] viewed the problem as the next-basket prediction in e-commerce. They proposed a RNN model with real-valued representations of the baskets as an

input and trained to rank the items in the next basket. In those scenarios, items for a shopping cart are suggested based on a user's history of past shopping carts.

As previous studies have shown, attempts have been made to improve the accuracy of recommender systems using a RNN model for learning time-related or sequential information from continuous data. In order to capture customers' preference, we first divide the recommendation point into several sections to see whether customers' preferences change or not. In this study, the RNN and its variants will be reviewed and a RNN-based recommendation model considering purchase order is presented to compare the results from traditional recommendation models.

3. Methodology

3.1. Overview

The purpose of this study is to introduce a recommendation model considering purchase order to capture customers' changing preferences and to examine the recommendation results of the proposed model from a multi-period perspective. Generally, customers received recommendation items at time T based on the purchase history up to the time point T-1. Likewise, in this study, it is assumed that the transaction information before the specific time point T-1 is analyzed and the target customer is recommended to purchase items at the next time point T. Therefore, the time point T does not mean a specific timespan, but basket sequences. For example, a customer purchases 10 times through the online food market, total T equals 10. The purchase at T-1 means the previous purchase(basket), i.e., 9th purchase(basket). More specifically, a set of all customers is represented as $U = \{u_1, \ldots, u_p\}$, and a set of all items is represented as $I = \{i_1, \ldots, i_q\}$. Let B_T be the set of all transaction data by all customers at time T, and represented as $B_T = \{B_T^{u_1}, B_T^{u_2}, \ldots, B_T^{u_p}\}$. To be precise, T stands for the timestamp of a purchase, not the actual time. $B_T^{u_p}$ is a subset of I and consisted of purchased items grouped into a basket according to the same purchase order at time T. Then, the total purchasing history of a specific customer u_p at time T is sorted in the order of purchase time, as $B_1^{u_p}, B_2^{u_p}, \ldots, B_{T-1}^{u_p}, B_T^{u_p}$. In this study, given the previous purchasing history, a problem of sequential prediction is set for each user u_p to recommend a set of items to be purchased at next time $B_{T+1}^{u_p}$. To do this, an RNN-based recommendation model is proposed to consider purchasing order. We divide the recommendation period into various time points, and experiment using real transaction data to see how well the proposed model reflects the changing customer's preference. The schematic framework of the proposed approach is described in Figure 1.
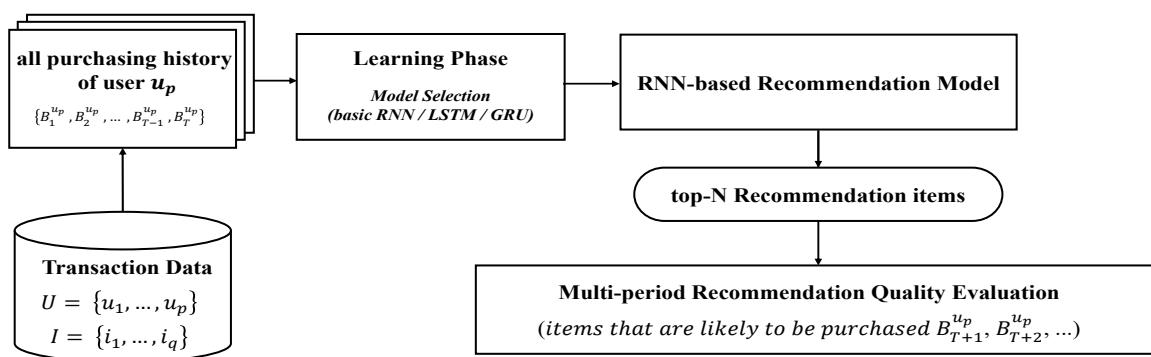
Figure 1. Schematic framework of the proposed recommendation system.

3.2. RNN-based Recommendation Model

The recommendation model presents items likely to be purchased by customers at next time T+1 based on the previous purchasing history. In this study, we propose a recommendation model considering both purchase information and purchasing order using RNN which could take sequential pattern into account. The proposed RNN-based recommendation model is shown in Figure 2. In the

case of repetitive purchasing pattern over time, the RNN-based recommendation model could learn the temporal changes according to the purchasing order because RNN has a mechanism for storing the previous information in hidden units. To design a RNN model for learning purchase information and purchasing order, data must be input to learn time-dependent features. Since the input of the neural networks should be converted into a vector, each item i_q is encoded as a one-hot encoding and the purchase information $B_T^{u_p}$ of the customer u_p is converted into multi-hot encoding by adding one-hot encoding of items at the same purchasing order at time T. x_T is represented as a vector converted into a multi-hot encoding of all items in $B_T^{u_p}$ and the total length of the vector is same as the number of all items. i_q is 1 if a customer purchased the item, otherwise it is 0. The output o_T is passed through the softmax function and represented as the probability of purchase. The model can be seen as learning the previous purchase information and representing the purchase pattern that will appear at the next time in probability. In the learning process, the difference between the predicted output (o_T) and the actual target (y_T) is calculated by the loss function, here by category cross-entropy, and the weights are updated through back-propagation of the error. Finally, the top-N items with the greatest probability are recommended.

x_T : vector of all items in basket $B_T^{u_p}$ ($i_p = 1\ if\ purchased,\ if\ not\ 0$), o_T :a real value for predicted items
|T| = all previous purchase history
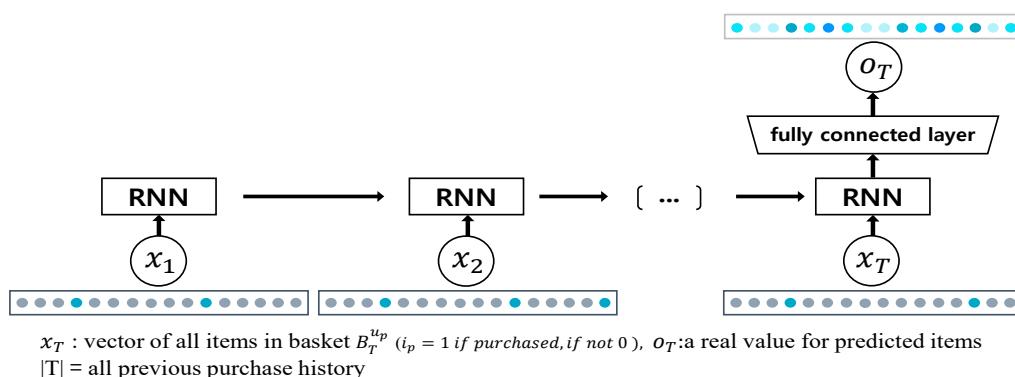
Figure 2. The proposed RNN-based recommendation model.

3.3. Multi-period Recommender Systems

In this study, the proposed RNN-based recommendation model is evaluated by multiple periods observing the performance of various recommendation time-steps. Traditional recommender systems study only focus on the model accuracy of the next point in time when recommending items to the customer and evaluating model performance, which means they evaluate the accuracy of the model only once. Indeed, customers' preferences may change over time which may degrade the performance of the recommendation model. So, in this study, recommendation periods are segmented as various time-steps, and the proposed RNN-based recommendation model is evaluated by multiple periods in a time sequence. To be more precise, multi-period recommender systems evaluate the performance not only at time point T but also the subsequent time points such as T + 1, T + 2, and so on. Figure 3 as below shows the example of a multi-period recommender system.

item recommendation with multi periods(T, T+1, ...)
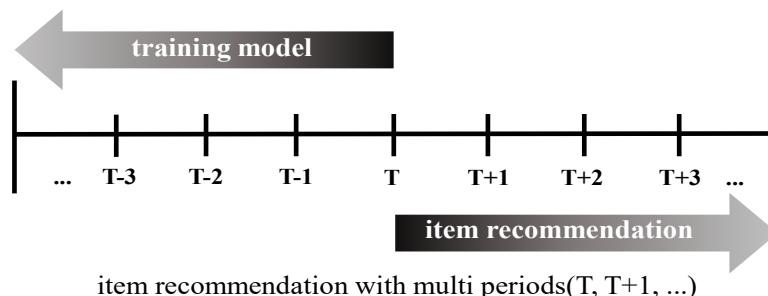
Figure 3. Example of the multi-period recommender systems.

3.4. Evaluation Metrics

Most of the recommender systems measure the accuracy of the recommended items because if the accuracy is not high, it means that the recommended items were not consumed by the users. To measure the accuracy, recall, precision and *F*1 metrics are widely used in previous studies [4,11,12,18,31]. In this paper, *F*1 measure is used to measure the accuracy of recommender systems because it considers both precision and recall to compute the score. On the other hand, if the recommendation systems recommend similar items each time, there is a risk of reducing the diversity of the entire consumer, and if the similar items are recommended every time, the satisfaction of the recommendation systems will decrease. Lathia et al. [32] mentioned that diversity should be pursued while maintaining a certain level of accuracy to increase satisfaction with the recommendation systems, and the following diversity metric is suggested.

$$\text{diversity}(\text{L1}, \text{L2}, \text{ N}) = \frac{|L2 - L1|}{N} \tag{1}$$

*L*1 and *L*2 are the recommended list and N is the number of recommended items. In this study, this temporal diversity metric is also used to measure the diversity of the recommended list.

4. Evaluation

4.1. Data Description

The data used in this study is transaction data from Fresh Food Delivery Service Company in USA, published in 2017 at the data analysis competition platform Kaggle. As mentioned above, its products' prices are generally low and customers habitually purchase. Therefore, it is a good data set for our experiment because our methodology considers repurchase behavior that other recommender systems are not interested in. Moreover, the transactional data was collected for one year, so we could ignore seasonal factors that could affect model building and performance. The data provides real purchase information for each customer and the order number which is indicated by an index assigned to each customer according to the order of purchase of items. For experiments, the purchased items by customers are arranged according to the order of purchase time, and finally, all the buying information corresponding to the same order number is composed of the same shopping list. In order to compare the recommended performance of the models over time, we used 7716 customers' shopping information with 10 shopping carts. Generally, recommender systems based on these transaction data could not infer with the preference of extremely popular items because they are products almost everyone buys. On the other hand, because sales sub-products are usually purchased by customers with unusual tastes, they could be outliers to build a model. For these reasons, a total of 9073 items were used, except for items that appeared too often or appeared less frequently, so we excluded the top and lower 10% of the sales volume for the experiments. As mentioned earlier, in this study, a recommendation model is measured by multiple recommendation periods. For this purpose, at the recommendation time point T, all the information before point T is regarded as training data and the subsequent buying information is considered as test data.

4.2. Experimental Setup

As with most studies on neural networks, we have also experimented various recurrent neural networks structure (basic RNN, LSTM, GRU) and parameters to be used as a recommendation model. First, we experiment with the number of hidden nodes in the basic RNN, LSTM, and GRU. As the number of hidden nodes increases, the initial learning rapidly progresses and quickly converges. However, since the number of parameters to learn increases as the number of hidden nodes increases, the optimal number of hidden nodes is set to 100. Also, since LSTM is slightly better than basic RNN and GRU, LSTM is used as a final model of recommendation in this study. Since the increase in the number of layers does not contribute to the improvement of our suggested model performance, so the number of hidden layers is set to one in our LSTM structure. Also the optimization function

should be determined by experiment, as it is known that there is no optimization function that fits into all problems so it should be set on an experimental basis. Figure 4 shows the experimental results of various optimization algorithms. In this figure, when an entire dataset was passed through the neural network model, an epoch was complete. For example, 10 epochs mean that an entire dataset was passed through the model 10 times. Therefore, as the epoch increases, the loss value decreases. Category cross-entropy was used as a loss function. Among optimization algorithms, Adam optimizer which shows the best performance is selected, and the hyper-parameter is set by the value known as the best default value.
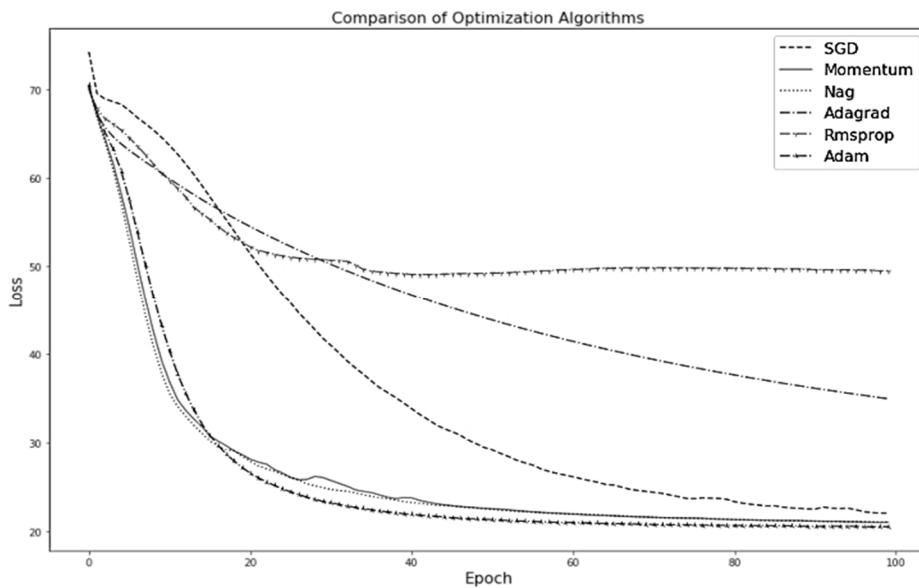
Figure 4. Comparison of various optimization algorithms.

4.3. Experimental Results

The proposed LSTM-based recommendation model and the comparison recommendation model are examined using accuracy metrics when the top five items are recommended in the multi-period perspective. First, we analyze the models by dividing time from various recommendation periods and examining whether the accuracy changes as periods change over time. The comparison model is an item-based CF and popularity model. Item-based CF (represented by CF) is a similarity-based recommendation model that recommends items which are similar to the items purchased by the target user. Popularity model (represented by POP) is a recommendation model that recommends the best-selling items, which is simple but widely applied in many circumstances.

Experiment I: In this experiment, the model is trained with five shopping items for all customers, which are contained from the first to fifth shopping lists. Then the recommendation model is used to predict what customers will purchase in multi-period perspective. The recommendation model is trained at time T with the purchasing history up to time T-1. So, the recommended item list after time T, which is denoted by T+1, T+2 and so on, is also predicted using the model trained at time T. Since each recommendation model, which is trained until time T, is examined from the multi-period perspective, and it is denoted as MP (multi-period) in this study. Figure 5 shows the experimental method and the result of recommendation accuracy measured from a multi-period perspective using a $F1$ score when recommending the top five items.

(**a**) Multi-period (MP) method

(**b**) *F*1 score
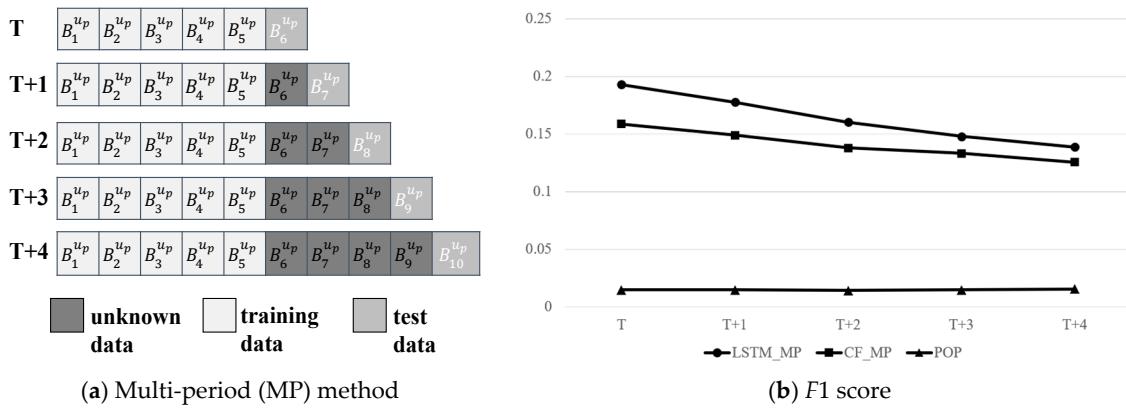
Figure 5. Training and testing method of MP and evaluation results of the MP method.

The experimental results evaluated by multiple periods from T to T+4 in a time sequence show that the accuracy of the proposed model and the comparison model decreases over time. Overall, the accuracy of the popularity model is as low as about 1%. The popularity model is like a mass marketing strategy which recommends the best-selling items and not a model that provides a personalized recommendation. This result shows that the personalized recommendation model can more satisfy consumer satisfaction. As shown in Figure 5, the *F*1 score of the proposed LSTM-based recommendation model is higher than that of the CF-based model, and is about 21% higher at T and about 10% higher at T+4. These results show that the LSTM-based recommendation model considering purchase order not only recommends items more accurately than CF-based model, but also predicts more accurately from the multi-period perspective.

Experiment II: In experiment I, the model was trained until time T, and the recommendation results were examined from the multi-period perspective. Experiment II is conducted assuming that the actual purchase information at time T+1, T+2 and so on for all customers is known. In other words, the recommendation model is trained again as the time changes. In these experiments, a moving window method is used which means that the latest five shopping lists are used to train the recommendation model, denoted as MOV (MOVing window) in this study. Figure 6 shows the MOV method (a) and the resulting *F*1 score (b) when the model recommends top five items.

(**a**) Moving window method

(**b**) *F*1 score
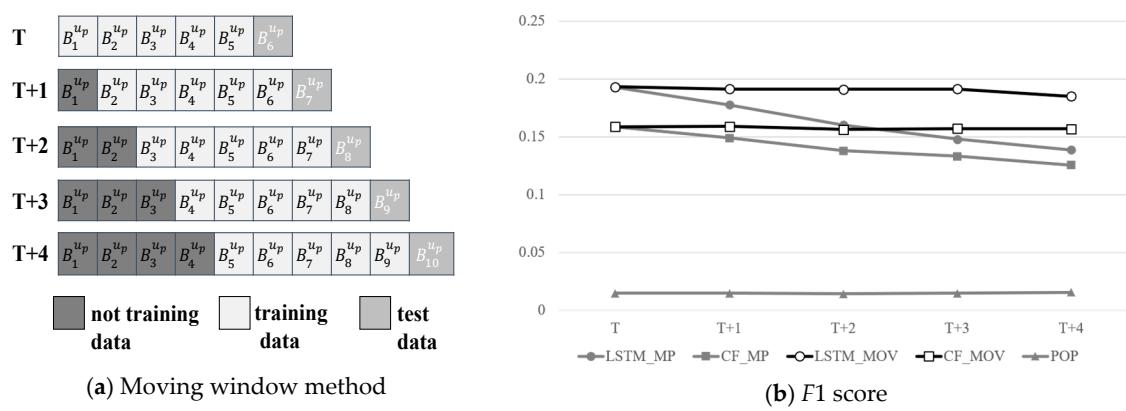
Figure 6. MOV method and evaluation results of MP and the MOV method.

As can be expected, when the recommendation model learns the actual purchase information, it shows higher accuracy than the recommendation model fixed at T point. The popularity model also shows about 1% accuracy, but with slightly more accuracy using the moving window method. The *F*1 score of the proposed LSTM-based recommendation model with the MOV method is higher than that of the MP method by about 8% at T+1, 19% at T+2, 29% at T+3 and 33% at T+4. On the other hand, the

CF-based recommendation model with the MOV method is higher than the MP method at about 7% at T+1, 13% at T+2, 18% at T+3 and 25% at T+4. These results show that the proposed LSTM-based recommendation model predicts more accurately what customer prefers than the CF-based model. Also, this result implies that customers' preference is changing over time, because the result of moving window is more accurate than that of multi-period method.

Experiment III: In experiment I and II, the training data is set to only five items. To identify whether there is data recency effect, experiment III is conducted, where all previous transaction data at recommending time T are used as training data. We denote this cumulative method as CUM and MOV of LSTM model and CF model.

Experimental results show that the overall F1 score of LSTM-based model is higher than that of the CF-based model. But the graph in Figure 7b shows a different pattern between the LSTM-based model and CF-based model. In the CF-based recommendation model, the moving window method, which learns data of specific window size only, is more accurate than the cumulative method. However, there is no significant difference between the moving window method and the cumulative method in the proposed LSTM-based model. It implies that the LSTM-based model has little effect on data recency compared to the CF-based model. The reason is that the LSTM model could consider long and short-term memory in the process of training process using customers' purchase order.

(**a**) Cumulative method　　　　　　　　　　　　　　　(**b**) *F*1 score
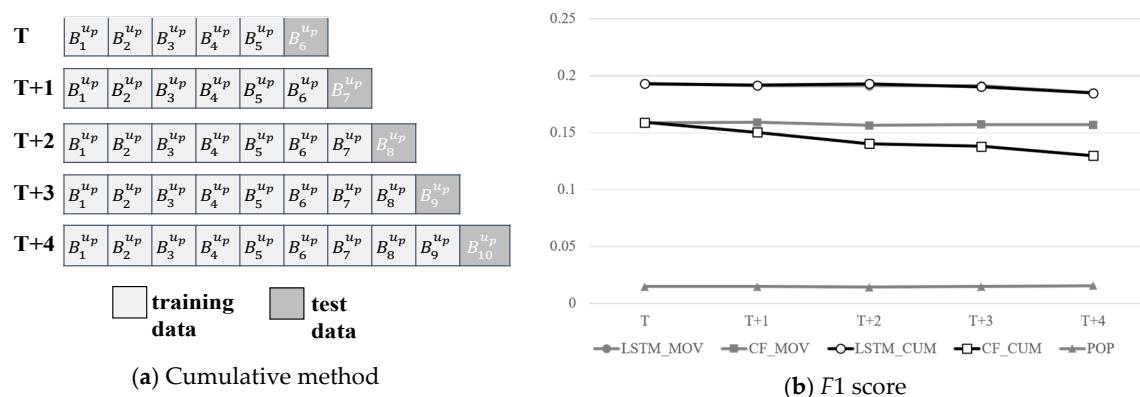
Figure 7. CUM method and evaluation results of MOV and CUM method.

Table 1 shows the diversity results of the LSTM-based recommendation model and the CF-based recommendation model. The diversity measure used in this experiment has a value ranging from 0 to 1. If the score is closer to one, it means that the result of the recommendation model is more diverse between the previous recommended item list and the subsequent recommended items list.

Table 1. Diversity results of the CUM and MOV method.

Recommended Period	LSTM_CUM	LSTM_MOV	CF_CUM	CF_MOV
from T to T+1	0.836	0.838	0.302	0.364
from T+1 to T+2	0.838	0.838	0.291	0.361
from T+2 to T+3	0.838	0.839	0.288	0.362
from T+3 to T+4	0.841	0.839	0.289	0.364

The experimental result shows that the CF-based model recommends almost similar items between the previous and subsequent recommend time, while the proposed LSTM-based model recommends more diverse items. Although the CF-based model recommends more diverse items when using the moving window method compared to the cumulative method, the LSTM-based model has no significant difference between the moving window method and the cumulative method. However, since the moving window method can utilize transaction data more efficiently, so it is thought to be a more effective learning method than the cumulative method, if there is not a significant difference

in recommendation accuracy. Furthermore, the diversity of the proposed model is relatively higher even in the condition that items purchased previously are also recommended. This result implies that the LSTM-based model not only considers well customers' purchasing orders but also captures well customers' purchasing patterns so it could suggest more diverse items.

Experiment IV: The results of the previous experiments show that when knowing the actual purchasing information, it is helpful to recommend more accurate items. But in the real world, when recommending items at time T+1, T+2 and so on, the actual purchasing information is unknown like the multi-period method. Also, as shown above, in the multi-period method, the results of the recommendation model at time T is also used at a subsequent recommendation time (T+1, T+2 and so on), which leads to degrading of the recommendation quality over time in a long-term perspective. Therefore, we propose another method of predicting and recommending items that are expected to be purchased from a long-term perspective when the customer is going to buy the recommended items of the model. We assume that the customer purchased the top five items recommended by the model at the previous time. Then, the top five items are used as training data to retrain the model at the next time. For the comparison with the multi-period method, this method which uses both all the previous transaction data and the model's top five recommended items as training data is called the cumulative multi-period method, which is denoted as CUMMP in this study. On the other hand, the moving window multi-period method, which is denoted as MOVMP, excludes the oldest transaction data as training data. MOVMP uses both only the latest transaction data and the model's top five recommended items as training data. As shown in Figure 8, the actual purchase information after the time T is unknown as the multi-period method. However, in the CUMMP and MOVMP method, the unknown data are replaced with the model's recommended top five items. Figure 9 shows the comparison results of the proposed LSTM-based recommendation model and the CF-based recommendation model, specifically the *F*1 score, among the MP, CUMMP and MOVMP method.
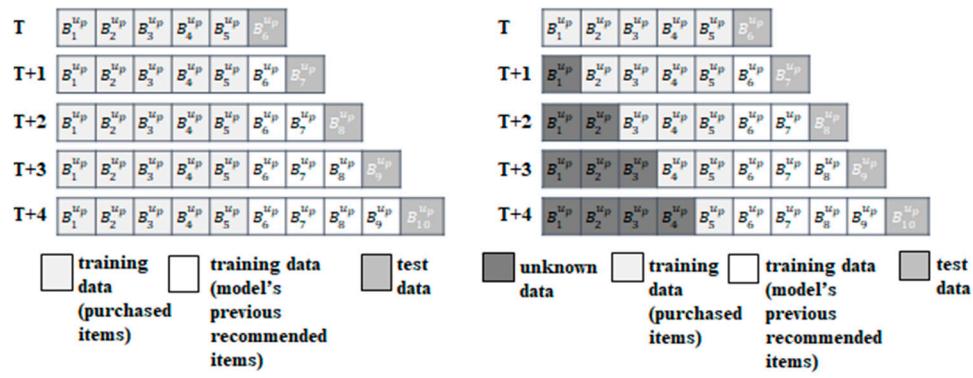
Figure 8. Cumulative multi-period (CUMMP) and moving window multi-period (MOVMP) method.

As can be expected, both the proposed LSTM-based recommendation model and CF-based model show that in terms of long-term prediction recommendation accuracy decreases with time. The LSTM-based recommendation model has about a 28%, 28%, 29% accuracy decline in MP, CUMMP and MOVMP methods respectively. Meanwhile, the CF-based recommendation model has about a 21%, 24%, 24% accuracy decline in MP, CUMMP and MOVMP methods respectively. The LSTM-based model shows a slightly larger decline, but the recommendation accuracy is still high. Also, there is no significant difference among MP, CUMMP and MOVMP methods in the proposed LSTM-based model, but the CF-based model shows that the accuracy of the CUMMP and MOVMP methods is found to be further decreased than the MP method. Since the CF-based model using CUMMP or MOVMP methods uses the model's recommended items (together with purchased items) as training data, it can be interpreted that the CUMMP or MOVMP methods do not reflect the changing preferences of customers compared to MP. On the other hand, while the accuracy of the LSTM-based recommendation

model is high even in the CUMMP or MOVMP methods compared to CF, however, it is found that there is no significant difference compared to the MP method. From experiment IV, we conclude that the LSTM-based recommendation model gives robust results compared to CF-based model, because the LSTM-based model gives almost the same result with purchase data only and purchase data with predicted data.
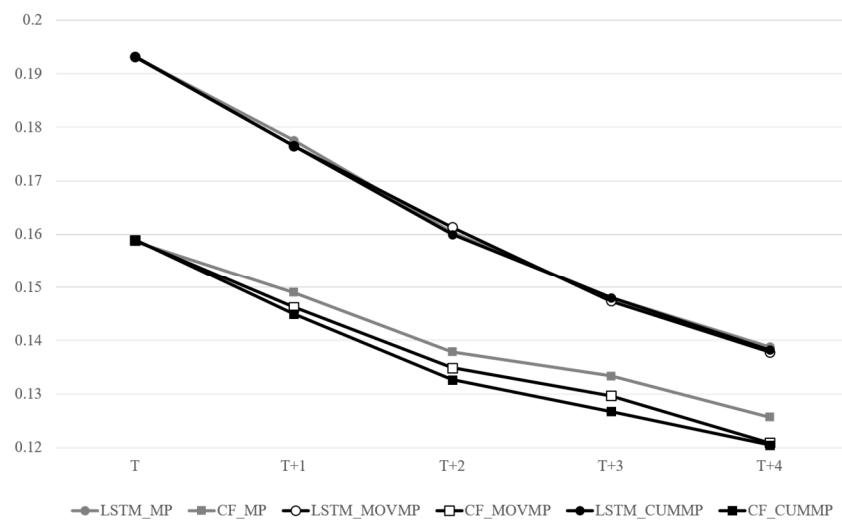
Figure 9. Evaluation results of MP, CUMMP and MOVMP by *F*1 score.

5. Conclusions

In this study, Recurrent Neural Network, which is specialized in time series data analysis, is applied to recommendation model. The proposed LSTM-based recommendation model is comparatively evaluated with the following two models; item-based collaborative filtering model, which is widely used in the recommender systems research as a benchmark system, and the popularity model, which is relatively simple but still used in the business field. Recommendation periods are segmented as various time-steps, and the proposed LSTM-based recommendation model is evaluated by multiple periods in a time sequence. Real online transaction data of a fresh food delivery market is used as a data set and the recommendation model's performance is evaluated by accuracy and diversity from a multi-period perspective.

The experimental results are as follows. First, the LSTM-based recommendation model outperformed the CF-based model in a multi-period perspective. Precisely, the proposed LSTM-based recommendation model is about 21% higher at T and about 10% higher at T+4. This result shows that considering the purchase order not only helps to improve the recommendation quality but also gives a more accurate prediction in multi periods. In addition, the proposed LSTM-based recommendation model recommends more diverse items than the CF-based model. It implies that the proposed model captures the customers' purchase patterns well and offers various items to customers. Also, experimental results show that the proposed model has no significant difference in model accuracy regardless of the size of the training data, which represents the robustness of the proposed model. However, even if there is no significant difference in recommendation accuracy and diversity, it is better to use the data efficiently through the moving window method, considering the cost of learning the LSTM-based recommendation model. Finally, in the perspective of long-term predictions, both the LSTM-based model and the CF-based model results in decreasing recommendation accuracy over time, but the accuracy curve of the LSTM-based model is of more gradual descent than that the CF-based model.

This study extended the recommendation periods as various time-steps and evaluated the performance by multiple periods in a time sequence. Unlike previous recommender systems researches,

which focus on recommendation accuracy at the single point of view, we compare the accuracy of the recommendation model with multiple periods and show that the proposed model has a better performance even at a multi-period perspective.

In applying the suggested recommendation methodology in the real market, it will be necessary to set the update frequency of the recommended model. In the actual stage of operation, the model is suggested to be updated daily or weekly to reflect the changing preferences of users to increase the accuracy and diversity of recommendation results. But the exact update frequency is to be decided by several experiments with a real data set, and may be different from the characteristic of items, number of items, number of customers, and the average time between two sequenced purchases.

In terms of sustainability, highly accurate multi-point recommendations that reflect changing customer preferences can help market managers prevent products with a very short shelf life, such as fresh vegetables, from being discarded. Furthermore, as our suggested multi-point recommender system is a kind of decision support system, it could help customers to make rapid routine decisions and save their time and money. Moreover, we expect that our study contributes to the customers' reduction of food wastes by inducing planned consumption.

In this study, online-based retail transaction data are used, but applying the LSTM-based recommendation model to more diverse transaction data set is a promising future research area. In addition, it will be also a promising research topic to combine the CF-based model with the proposed RNN-based recommendation model.

Author Contributions: Conceptualization, H.I.L. and J.K.K.; Data curation, I.Y.C. and H.S.M.; Formal analysis, H.I.L.; Investigation, H.I.L. and I.Y.C.; Methodology, H.I.L. and J.K.K.; Resources, I.Y.C. and H.S.M.; Software, H.S.M.; Validation, H.I.L, I.Y.C., H.S.M. and J.K.K.; Visualization, I.Y.C.; Writing—original draft, H.I.L.; Writing—review & editing, J.K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a grant from Kyung Hee University in 2019. (KHU-20191247).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Su, X.; Khoshgoftaar, T.M. A Survey of Collaborative Filtering Techniques. *Adv. Artif. Intell.* **2009**, *2009*, 1–19. [CrossRef]
2. Schmittlein, D.C.; Peterson, R.A. Customer Base Analysis: An Industrial Purchase Process Application. *Mark. Sci.* **1994**, *13*, 41–67. [CrossRef]
3. Goldberg, D.; Nichols, D.; Oki, B.M.; Terry, D. Using collaborative filtering to weave an information tapestry. *Commun. ACM* **1992**, *35*, 61–70. [CrossRef]
4. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Analysis of recommendation algorithms for e-commerce. In Proceedings of the EC00: The 2nd ACM Conference on Electronic Commerce, Minneapolis, MN, USA, 1–31 October 2000; Association for Computing Machinery: New York, NY, USA, 2000; pp. 158–167. Available online: https://dl.acm.org/doi/pdf/10.1145/352871.352887 (accessed on 2 July 2019).
5. McAfee, A.; Brynjolfsson, E. Big data: The management revolution. *Harv. Bus. Rev.* **2012**, *90*, 60–68. [PubMed]
6. Einav, L.; Levin, J.; Popov, I.; Sundaresan, N. Growth, Adoption, and Use of Mobile E-Commerce. *Am. Econ. Rev.* **2014**, *104*, 489–494. [CrossRef]
7. Liu, D.-R.; Lai, C.-H.; Lee, W.-J. A hybrid of sequential rules and collaborative filtering for product recommendation. *Inf. Sci.* **2009**, *179*, 3505–3519. [CrossRef]
8. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
9. Schmidhuber, J.; Wierstra, D.; Gomez, F.J. Evolino: Hybrid neuroevolution/optimal linear search for sequence prediction. In Proceedings of the 19th International Joint Conference on Artificial Intelligence, Edinburgh Scotland, 30 July–5 August 2005; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2005; pp. 853–858. Available online: https://mediatum.ub.tum.de/doc/1290202/file.pdf (accessed on 19 July 2019).

10. Sutskever, I.; Martens, J.; Hinton, G.E. Generating text with recurrent neural networks. In Proceedings of the 28th International Conference on Machine Learning, Washington, DC, USA, 28 June–2 July 2011; pp. 1017–1024. Available online: https://www.cs.utoronto.ca/~{}ilya/pubs/2011/LANG-RNN.pdf (accessed on 16 July 2019).

11. Hu, Y.; Koren, Y.; Volinsky, C. Collaborative Filtering for Implicit Feedback Datasets. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 263–272. Available online: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4781121 (accessed on 23 July 2019).

12. Jannach, D.; Zanker, M.; Felfernig, A.; Friedrich, G. *Recommender systems: An introduction*; Cambridge University Press: Cambridge, UK, 2010.

13. Cho, Y.H.; Kim, J.K. Application of Web usage mining and product taxonomy to collaborative recommendations in e-commerce. *Expert Syst. Appl.* **2004**, *26*, 233–246. [CrossRef]

14. Cho, Y.H.; Kim, J.K.; Kim, S.H. A personalized recommender system based on web usage mining and decision tree induction. *Expert Syst. Appl.* **2002**, *23*, 329–342. [CrossRef]

15. Lawrence, R.D.; Almasi, G.S.; Kotlyar, V.; Viveros, M.; Duri, S.S. Personalization of supermarket product recommendations. In *Applications of data mining to electronic commerce*; Springer: Boston, MA. USA, 2001; pp. 11–32.

16. Burke, R. Hybrid Web Recommender Systems. In *The Adaptive Web*; Springer Science and Business Media LLC: Berlin, Germany, 2007; pp. 377–408.

17. Papagelis, M.; Plexousakis, D.; Kutsuras, T. Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences. In *SOFSEM 2020: Theory and Practice of Computer Science*; Springer Science and Business Media LLC: Berlin, Germany, 2005.

18. Sarwar, B.; Karypis, G.; Konstan, J.; Reidl, J.; Riedl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the WWW01: Hypermedia Track of the 10th International World Wide Web Conference, Hong Kong, 1–31 May 2001; Association for Computing Machinery: New York, NY, USA, 2001; pp. 285–295. Available online: https://dl.acm.org/doi/pdf/10.1145/371920.372071 (accessed on 1 July 2019).

19. Song, H.S.; Kim, J.K.; Kim, S.H. Mining the change of customer behavior in an internet shopping mall. *Expert Syst. Appl.* **2001**, *21*, 157–168. [CrossRef]

20. Liu, N.N.; Zhao, M.; Xiang, E.; Yang, Q. Online evolutionary collaborative filtering. In Proceedings of the fourth ACM conference, Barcelona, Spain, 1–30 September 2010; Association for Computing Machinery (ACM): New York, NY, USA, 2010; pp. 95–102. Available online: https://arxiv.org/abs/1406.1078 (accessed on 25 July 2019).

21. Sun, J.Z.; Parthasarathy, D.; Varshney, K.R. Collaborative Kalman Filtering for Dynamic Matrix Factorization. *IEEE Trans. Signal Process.* **2014**, *62*, 3499–3509. [CrossRef]

22. Vinagre, J.; Jorge, A.M. Forgetting mechanisms for scalable collaborative filtering. *J. Braz. Comput. Soc.* **2012**, *18*, 271–282. [CrossRef]

23. Matuszyk, P.; Vinagre, J.; Spiliopoulou, M.; Jorge, A.M.; Gama, J. Forgetting techniques for stream-based matrix factorization in recommender systems. *Know. Inf. Syst.* **2018**, *55*, 275–304. [CrossRef]

24. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]

25. Cho, K.; Van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics (ACL): Seattle, DC, USA, 2014; pp. 1724–1734. Available online: https://arxiv.org/abs/1406.1078 (accessed on 8 July 2019).

26. Zhang, Y.; Dai, H.; Xu, C.; Feng, J.; Wang, T.; Bian, J.; Wang, B.; Liu, T.Y. Sequential Click Prediction for Sponsored Search with Recurrent Neural Networks. In Proceedings of the AAAI'14 Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec, Canada, 27–31 July 2014; pp. 1369–1375. Available online: https://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8529/8581 (accessed on 25 July 2019).

27. Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; Tikk, D. Session-based recommendations with recurrent neural networks. In Proceedings of the 4th international conference on learning representations, San Juan, Puerto Rico, 2–4 May 2015; Available online: https://arxiv.org/pdf/1511.06939.pdf (accessed on 19 July 2019).

28. Tan, Y.K.; Xu, X.; Liu, Y. Improved Recurrent Neural Networks for Session-based Recommendations. In Proceedings of the DLRS 2016: Workshop on Deep Learning for Recommender Systems, Boston, MA, USA, 1–30 September 2016; Association for Computing Machinery (ACM): New York, NY, USA, 2016; pp. 17–22. Available online: https://dl.acm.org/doi/abs/10.1145/2988450.2988452 (accessed on 8 July 2019).

29. Song, Y.; Elkahky, A.M.; He, X. Multi-Rate Deep Learning for Temporal Recommendation. In Proceedings of the SIGIR '16: The 39th International ACM SIGIR conference on research and development in Information Retrieval, Pisa, Italy, 1–31 July 2016; Association for Computing Machinery (ACM): New York, NY, USA, 2016; pp. 909–912. Available online: https://dl.acm.org/doi/abs/10.1145/2911451.2914726 (accessed on 12 July 2019).

30. Yu, F.; Liu, Q.; Wu, S.; Wang, L.; Tan, T. A Dynamic Recurrent Model for Next Basket Recommendation. In Proceedings of the SIGIR '16: The 39th International ACM SIGIR conference on research and development in Information Retrieval, Pisa, Italy, 1–31 July 2016; Association for Computing Machinery (ACM): New York, NY, USA, 2016; pp. 729–732. Available online: https://dl.acm.org/doi/abs/10.1145/2911451.2914683 (accessed on 30 July 2019).

31. Herlocker, J.L.; Konstan, J.A.; Terveen, L.G.; Riedl, J.T. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **2004**, *22*, 5–53. [CrossRef]

32. Lathia, N.; Hailes, S.; Capra, L.; Amatriain, X. Temporal diversity in recommender systems. In Proceedings of the SIGIR '10: The 33rd International ACM SIGIR conference on research and development in Information Retrieval, Geneva, Switzerland, 1–31 July 2010; Association for Computing Machinery (ACM): New York, NY, USA, 2010. Available online: https://dl.acm.org/doi/pdf/10.1145/1835449.1835486 (accessed on 1 August 2019).