Deep Learning Models for Serendipity Recommendations: A Survey and New Perspectives

ZHE FU, XI NIU, and MARY LOU MAHER, University of North Carolina at Charlotte, USA

Serendipitous recommendations have emerged as a compelling approach to deliver users with unexpected yet valuable information, contributing to heightened user satisfaction and engagement. This survey presents an investigation of the most recent research in serendipity recommenders, with a specific emphasis on deep learning recommendation models. We categorize these models into three types, distinguishing their integration of the serendipity objective across distinct stages: pre-processing, in-processing, and post-processing. Additionally, we provide a review and summary of the serendipity definition, available ground truth datasets, and evaluation experiments employed in the field. We propose three promising avenues for future exploration: (1) leveraging user reviews to identify and explore serendipity, (2) employing reinforcement learning to construct a model for discerning appropriate timing for serendipitous recommendations, and (3) utilizing cross-domain learning to enhance serendipitous recommendations. With this review, we aim to cultivate a deeper understanding of serendipity in recommender systems and inspire further advancements in this domain.

CCS Concepts: • **Information systems** → **Recommender systems;**

Additional Key Words and Phrases: Deep learning, recommendation models, serendipity recommendations

ACM Reference format:
Zhe Fu, Xi Niu, and Mary Lou Maher. 2023. Deep Learning Models for Serendipity Recommendations: A Survey and New Perspectives. *ACM Comput. Surv.* 56, 1, Article 19 (August 2023), 26 pages.
https://doi.org/10.1145/3605145

1 INTRODUCTION

Recommender systems are designed to mitigate information overload by offering personalized recommendations based on user preferences. In recent years, deep learning has attracted wide interest for its performance and its capacity to learn feature representations. With the wide applications of deep learning methods, tremendous success has been achieved in recommender systems to capture complex user-item relationships from data and provide accurate recommendations toward users' preferences (e.g., [1–4]). However, these deep learning based recommender systems often suffer from a narrow focus on users' historical behavior and excessive pursuit of recommendation accuracy (precision on relevance, recall on relevance, etc.). Consequently, these systems tend to overlook unexpected yet valuable items, and only recommend items that users are already familiar

This research was supported by National Science Foundation (NSF) award 1910696.
Authors' address: Z. Fu, X. Niu, and M. L. Maher, University of North Carolina at Charlotte, 9021 University City Boulevard, Charlotte, NC 28223-0001; emails: {zfu2, xniu2, M.Maher}@uncc.edu.

0360-0300/2023/08-ART19 $15.00
https://doi.org/10.1145/3605145

with, which might make them feel bored and dissatisfied [5–7]. This problem parallels the "filter bubble" phenomenon observed in news agencies [8], where users are exposed only to information that aligns with their viewpoints, leading to a repetitive stream of homogeneous recommendations [9]. Additionally, recommender systems often neglect unpopular or "long-tail" items [10] buried among a vast array of options, favoring popular items for better algorithmic accuracy [11]. Although the systems increasingly enhance the recommendation accuracy, the results are restricted to a limited scope of items, preventing those "long tail" items from being discovered by the users, which may hurt both users' satisfaction and systems' performance in the long run.

To address the challenges posed by the filter bubble and the under-representation of long-tail items, the concept of serendipity is advocated by many researchers for a richer set of information resources for the users. The word *serendipity* was created in 1754 to describe unexpected but valuable discoveries [12] and has gradually became popular after the 1950s [13]. In the early 2000s, serendipity was first introduced to the context of recommender systems [14] to broaden users' selections and increase their satisfaction [15]. Although currently there is no consensus on the definition of serendipity in the context of recommender systems, we believe that most researchers explain this concept with four components: unexpectedness, novelty, diversity, and relevance. Numerous studies have explored integrating these components into recommendation algorithms, including content-based and collaborative filtering algorithms employing techniques such as naive Bayes, *k*-nearest neighbors, and random forest models. In recent years, to discover users' potential preferences, there are increasing demands in deeply mining the relationships between users and items. Under those demands, the capacity and the advantages of deep learning models to learn multi-level representations [16] have gained great attention, triggering the adoption of deep learning in serendipity-oriented recommendations. Various deep learning models, such as **Multi-Layer Perceptron (MLP)**, the recurrent neural network, the **Convolutional Neural Network (CNN)**, CapsNet (the capsule neural network), and deep reinforcement learning, are used to learn the users' potential needs for serendipity. Although deep learning techniques are powerful in accuracy-based recommendation tasks, we believe that it is challenging to leverage them in serendipity-oriented recommendations due to four key challenges: lack of consensus in the concept definition, difficulty in representing serendipity in deep learning algorithms, lack of available ground truth data, and no well-established evaluation methods.

There are several existing survey papers on serendipity recommendations. Kaminskas and Bridge [17] conducted a survey on the definitions of diversity, novelty, coverage, and serendipity, and reviewed the corresponding models in the literature. Kotkov et al. [18] summarized the existing approaches to serendipity in recommender systems at that time and the evaluation strategies to these approaches. Ziarani and Ravanmehr [19] reviewed relatively more recently published papers from 2013 to 2019 on serendipitous recommendation methods, most of which are traditional machine learning models rather than deep learning. Abbas and Niu [20] also reviewed research studies up to 2019 on serendipity recommendations. Compared with those previous works, this survey contributes a systematic review of serendipity recommendations using deep learning techniques, proposing a classification scheme for deep learning models, and outlining future research directions in the field. To summarize, the key contributions of this article are threefold:

- A systematic review of recent research efforts of serendipity recommendations through deep learning techniques. Various aspects, such as serendipity's conceptual definition, computational operationalization, deep learning model development, and evaluation experiments, are included.
- A classification scheme is proposed to categorize the existing deep learning models into three types: pre-processing, in-processing, and post-processing. In each type, some subtypes

Table 1. Various Definitions of Serendipity

Component	Definition of Serendipity	Study	Year
Unexpectedness (Surprise)	Being unexpected and relevant	Fu et al. [21]	2023
	Being both unexpected and useful	Afridi [22]	2018
	The relationship between entity and the query that has not been discovered by the user	Huang et al. [23]	2018
	Items not included in users' previous purchases and depart from their expectations	Li et al. [24]	2020
	Items different from the users' selected items of interests	Inoue and Tokumaru [25]	2020
	Discoveries by accidents for the things users were not in quest for	Wang and Chen [26]	2021
	Difference between an item and a user profile	Zhang et al. [27]	2021
Novelty	Points of interest (POIs) that users have not visited before	Zhang et al. [28]	2015
	The recommended item previously unknown to the user	Lu et al. [29]	2018
	Items that users have not chosen before	Wang et al. [30]	2018
	Unpopularity, items that few people are aware of	Coba et al. [31]	2018
	Unrated items that users may be interested in and satisfied with	Xu et al. [32]	2020
	Items that users would have never found or thought of by themselves	Shrestha et al. [33]	2020
	Long-existing but unpopular items	Lo and Ishigaki [34]	2021
Diversity	POIs that geographically cover the frequently visited area as much as possible	Han and Yamana [35]	2019
	Different types of venues	Ge et al. [36]	2020
	Items different from those involved in users' historical interactions	Sun et al. [37]	2020
	Items different from each other in the recommendation list	Cui et al. [38]	2020
	Recommendations that cover users' multiple interests	Chen et al. [39]	2021
	POIs from various categories	Lee et al. [40]	2022

are identified. The classification scheme provides a better understanding of current deep learning models in the serendipity recommendation research.
- Three future directions are suggested along with discussions on the challenges of each direction.

2 SERENDIPITY DEFINITION AND ITS COMPUTATIONAL OPERATIONALIZATION

Since there is no agreement on the definition of serendipity in the context of recommender systems, how to introduce an appropriate definition of serendipity for recommender systems is essential for the later computational operationalization and evaluation. Existing papers have put forward various definitions from different aspects based on their own understanding. Most definitions involve one or more of the three components: unexpectedness, novelty, and diversity. In addition to the three components, all of these studies agree on that the serendipity should be bounded. Therefore, it should contain a fourth element: relevance (or recommendation accuracy). Table 1 summarizes definitions for serendipity from different recommender systems studies. Each of these components will be discussed in the following sections.

2.1 Unexpectedness

Many researchers believe that serendipity means accidental discovery that is out of one's expectation. For example, Afridi [22] defined the concept of *serendipity* as "the quality of being both unexpected and useful." Huang et al. [23] conducted extensive experiments on large-scale datasets to reveal that unexpectedness could significantly contribute to the recommendation effectiveness. Li et al. [24] also selected the concept of unexpectedness as a crucial objective of recommendations. They thought that unexpectedness had a positive correlation with user experience and could more effectively overcome the problem of overspecialization. Additionally, some of the researchers also describe such unexpectedness using the term *surprise* [6, 18, 41–44]. As a result, in this survey, "surprise" and "unexpectedness" are considered equivalent. We will use the term *unexpectedness* in the following sections.

From the computational operationalization perspective, some researchers realized unexpectedness as the distance or dissimilarity to a user's preferences, or a deviating direction from those preferences. For example, Kotkov et al. [15] computed unexpectedness through the dissimilarity between an item and a user's rated items. Li et al. [24] constructed the representation

vector of unexpectedness in the latent space. By first clustering the user's historical records as the preferences, the distance between a candidate item and the user's preference clusters was then used to calculate the unexpectedness of the item. Huang et al. [23] quantified the unexpectedness in a search engine context as the distance between a user's query and a recommended entity, which considered both the content dissimilarity and the difference of click frequencies. Zhang et al. [27] calculated unexpectedness from the aspects of both category difference and the latent representation difference between a candidate item and a user's profile. For the category difference, they explored the categories that shared the least related items with the user's historical records. For the latent representation difference, by constructing representation vectors in a latent space, the researchers calculated the dissimilarity between the candidate item and the user's profile.

Additionally, there are many researchers who computationally realized unexpectedness based on calculating co-occurrence. Cleverley and Burnett [45] explored the usefulness of the word *co-occurrence* in **Information Retrieval (IR)** and leveraged it to facilitate serendipitous encounters during exploratory searches. Inspired by "bag of words" in text mining, Niu et al. [46–48] modeled the users' expectation on news as the expected likelihood of a particular bag of co-occurring topics of a piece of news. A lower likelihood of topic co-occurrence compared to the expected likelihood is regarded as unexpectedness. By evaluating the co-occurrence of corresponding keywords, Maake and Tranos [49] designed a research paper recommender system that recommended papers with serendipitous topics.

2.2 Novelty

Some researchers believe that a serendipitous item should be unknown to the users, and they describe serendipity using the term *novelty* [29, 33, 50]. By identifying these novel items, the system provides recommendations that are different from users' habits. For example, Zhang et al. [28] believed that there was an embedded tendency in human brains to explore novelty in dining behavior. They introduced novelty to the recommender system to stimulate users to dine out. Although it was important to serve accurate recommendations, Wasilewski and Hurley [51] agreed that novelty was another significant utility beyond accuracy. By enhancing the novelty of recommendations, they believed that it would improve users' experiences by widening the range of possible item types recommended to the users. Similarly, Xu et al. [32] integrated novelty in traditional accuracy-based recommender systems to increase users' interests in recommendations with the possible risk of sacrificing satisfaction.

The novelty of recommendations is commonly computed based on two approaches. One is finding newly listed items in the recommender system, and the other one is exploring the already existing items that are not likely to be known to a user. For newly listed items, the researchers recommended new items that were recently included in the systems without sufficient information to users. The problem of serendipity recommendation becomes a cold-start task. For example, to increase novelty in music recommendations, Mohamed et al. [52] recommended new music products to users with the intent of making the recommendations more attractive. Similarly, Deldjoo et al. [53] focused on recommending newly released movies to increase the novelty of the recommendations and help reach the goal of business-centric recommendations. Mazumdar et al. [54] described the novelty in **Point-of-Interest (POI)** recommendations as newly added POIs in the systems and treated the task as a cold-start problem. However, in addition to newly listed items, users may also feel fresh on some long-time existing items that are outside their typical reach. For example, Zhang et al. [28] regarded a novel restaurant as a restaurant that had not been a usual type of visit for a target user. To make novelty-oriented recommendations, Chen et al. [42] calculated novel items by extracting items from categories/domains outside of the user's profile

that were unlikely to be known by the user. Shrestha et al. [33] considered the novelty of an item as the dissimilarity of the item compared to what the user had previously seen before. Lo and Ishigaki [34] regarded a novel item as an unpopular item that had rare interactions among the user population and therefore was unlikely to be known by the target user.

2.3 Diversity

Diversity was introduced in serendipity recommendations to provide different and diverse kinds of items for users. Diversity is believed to increase the probability of serendipity discovery and broaden the users' preferences. Cheng et al. [55] believed that poor diversity might narrow users' horizons and make them frustrated. Therefore, diversity is now emphasized by an increasing number of researchers as a crucial element of serendipity. Diversity is usually defined as the recommendations different from users' historical records and covering a wide range of different kinds of items [35–37, 47, 56–58].

To represent and calculate the diversity, Li et al. [58] operationalized diversity as elasticity and calculated from both the user and item sides. The user elasticity means a user's acceptance range measured by the number of movie genres watched. Meanwhile, the movie elasticity means the number of different user groups that watched this movie. The combination of the user and movie elasticity represents a general diversity degree of the recommendations. Lee et al. [40] regarded diversity as recommended POIs from various categories. They calculated diversity by the number of categories that the POI recommendation list involved. Cui et al. [38] calculate the diversity level of the recommendations by the inverse value of cosine similarity of any item pair in the recommendation list.

2.4 Relevance

There is a growing consensus that the items generated by serendipity-oriented recommender systems should be relevant to users. Recommending serendipity is not recommending items randomly and should depend on users' personal preferences. It is believed that at the heart of the experience of serendipity was the emergence of powerful personal relevance out of seemingly random coincidence of events [59]. The component of relevance is to restrict the scope of serendipity, preventing unlimited recommendations from annoying users. For example, Rahman and Wilson [60] argued that although the concept of serendipity has a nature of subjectiveness and personalization, serendipity search must be personally relevant rather than from fictional tasks. Wang et al. [61] pointed out that instead of purely stressing on serendipity, a serendipity-oriented recommender system should target to balance between serendipity and relevance.

Relevance is a well-established metric in IR. In the past few decades, researchers have developed various models, from traditional statistic models to machine learning models and then to deep learning models more recently, to predict relevance between a query and a document in a search engine, and between a candidate item and a user's historical items in a recommender system. Therefore, serendipity recommenders have a rich foundation in IR to leverage to incorporate relevance as one of their components.

To sum up, although these serendipity components (unexpectedness, novelty, diversity, and relevance) may have overlaps in definitions and computations, they are widely believed to contribute to providing a serendipitous set of information resources for the users. The definition of serendipity remains ambiguous in the recommender systems community. The ambiguity could lead to different model development methods and evaluation experiment settings [19], which makes the current research lack comparability and generalizability. Without a universal definition, it is challenging to verify the effectiveness of serendipity recommenders and to sustain systematic future works on serendipity-oriented recommendation research.

3 DEEP LEARNING BASED SERENDIPITY RECOMMENDER SYSTEMS

There have been several research efforts on serendipity recommendations in the past few decades before the success of deep learning. Earlier attempts to develop tools that support serendipity included the LyricTime music system [62], a recommender system that accommodated serendipitous access by occasionally adding randomly picked songs to a user's playlist. Campos and de Figueiredo [63] developed a software agent called *Max* that incorporated the knowledge of concept association to find information that imperfectly matched a user's profile. Iaquinta et al. [64] used a machine learning model to predict the user ratings of an unseen item. Items for which the prediction was uncertain between positive and negative were considered as potentially serendipitous. Onuma et al. [65] introduced a metric called the *bridging score* for item nodes in a graph. Nodes connecting disparate subgraphs in a graph received high bridging scores and therefore had higher serendipity scores. Zhang et al. [66] used a topic modeling approach to represent each artist as a distribution of latent user clusters to find some serendipitous artists outside the user's playlist. Adamopoulos and Tuzhilin [67] derived a set of unexpected recommendations by deducting the items that were recommended by a primitive prediction. These previous studies, although not using deep learning models, all imply that carefully designed conventional machine learning models may have helped in increasing the chance of serendipitous discoveries.

In recent years, deep learning has widely been used in the research of recommender systems to address the challenge of the growing volume of online information. With the support of big data and powerful computing resources, deep learning methods have recently demonstrated huge advantages in capturing more complex attributions and more subtle relationships between users and items. Therefore, how to effectively learn the knowledge of serendipity from data has become the core research problem for deep learning based serendipity recommender systems. By exploring ways to appropriately integrate the serendipity components into established deep learning structures, various approaches have been proposed. We classify their efforts into three different types according to the stages of the serendipity incorporation: the pre-processing stage, the in-processing stage, and the post-processing stage. Under the in-processing stage, we further find two subtypes: static representation learning and dynamic representation learning. Under static representation, we further classify into unified representation learning and decomposed representation learning. Under dynamic representation learning, there are switching systems and deep reinforcement learning systems. The entire classification scheme is presented in Figure 1. This figure represents one major contribution of this survey and serves as the outline for Section 3.

3.1 Pre-Processing Stage

In the pre-processing stage, the serendipity components are considered prior to constructing the recommendation model. They typically serve as the input of a recommendation model. Some of the researchers generated a pseudo user behavior sequence before model training by inserting unexpected or novel items into it. For example, to recommend unpopular and niche items to users in a sequential recommendation context, Kim et al. [68] first made a pseudo user sequence by clustering and inserting the unpopular items to supplement the original sequential data. Then, the reorganized sequential data were fed into a **Gated Recurrent Unit (GRU)**-based model that generated recommendations for users. The approach successfully increased the diversity and preserved accuracy at the same time. Meanwhile, some other researchers engineered serendipity features as additional input data to the deep learning models. These serendipity features could be derived from some demographic or statistical information from users or items. Li et al. [69] focused on addressing the challenge of recommending long-tail hashtags of micro-videos to users. They first introduced external knowledge of the hashtags' correlation network to the raw data, which could help the model discover long-tail hashtags. They characterized the edges of the correlation network

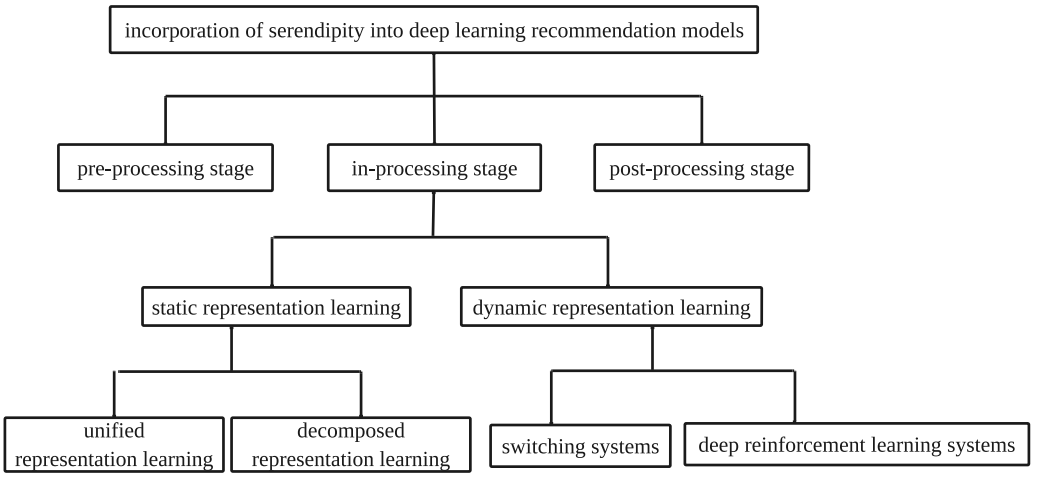


Fig. 1. Classification of the deep learning models for serendipity recommendations.

with four types of relations between the hashtags. By using these relations, the authors augmented the edges of long-tail hashtags in the network, which were then processed by a graph convolutional network [70] model for long-tail hashtag recommendations. In another work, based on the well-known word embedding algorithm GloVe [71], Grace et al. [72] proposed a new embedding approach, called *s-GloVe*, to find surprising pairs of words that do not commonly co-occur, such as chocolate and garlic, bananas and basil, and vanilla. The new embedding approach leveraged the opposite information of the word co-occurrence frequency captured by GloVe. Although the study was not for developing a recommender, the s-GloVe approach has the potential to generate input embeddings for a deep learning recommender model. Li et al. [58] defined a concept of elasticity for users and items, which could help the deep learning model discover serendipitous movies. Before training the model, they first leveraged the users' demographic information (age, occupation, etc.) and movies' statistical information (number of genres, size of user groups, etc.) to calculate the elasticity for users and movies, respectively. To be more specific, the user elasticity $E(u_i)$, which indicates the probability of the user accepting different movies, is calculated based on the range of movie genres consumed by the user:

$$E(u_i) = \frac{|G(u_i)|}{|G_{max}(U)|}, \tag{1}$$

where $|G(u_i)|$ denotes the number of genres watched by the user and $|G_{max}(U)|$ denotes the maximum possible number of genres watched by a user from the user set. Similarly, the movie elasticity $E(m_i)$, which represents the possibility of the movie being watched by different users, is calculated based on the types of different users attracted by the movie:

$$E(m_i) = \frac{D(m_i)}{D_{max}(M)}, \tag{2}$$

where $D(m_i)$ is the diversity level of the movie, calculated as follows:

$$D(m_i) = \frac{\alpha \times |A(m_i)| + \beta \times |O(m_i)| + |U(m_i)|}{\alpha + \beta + 1}, \tag{3}$$

where $|A(m_i)|$ denotes the number of age groups, $|O(m_i)|$ denotes the number of occupations, and $|U(m_i)|$ denotes the number of users who watched this movie. α and β are the hyperparameters to adjust the impact of each component. Then, the elasticity of users and movies were used as

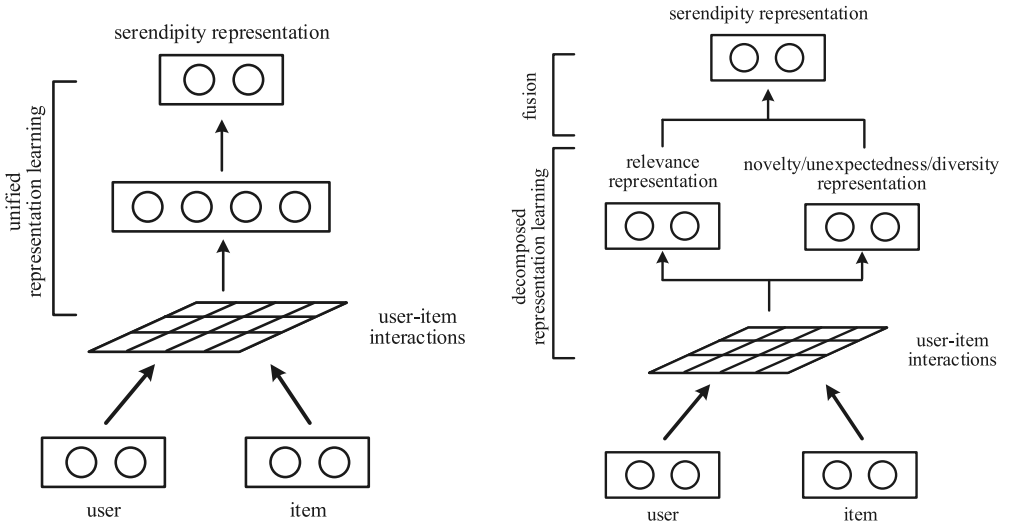


Fig. 2. Static representation learning for serendipity.

crucial input features into a GRU-based model to generate serendipitous recommendations. Their proposed method achieved significant improvements on their self-defined serendipity metrics.

The advantage of integrating the serendipity components in the pre-processing stage is that the serendipity features can be pre-calculated in advance. That way, it is easier to explain which feature plays a role in the future model results. However, the challenge is that these pre-calculated serendipity features are fixed and static, not dynamically updated during the model training. It is not ideal for the ever-changing users and the ever-changing context. As a result, in addition to the pre-processing stage, serendipity components are also incorporated with deep learning models during the training stage.

3.2 In-Processing Stage

Research has made efforts on incorporating serendipity components with deep learning models during the model training stage. The core challenge of this stage is to learn an appropriate representation vector for serendipity. The existing research mainly has two different strategies. One is to learn a static serendipity representation from users' historical behaviors, and the other is to dynamically update the representation toward serendipity using users' real-time feedback. In the following sections, we introduce the two strategies in detail.

3.2.1 Static Representation Learning for Serendipity. For static representation learning, the deep learning models are proposed to learn a static representation vector of serendipity for a user, which encodes the users' needs on serendipity. Various models are designed for the static serendipity representation. We further categorize them into two subtypes: unified representation as shown in Figure 2(a) and decomposed representations as shown in Figure 2(b).

Unified Representation. In a unified representation, deep learning models typically learn one comprehensive vector to represent the users' needs on serendipity. These models learn the knowledge of serendipity by encoding the valuable serendipity features from the users' behavior or preferences, as shown in Figure 2(a). Various models such as auto-encoder, the recurrent neural network, CNN, and the graph neural network were leveraged to represent serendipity from raw data, which

tends to be high dimensional, sparse, and full of noises. Lee et al. [73] proposed a recurrent variational auto-encoder to represent serendipity as a vector for the purpose of recommendations. The variational auto-encoder approach served as a feature extraction tool to extract signals that were helpful in representing serendipity. Li et al. [74] represented the serendipity need of each user as one vector that deviated from the short-term demands but related to the long-term preferences. In their proposed model, the users' long-term preferences were extracted from a Gaussian mixture model and their short-term demands a capsule network. Then, the serendipitous item was an item with the vector representation having a direction pointing from the short-term demand to the long-term preference as well as having an appropriate distance to the short-term demand. To be more specific, the direction of the users' serendipity vectors *serendipity_vector* is calculated by Equation (4), which flows from the users' short-term preferences *short_term_vector* to the scope of the users' long-term preferences *long_term_vector*.

$$\frac{\overrightarrow{serendipity\_vector}}{\| \overrightarrow{serendipity\_vector} \|} = \frac{\overrightarrow{(long\_term\_vector) - (short\_term\_vector)}}{\| \overrightarrow{(long\_term\_vector) - (short\_term\_vector)} \|} \tag{4}$$

Additionally, the magnitude of the serendipity vectors *serendipity_vector* is calculated based on the number of the users' long-term preferences, as shown in Equation (5), and user with a more diverse long-term preference will obtain a larger magnitude of the serendipity vector:

$$\| \overrightarrow{serendipity\_vector} \| = m_{base}(1 + num\_preference)(1 + num\_item\_cluster), \tag{5}$$

where $num\_preference$ is the number of users' long-term preferences generated by the Gaussian mixture model, $num\_item\_cluster$ denotes the number of item clusters related to users' long-term preferences, and m_{base} is a hyperparameter. The evaluation experiments show that the proposed directional serendipity vector effectively improved the serendipity identification and its interpretability.

At the same time, some researchers have noticed the advantage of the attention mechanism on automatically and selectively extracting the most valuable information from limited data [75], which helps the deep learning model pay more attention on the serendipity-related aspects in the unified serendipity representation. For example, Raza and Ding [76] aimed at preventing readers from getting bored with similar news and exposing them to different views or opinions. They introduced a diversity-aware interest learning module [77], which leveraged the attention mechanism to capture the diversity pattern from users' historical behaviors. By combining the diversity pattern and the regular user interest pattern, the model was able to learn one general representation of users' diversified interests. Similarly, Xie et al. [78] proposed a graph-based deep learning model to learn one representation for users' diverse preferences on videos. They first constructed a graph network with five different types of nodes: video, tag, media (video provider), user, and words in the video title. The graph network brought in users' diverse preference information from different types of nodes. A graph attention network was introduced to learn one representation vector for each user by aggregating all types of nodes. The attention mechanism inside the model was used to adjust the impacts of these nodes on user preference representation. Wang et al. [30] leveraged the attention mechanism to capture the users' potential interests on novelty. By calculating the impact of items in a user's historical transaction records, the attention mechanism weighed the importance of each interacted item and generated one overall representation vector for the user's historical records. The representation comprehensively considers the influence of contextual items, which can avoid recommending duplicated items and discover novel items to the users. Finally, this user representation was used to recommend novel items that were relevant to the user history but had not been chosen by the users.

Most recently, Fu et al. [21] designed a self-enhanced module called *SerenEnhance* to learn the fine-grained facets of serendipity, to enhance the generic unified representation of serendipity generated by Transformers [79]. The self-enhanced module is general enough to be applied with many base deep learning models for serendipity. A series of experiments have been conducted to show that *SerenEnhance* outperforms the state-of-the-art baseline models in predicting serendipity.

The unified representation learns a serendipity representation for a user with a relatively simple structure. The downside is that the unified representation, for most of the time, is a black box and lacks interpretability. It is not straightforward for the researchers to understand what is included and what is not included in this unified serendipity representation, and why it is effective. Additionally, without direct ground truth data on serendipity, it is difficult for the deep learning models to learn the subtlety of the serendipity features in one representation. As a result, many other researchers focus on a decomposed approach to constructing the serendipity representation in a separate and more explainable way, which we will discuss in the following section.

Decomposed Representation. Different from the unified representation, the decomposed representation decomposes serendipity into several components to represent. As shown in Figure 2(b), the decomposed framework typically consists of two representation components: one for relevance and the other for one of the serendipity components (unexpectedness, novelty, or diversity). The recommender framework will then combine the two representations through either vector aggregation or joint learning optimization. For example, to promote item novelty, Lo et al. [34] proposed a model considering both relevance and novelty with two decomposed components. On one hand, a relevance-based deep learning model (e.g., NCF [80], CMN [81], or NGCF [82]) was utilized to find relevant items to users. On the other hand, a modified Gaussian radial basis function was adopted to model the novelty tolerance in users' preferences. The novelty component was integrated with the relevance component through a joint loss function, guiding the model optimization toward both novelty and relevance. Similarly, Li et al. [24] proposed a deep learning model to learn the representation of serendipity from the aspects of relevance and unexpectedness. For the relevance part, they leveraged a GRU-based model to get the prediction on relevance. For the unexpectedness part, they estimated the unexpectedness level of each item through a self-attentive MLP, which focused on the items that were distant from the users' historical behaviors. To generate serendipitous recommendations, a utility function, which balanced the relevance and unexpectedness, was used to calculate the loss and optimize the parameters for the model:

$$Utility_{u,i} = r_{u,i} + unexpectedness_{u,i}, \tag{6}$$

where $r_{u,i}$ represents the relevance between the user u and the item i, and $unexpectedness_{u,i}$ represents the unexpectedness between u and i. Extensive offline and online experiments illustrated that their proposed model had better performance compared with state-of-the-art relevance-based prediction models. To provide novel items in the recommendations, Zou [83] proposed a deep learning model to predict the novelty level of user-item pairs. After obtaining the relevance and novelty representation simultaneously from two decomposed modules, an MLP layer was leveraged for the integration and generation of an overall vector for the prediction. This decomposed method eventually outperformed the random predictor model and other novelty-seeking models. Xu et al. [32] decomposed serendipity as unrated items that users may have high satisfaction with but low interests in. To strike a balance between satisfaction and interests, they proposed a recommender system with two parallel structures: a weighted matrix factorization based model to tackle the users' interest prediction, and an MLP-based model to capture the complex relationships between users and items for satisfaction prediction. Finally, the system combined interest prediction and satisfaction prediction to extract unrated items and generate personalized recommendations. Extensive experiments on real-world data demonstrated that their proposed model

had promising performance compared with the state-of-the-art baselines. To quantify serendipity for POI recommendations, Zhang et al. [27] also constructed decomposed representations of unexpectedness and relevance separately. After first using the stacked transformer blocks to learn the optimal representations for the users' trajectory sequences, they generated the representations for unexpectedness by using an average pooling layer to aggregate the trajectory representations and a dense layer to further process the aggregated representations. At the same time, they generated the representation for relevance by concatenating vectors of four types of user preferences: the general preference aggregated from trajectory representations, the current preference represented by the user's last visited POI, the long-term preference calculated by the temporal and geographical information of POIs, and the short-term preference generated by the user's current trajectory. Finally, the representations of unexpectedness and relevance were added together as the serendipity representation for a user:

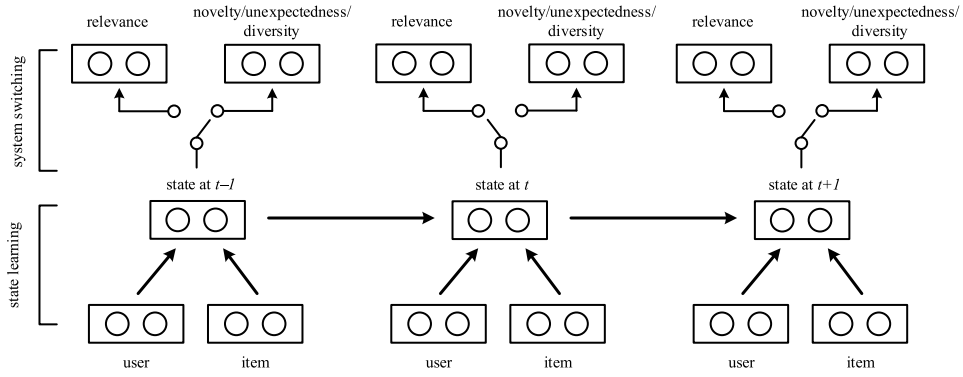
$$serendipity(u, i) = \lambda \times relevance(u, i) + (1 - \lambda) \times unexpectedness(u, i), \tag{7}$$

where $relevance(u, i)$ denotes the user's relevance representation learned from user's preference on a POI, and $unexpectedness(u, i)$ denotes the user's unexpectedness representation learned from the relations between a POI and a user profile. To combine the two components, the model further adds a hyperparameter λ as a tradeoff parameter. The model training process jointly optimizes the two components. The experimental results showed that their proposed model outperformed the state-of-the-art POI recommendation models in terms of both accuracy and serendipity. To provide diversified recommendations, Chen et al. [39] focused on extracting users' multiple interests from their sequential behaviors. A GRU-based module was proposed to first mine users' general interests to learn the representation of relevance. Then a multi-head attention mechanism was introduced to learn the users' interests from different aspects, which was used as a user's diversity representation. The recommendations were generated based on the combination of the two representations through their joint loss function.

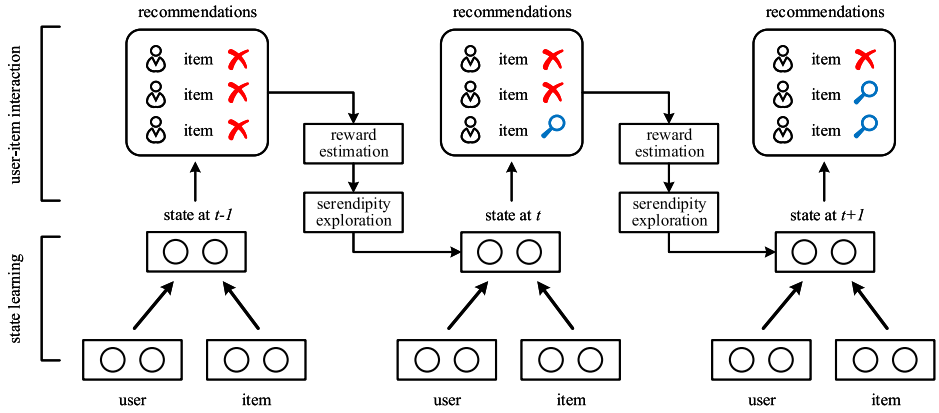
Although the decomposed approaches may generate a more explainable serendipity representation, the tradeoff between different components is usually arbitrarily set as a hyperparameter in the models or the loss functions. If the value of this tradeoff hyperparameter is set small, the final recommendations may end up being regular relevance-oriented recommendations. On the contrary, if the value of the hyperparameter is large, the recommender systems may risk recommending irrelevant items. Additionally, many researchers argue that the experiences of serendipity vary for different users. Even for the same user, the feeling of serendipity changes over time. The balance between relevance and serendipity should be calculated individually and dynamically. Some researchers believe that it may be beneficial to estimate the current state for a user: whether the user would prefer serendipitous recommendations or not, then dynamically adjust the balance between relevance and serendipity. These methods regard the serendipity-oriented recommendation as a dynamic decision-making process, which will be introduced in the following section.

3.2.2 Dynamic Representation Learning for Serendipity. Compared with the static serendipity representation, a dynamic representation could help the model first track the change of users' real-time expectations and then make a decision on recommendations. We further categorize the current models of this type into two subtypes: switching systems as shown in Figure 3(a) and reinforcement learning systems as shown in Figure 3(b).

Switching Systems. In the switching systems, the models will provide recommendations depending on a user's dynamic state, as shown in Figure 3(a). If a user's state indicates that the user is now more interested in unexpected or novel items, the model will work in the mode of serendipity recommendations. Otherwise, the model will be a regular recommender. For example,



(a) Switching systems



(b) Deep reinforcement learning systems

Fig. 3. Dynamic representation learning for serendipity.

Zhang et al. [28] recommended novel restaurants for users for their next dining options. To improve the users' satisfaction, they designed a switching recommendation system, which first estimated the users' willingness on seeking novel restaurants and then selected to model the users' behaviors for novel or regular restaurant recommendations accordingly. To be more specific, before making recommendations to users, a conditional random field model with prior knowledge was proposed to infer the novelty-seeking state of a user:

$$p(y_t|x) = \frac{1}{Z(w_k)} \exp\left(\sum_{k=1}^{K} w_k f_k(y_{t-1}, y_t, X(t))\right), \tag{8}$$

where $p(y_t|x)$ is the probability of a user being willing to visit a novel restaurant at timestep t. f_k is one latent feature extraction function, and there are K such functions. $X(t) = (X_1(t), X_2(t), \ldots, X_m(t))$ is the set of user or item explicit (not latent) input features at timestep t. w_k is a learnable weight for the k^{th} latent feature. $Z(w_k)$ is calculated as follows:

$$Z(w_k) = \sum_{y} \exp \sum_{k=1}^{K} w_k f_k(y_{t-1}, y_t, X(t)). \tag{9}$$

If a user was predicted to be novelty seeking, a context-aware model would be selected to explore novel restaurants that the user had not visited before. Otherwise, a hidden Markov model would be employed to recommend regular restaurants that users had visited in the past.

The switching systems completely split the serendipity and relevance as two separate scenarios. However, some real-life scenarios may need to blend the two and dynamically adjust the balance between them. To allow such capacity, some research introduced the well-established deep reinforcement learning approach into the serendipity recommendation research. In deep reinforcement learning, the system collects the users' real-time feedback for state updating and for serendipity recommendations.

Deep Reinforcement Learning Systems. The approach of deep reinforcement learning combines the advantages of deep learning and reinforcement learning, which dynamically updates users' current state representation from their real-time feedback, and optimizes the model parameters toward maximizing the expected benefits in the future. How the approach works is presented in Figure 3(b). Deep reinforcement learning is essentially a mathematical framework for sequential decision making, which is capable of tracking the users' shift of expectations through real-time interactions between users and the system [84].

Different from the pure deep learning models, the parameters in deep reinforcement learning are updated dynamically. Therefore, deep reinforcement learning systems are able to sensitively track the tendency of a user's expectation shift and generate a more effective state representation for serendipitous recommendations. For example, to avoid recommending similar news to users, Zheng et al. [85] proposed a deep Q-learning network based recommendation framework, which incorporated an exploration strategy to find new attractive news for users. To track the dynamic state of user preferences, they first used survival models [86, 87] to model the users' state representations based on the users' activeness on the news application. Then, a deep reinforcement learning framework was applied to generate recommendations and estimate the future rewards from the users' feedback. To enhance the serendipity level in recommendations, a double training structure with an original network and an auxiliary network was further introduced in the system. By adding uncertain noises to the original recommendation network, an auxiliary network was constructed to explore the new items and compete with the original model. As shown in Equation (10), the weight in the auxiliary network W_{aux} is generated by adding random noise $(\alpha \cdot rand(-1, 1) \cdot W_{org})$ to the weight in the original network:

$$W_{aux} = W_{org} + \alpha \cdot rand(-1, 1) \cdot W_{org}, \tag{10}$$

where α is the exploration coefficient, and $rand(-1, 1)$ denotes a random number between −1 and 1. If the auxiliary network performed better, it means that the user was more in the mood for exploring unexpected news and the original network would be replaced by the auxiliary network. In another study, to explore the songs that were far from the user's preferences, Sakurai et al. [88] proposed a reinforcement learning based recommender system to recommend playlists with high diversity for the users. By first projecting the acoustic features of each song to a two-dimensional vector using t-distributed stochastic neighbor embedding, the authors constructed a map for all of the songs in two-dimensional space. Then, a deep reinforcement learning model was leveraged to explore the songs in the map that users might be interested in. Three strategies were proposed by the authors to explore a wider range of the songs. First, they initialized the position of the starting point in the map based on the users' preferences, which could prevent the model from exploring songs that were irrelevant to the users' preferences at the beginning. Second, two types of rewards were defined in the reinforcement learning to update a user's state toward diversity. One was the discovery reward, calculated by the number of different songs discovered by the model at each timestep. The other one was the moving reward, which gave the punishment to the model once it

discovered irrelevant songs that shared no acoustic features with the user profile. Third, to avoid recommending the repetitive songs, the songs already discovered by the reinforcement learning model were removed from the map. The proposed model was proven to be efficient in generating the music playlist with high diversity. In another study, to promote the diversity and novelty of the recommendation results, Stamenkovic et al. [89] adopted the deep reinforcement learning framework to estimate the model rewards on accuracy, diversity, and novelty based on users' real-time feedback and update the users' state representation accordingly. Specifically, the reward on accuracy was defined as a binary score of whether the recommended item was clicked or not. The reward on diversity was defined as the inverse cosine similarity value between the user's last clicked item and the top predicted item from the model. Last, the reward on novelty was defined as a binary score of whether the recommended item was an unpopular item or not. By combing these three different types of rewards, the proposed deep reinforcement learning framework was able to simultaneously satisfy three objectives: improving click rates, diversifying recommendations lists, and introducing novel items. Extensive experiments were conducted on two real-world sequential e-commerce datasets and the results revealed the model effectiveness.

To sum up, deep reinforcement learning dynamically captures a user's state and the shift of the user's preferences on serendipity. However, most of the deep reinforcement learning studies model the users' historical behaviors as a **Markov Decision Progress (MDP)**, which only considers the impact of the users' immediately previous state. Although it helps decrease the computational complexity of deep reinforcement learning, this first-order MDP assumption ignores the information from users' long-term behaviors. It would be more meaningful to enhance the state representation on serendipity for users by considering both real-time feedback and longer history in the future studies.

3.3 Post-Processing Stage

In the post-processing stage, a pool of candidate items has already been identified by a deep learning model. By selecting from the candidate pool and rearranging the ranking order in the post-processing stage, the recommender systems aim at generating a serendipity-oriented recommendation list. For example, after first obtaining a candidate set of items, Cheng et al. [55] generated a diversified recommendation list by selecting a subset from the candidate items that could maximize the diversity score, calculated as the sum of the dissimilarity between each candidate item pair:

$$diversity(y) = \frac{\sum_{i,j \in y, i \neq j} d(i,j)}{\frac{1}{2}n(n-1)}, \tag{11}$$

where $diversity(y)$ is the diversity score of an item set y. n denotes the total number of the items in the item set y. $d(i,j)$ denotes the dissimilarity between item i and j in the item set y, which is calculated by the inverse value of cosine similarity. Their proposed method outperformed all baseline models on the coverage of different genres in movie recommendations. Abdollahpouri et al. [90] proposed a personalized diversification re-ranking approach to increase the proportion of less popular items in recommendation lists. To increase the chance of discovering unpopular items, they first generated a candidate item list for recommendations based only on the relevance score. Then, by adding a bonus score to the candidate items whose categories were not in the initial top-k recommendation list, the system re-ranked the candidate item list and generated a new top-k recommendation list with a high probability to discover unpopular items. Two offline experiments showed that the proposed re-ranking approach was capable of reducing the popularity bias and balancing the tradeoff between accuracy and diversity more effectively. Huang and Wu [91] focused on providing a diversified recommendation list to improve the users' experiences on exploring POIs. The main goal of this study is to find POIs that were not similar to the users' historical records but were

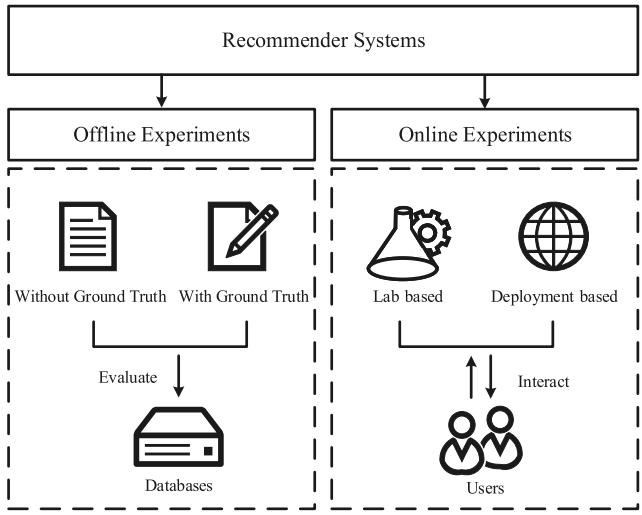


Fig. 4. Classification of serendipity recommendation model evaluations.

still correlated to their interests. The authors leveraged a Bi-LSTM model to learn the representation for user preferences and generated an initial candidate POI set that was relevant to the target user's historical behaviors. After training the model, to improve the diversity in the recommendation list, they further used the user's preference vector to find top-n similar users and collected another set of POIs that were viewed by these similar users but not by the target user. Finally, the recommender system mixed these two POI sets and generated a more diverse recommendation list.

To sum up, various efforts on incorporating serendipity components into deep learning structures have been conducted in recommender systems. The explorations on different structures of models provide diverse and inspiring views for further research. However, modeling serendipity is still a great challenge due to the elusive nature of serendipity. Its elements of accident and uncertainty are susceptible to systematic control and prediction. Once you model it, you tend to lose it. Additionally, as in most application domains, deep learning models lack interpretability. Although having promising results, deep learning models may not truly understand serendipity. With the recent development for transparent and explainable AI, serendipity recommendation research will benefit from casting light into the "black box" of the neural networks and obtaining some formats of explanations.

4 SERENDIPITY RECOMMENDATION EVALUATION

In this section, we discuss two commonly used methods in evaluating serendipity recommendation models, similar to evaluating most recommendation models: offline experiments and online experiments, as shown in our evaluation classification scheme in Figure 4. In the offline experiments, the traditional machine learning testing methods are adopted. The performance of the recommendation model will be evaluated using part of users' historical data reserved in the testing dataset. There is no need to recruit real users. In the online experiments, the researchers will recruit users either in a lab or in a deployed platform to try out the recommender and provide feedback. We will discuss each of them in the following sections.

4.1 Offline Experiments

For the offline experiments, the models are evaluated against the existing historical data. Similar to the traditional relevance-oriented models, the evaluation of serendipity-oriented models needs

both ground truth data and evaluation metrics. However, compared to the relevance-oriented ground truth data, fewer serendipity-oriented ground truth datasets are available. Therefore, we will organize the offline experiments into two scenarios: experiments without and with ground truth data on serendipity.

4.1.1 Offline Experiments Without Ground Truth Data on Serendipity. There are widely used public datasets for the evaluation of relevance-oriented recommender systems, such as MovieLens [92], the Amazon Review datasets [37, 74], the Yelp Challenge dataset [93], the Book-Crossing dataset [94], and the Netflix Competition dataset [95]. A summary of the datasets is presented in Table 2. All of these publicly available datasets contain users' historical ratings on the items, which could serve as the ground truth for relevance but not for serendipity. Some researchers attempt to engineer a pseudo ground truth dataset for serendipity from the original relevance-oriented datasets so that they can directly train the serendipity models using the engineered ground truth data and leverage the accuracy-based metrics to evaluate the experimental results. For example, Zhang et al. [28] tested the performance of their novelty-seeking system using two publicly available datasets from two websites: SinaWeibo and DianPing. However, these two datasets had no ground truth label for novel restaurants. They defined the novel restaurants as the restaurants that users had not visited before. Through the definition, they obtained the ground truth on novelty. Then they directly leveraged accuracy and NDCG as metrics to evaluate the model's performance on novelty, and compared their model with various baseline models (e.g., logistic regression, PPTM [96], and NSTM [97]). The results showed that their proposed model outperformed all of the baseline models on novel restaurant identification for both of the two datasets. Zhang et al. [27] trained and evaluated their proposed **Serendipity-Oriented Next Point-of-Interest Recommendation (SNPR)** model based on two datasets. However, these two datasets had no ground truth data on serendipity. To train their model, the authors defined and then calculated a serendipity score of each item as the ground truth. Specifically, the serendipity score was calculated by adding together two components: a relevance score and an unexpectedness score. The relevance score was available in the two original relevance-oriented datasets. The unexpectedness score was defined as the differences between a user and a POI. The authors then trained and tested a transformer-based deep learning model using the engineered ground truth data. The experiments were conducted by comparing their proposed model SNPR with three different groups of baseline models. The results showed that their proposed SNPR model achieved the best performance in identifying serendipity on all datasets.

Meanwhile, some researchers might find it difficult to define the ground truth data on serendipity due to its complex and subtle nature. Most researchers therefore trained the serendipity-oriented models on the data with relevance-oriented ground truth and evaluated the performance of these models through some self-defined evaluation metrics for serendipity, such as the proportion of unpopular or new items, the number of different items' categories, the overlaps between the recommendations and the users' historical records, and so forth. These serendipity evaluation metrics are all defined in a post hoc manner after the model training. For example, Li et al. [58] trained and evaluated their proposed movie recommender system based on two large-scale relevance-oriented datasets—MovieLens-1M and MovieLens-latest-small—also listed in Table 2. They evaluated the performance of their model using one traditional accuracy-based metric (F1 score) and two self-defined metrics for serendipity, representing the novelty aspect and the diversity aspect respectively. They measured the novelty of a recommendation list as the inverse value of a user's average rating on the movies in the recommendation list. They measured diversity using the sum of the Jaccard similarity of any movie pair in the recommendation list. While training the proposed recommendation model on three relevance-oriented datasets—Yelp, MovieLens, and Youku—Li et al. [24] evaluated the model using both the traditional accuracy-based metrics and the self-defined

Table 2. Publicly Available Datasets for Developing and Evaluating Recommendation Models

Dataset	No. of Users	No. of Items	No. of Ratings	Rating Scale
MovieLens-100K	943	1,683	100,000	1–5
MovieLens-1M	6,040	3,706	1,000,209	1–5
MovieLens-10M	69,878	10,681	10,000,054	1–5
MovieLens-20M	138,493	27,278	20,000,263	1–5
MovieLens-25M	162,541	62,423	25,000,095	1–5
MovieLens-latest-small	610	9,742	100,836	1–5
Amazon-Movies	44,439	25,047	1,070,860	1–5
Amazon-Books	367,982	603,668	8,898,041	1–5
Amazon-Kindle Store	3,061	6,073	132,594	1–5
Yelp Challenge	76,564	75,231	2,254,589	1–5
Book-Crossing	92,107	271,379	1,031,175	1–10
Netflix	480,189	17,770	100,480,507	1–5

serendipity-based metrics. For the accuracy-based metrics, they used AUC and hit ratio (HR@K). For the serendipity-based metrics, unexpectedness was proposed to measure the distance between the recommended item and each of a user's interest clusters. Another serendipity-based metric used in this study was coverage, which calculates the percentage of distinct items in all recommendation rounds against the total number of distinct items in the dataset. The results show their proposed model outperformed all baseline models on both accuracy-based and serendipity-based metrics in all three datasets. Wang et al. [98] trained and evaluated their model for recommending long-tail items on the review data from the RateBeer website. The dataset had no ground truth data on serendipity. The authors used users' ratings to first train their proposed model and then evaluated the performance of the model from three aspects: accuracy, recall, and coverage. Coverage was calculated as the ratio of the total number of different recommended items in all recommendation rounds among all items, which was the key metric to evaluate the model's performance on recommending the long tail items. By making comparisons with three commonly used baseline models, their proposed model outperformed all baseline models on the three metrics, suggesting the better performance on mining unpopular long-tail items while maintaining a good accuracy.

In summary, these preceding evaluation efforts are designed to leverage existing data and avoid the direct ground truth collection on serendipity. However, without a universal agreement on the definition of serendipity among researchers, the engineered ground truth as well as those self-defined evaluation metrics tend to be debatable, making both the models and the results not sufficiently systematic or generalizable.

4.1.2 Offline Experiments with Ground Truth Data on Serendipity. The studies with ground truth data on serendipity have advantages for both training and testing. For the testing purpose, accuracy-based metrics, such as precision, recall, MRR, and NDCG, can be directly used to measure the accuracy of recommending serendipity. Currently, as far as we know, there is only one publicly available dataset with human-labeled serendipity, named *Serendipity-2018*. The Serendipity-2018 dataset was collected by Kotkov et al. [15] from the GroupLens research group, who conducted a survey to collect users' opinions about whether a movie they watched was a serendipity. Specifically, they took advantage of an existing dataset they collected a few years back, called *MovieLens-1M*, and re-invited those users from that dataset to give additional ratings on the level of serendipity of the movies they watched in a retrospective way. As the result, they collected 475 users' serendipity responses regarding 2,146 movies. With the user annotations on serendipity of the movies, although not large scale, this Serendipity-2018 dataset has been used by several researchers

for model training and testing of their serendipity recommenders. For example, Pandey et al. [99] used the Serendipity-2018 dataset to train and evaluate their model. However, considering that the Serendipity-2018 dataset is not large scale enough to train the model, they first pre-trained the model on a large relevance-oriented dataset, then fine-tuned the model using 75% of the data from Serendipity-2018. The remaining 25% of the data served as the testing set to evaluate the performance of the model. By comparing with the state-of-the-art baseline models, their method achieved the best performance on NDCG. Ziarani and Ravanmehr [100] also trained and evaluated their proposed CNN-based serendipity recommendation model on Serendipity-2018. Instead of pre-training the model on a large relevance-oriented dataset, they first increased the size of the Serendipity-2018 dataset by randomly adding five non-serendipitous records for each user from MovieLens-1M, the original source data that were used to construct Serendipity-2018. Then, the authors separated the enhanced dataset into training and testing sets to train and evaluate their proposed model. They used one self-defined metric (SRDP) and two widely used accuracy-based metrics (hit ratio and NDCG). SRDP was defined as the proportion of serendipitous items, which were unexpected but rated positively. Unexpectedness labels were also available in Serendipity-2018. The authors conducted a series of offline experiments and compared the performance of their proposed model with four state-of-the-art methods in serendipitous recommendation systems at that time. The final results illustrated that their proposed model outperformed all baseline models in terms of SRDP and hit ratio. Additionally, their proposed model sacrificed less on NDCG. They concluded that their proposed model improved the performance of recommending serendipitous items and established a balance between accuracy and serendipity at the same time.

4.2 Online Experiments

Different from offline experiments, online experiments directly interact with users and collect the users' feedback. Based on the experiment setups, online experiments could have two types: lab-based experiments with a prototype recommender and deployment-based experiments with a deployed recommender.

For the lab-based experiments, the researchers implement the prototype of their proposed recommendation model and invite participants to their lab. For example, Afridi [22] evaluated the re-ranking algorithm for serendipity recommendations by conducting a lab-based evaluation at a university. He invited 60 students majoring in accounting and finance from a business school to participate in the study and conducted the online experiment in two steps. First, a recommendation list of study materials was generated by an accuracy-based model. It was presented to all of the participants, who were asked to fill out a questionnaire on their feelings of novelty, surprise, and satisfaction. Second, a new recommendation list re-ranked by a serendipity-oriented model was displayed, and the participants were asked again to fill out the same questionnaire of their feelings. By comparing the users' feedback between the accuracy-based model and the serendipity-oriented methods, the authors found that the proposed re-ranking methods can significantly improve users' satisfaction on the recommendations in the lab experiment. In another study, Jenders et al. [101] evaluated their proposed serendipity-oriented recommender system by conducting a lab-based experiment, which invited 27 people with a background in natural sciences. The participants were presented with the five highest ranked articles generated by one of the five ranking models. Then, the participants were asked to rate on a Likert scale from 1 to 5 on the relevance level and the serendipity level of the recommended articles. The experimental results showed that their proposed model received relatively balanced serendipity ratings and relevance ratings compared with other baseline models.

In the lab-based experiments, the number of participants is typically small-scale and may not be representative enough of potential users. Therefore, in addition to lab-based experiments, many

researchers, especially from industries with the help of the corporate infrastructure, tend to conduct the deployment-based online experiments, where the proposed recommender systems are live with real-time users. Compared with lab-based experiments, the deployment-based approach is able to reach a large number of users in a short period of time and therefore collect large-scale feedback. However, the requirement of infrastructure support could be a barrier for many researchers who do not have such resources immediately available. Currently, most of the deployment-based experiments are conducted by big-name companies with infrastructure support. For example, Huang et al. [23] conducted a 4-day online experiment in Baidu, a large search engine in China, and selected CTR (click-through rate) as the metric to evaluate the performance of their proposed CNN-based serendipity recommendation model. Compared with the other five baseline models, their proposed method achieved the highest CTR score, indicating the best user engagement levels. Chen et al. [42] and Wang et al. [102] both conducted a large-scale online experiment on a well-known mobile e-commerce platform in China called *Mobile Taobao* to obtain the users' perceptions toward serendipity in recommendations. The users were asked to answer questionnaires about personal background and serendipity assessment immediately after receiving recommendations from the system. In addition to serendipity, they evaluated the curiosity level of users after seeing those recommendations. They used the well-established Curiosity and Exploration Inventory-II [103], which has 10 questions to evaluate users' curiosity levels.

By deploying the recommendation model to a commercial news website for 1 month, Zheng et al. [85] conducted an online experiment to evaluate the performance of the proposed news recommendation model. They selected five typical machine learning or state-of-the-art deep learning models as the baseline models. User feedback was recorded for each model respectively. As the result, their proposed model outperformed all five baseline models on both accuracy-based metrics and the diversity-based metrics. Li et al. [24] also conducted online experiments by designing an online A/B test in addition to their offline experiments mentioned in Section 4.1.1. They deployed their model on Alibaba-Youku, which is a major video recommendation platform in China, and collected users' real-time feedback. During the A/B testing period, the authors compared their proposed model with the current model on the Alibaba-Youku platform. The two recommendation models were used randomly by the user traffic. The users' feedback (received by clicking or not clicking the recommendations) on the two models was recorded respectively. To evaluate the recommendation relevance, they used four standard accuracy-based metrics. To evaluate the unexpectedness and diversity, they used the same metrics of unexpectedness and coverage as their offline experiments, but the data were from user feedback this time. The model gained a significant improvement on both of the four accuracy-based metrics and two serendipity-based metrics.

To sum up, compared to offline evaluations, online experiments are able to evaluate the recommendation model performance more directly by collecting real-time user feedback. However, collecting sufficient users' feedback data is both time- and labor consuming, which is not able to scale up. Additionally, the deployment-based user studies need the support of platform infrastructures and existing user traffic, which tend to be inaccessible for researchers without an industry background.

5 FUTURE DIRECTIONS AND OPEN ISSUES

After reviewing the existing research on deep learning methods for recommending serendipity, we suggest the three future directions.

5.1 Ground Truth Data Generation for Serendipity

One of the prominent challenges in serendipity recommendation research is the lack of ground truth data. Human annotations of serendipity experiences are rare and difficult to collect. In

contrast, there are abundant sources of user reviews these days. User reviews are a rich source for understanding users' opinions and experiences, and therefore they are widely collected by various industries and governments. We expect user reviews to provide valuable self-reports of naturally occurring serendipity. Currently, this type of data is highly underused. If the reviews containing serendipity experiences could be automatically identified by machine learning or deep learning models, a huge amount of review data could be utilized as the "labeled" data for serendipity recommendation research. Therefore, research on effective text mining and natural language processing models to detect the users' serendipity experiences from the reviews is meaningful, although few researchers have been working on such areas. Recently, tremendous success has been observed in deep learning methods in various natural language processing tasks, including sentiment analysis and machine translation, among others. We believe that deep learning models are able to identify a serendipity experience not from a piece of user reviews. What makes user reviews more valuable is that those reviews were usually collected together with user profiles, user ratings, and other user behaviors. Connecting these data is useful for recommender systems research to link users, items, and their interactions.

5.2 Timing for Serendipity Recommendations

Most of the existing studies focus on identifying what is serendipitous to a user, but few of them work on when to recommend serendipity. Users may not always need serendipity, and their demands on serendipity may change over time. Finding appropriate timing to provide serendipitous recommendations could also help improve the users' satisfaction. Both user factors and context factors need to be considered to identify good timing. The user factors include users' demographics and behaviors. The context factors could be the time and the location. For example, a user who is on vacation at Miami Beach during Christmas would have a different need for serendipity than if the same user is having a regular day at the office. Both factors are ever-changing, and their impacts of serendipity tend to be uncertain. One possible solution is the deep reinforcement learning approach, which dynamically captures a user's state and updates the model using the user's real-time feedback. However, as mentioned in Section 3, most existing deep reinforcement learning models users' behavior patterns as a first-order MDP, constructing a state only based on the impact of the immediately previous state. While reducing the computational complexity, this first-order process loses much valuable information that could have been obtained from a longer sequence of states. Future research could go along the direction of the deep reinforcement learning approach but without the first-order sequence assumption to find timing for serendipity recommendations.

5.3 Cross-Domain Learning for Serendipity Recommendations

Cross-domain recommender systems have been proposed as a mitigation to the problem of data sparsity. Different from the traditional recommender systems that only focus on recommendations in one domain (e.g., only movies or only books), cross-domain recommendations utilize the information from multiple domains to enrich the information of the target domain. By providing extra knowledge, cross-domain recommendations bridge the different domains and help the recommendation models improve the performance in the target domain. Researchers believe that the users' behavioral data from different domains may potentially help to understand the user better and identify serendipitous items for this user [104]. Furthermore, cross-domain recommender systems can assist with tracking the shift of user interests by transferring knowledge from the source domain(s) within a time frame [105]. As mentioned before, lacking enough ground truth data on serendipity is a big challenge for applying deep learning methods to serendipity-oriented recommendations. If we collect ground truth data on serendipity in a few domains, it will be helpful to transfer these domains' knowledge to other domains for serendipity discovery. Additionally,

transferring knowledge from multi-domains will provide more diverse viewpoints on users or items and is therefore beneficial for serendipity-oriented recommendations. Currently, a lot of cross-domain approaches are leveraged by the researchers to solve the cross-domain recommendation problem, such as transfer learning, zero-shot learning [106, 107], and the pre-train and fine-tune mechanism [99]. Among them, transfer learning is the most commonly used approach in cross-domain recommendations with the ability of learning projection functions between domains to transfer knowledge between different but related domains [108]. However, as far as we know, there are few efforts in applying transfer learning in serendipity cross-domain recommendations. There are many open research questions, such as non-overlapping user or item knowledge transfer (this is especially the case for different serendipity domains), heterogeneous data, and robustness and privacy issues. More explorations on cross-domain learning for serendipity recommendations are worth pursuing in the future.

6 CONCLUSION

This survey summarized and classified the existing research on the deep learning approaches for serendipity-oriented recommendation models. It provided a thorough review of serendipity definitions, deep learning models, available datasets, and evaluation approaches. For the definition of serendipity, four components were discussed: unexpectedness, novelty, diversity, and relevance. As for the deep learning models, three different stages were categorized according to the incorporation of serendipity components: the pre-processing stage, in-processing stage, and post-processing stage. For the serendipity evaluation methods, offline experiments with or without serendipity ground truth data, as well as online experiments involving user interactions with prototypes in the lab or deployed systems on platforms were discussed.

Although challenges persist in applying deep learning models to serendipity-oriented recommendations, the remarkable success of deep learning in recommender systems motivates researchers to further explore new trends and potential directions. We believe that with the advancement of technology and the improved understanding of deep learning, researchers will be able to deliver serendipitous information more effectively in digital environments, empowering individuals with an increased chance of encountering beneficial discoveries.

ACKNOWLEDGMENTS

We are grateful to NSF for making this research possible.

REFERENCES

[1] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys* 52, 1 (2019), 1–38.

[2] Zhe Fu, Li Yu, and Xi Niu. 2022. TRACE: Travel reinforcement recommendation based on location-aware context extraction. *ACM Transactions on Knowledge Discovery from Data* 16, 4 (2022), 1–22.

[3] Youfang Leng, Li Yu, and Xi Niu. 2022. Dynamically aggregating individuals' social influence and interest evolution for group recommendations. *Information Sciences* 614 (2022), 223–239.

[4] Xi Niu and Xiangyu Fan. 2019. Deep learning of human information foraging behavior with a search engine. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. 185–192.

[5] Mukund Deshpande and George Karypis. 2004. Item-based top-*n* recommendation algorithms. *ACM Transactions on Information Systems* 22, 1 (2004), 143–177.

[6] Xi Niu and Ahmad Al-Doulat. 2021. LuckyFind: Leveraging surprise to improve user satisfaction and inspire curiosity in a recommender system. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*. 163–172.

[7] Xi Niu. 2018. An adaptive recommender system for computational serendipity. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*. 215–218.

[8] Eli Pariser. 2011. *The Filter Bubble: How the New Personalized Web Is Changing What We Read and How We Think*. Penguin.

[9] Engin Bozdag. 2013. Bias in algorithmic filtering and personalization. *Ethics and Information Technology* 15, 3 (2013), 209–227.
[10] Chris Anderson. 2006. *The Long Tail: Why the Future of Business Is Selling Less of More.* Hachette Books.
[11] Yoon-Joo Park and Alexander Tuzhilin. 2008. The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 ACM Conference on Recommender Systems.* 11–18.
[12] A. G. Borselli. 1965. Serendipity and the three princes: From the Peregrinaggio of 1557, Theodore G. Remer (Ed.). University of Oklahoma Press, Norman, OK. (The compilation of the work here is regarded as probably by Michele Tramezzino, and translated by Augusto G. Borselli and Theresa L. Borselli.)
[13] Lori McCay-Peet and Elaine G. Toms. 2017. Researching serendipity in digital information environments. *Synthesis Lectures on Information Concepts, Retrieval, and Services* 9, 6 (2017), i–91.
[14] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems* 22, 1 (2004), 5–53.
[15] Denis Kotkov, Joseph A. Konstan, Qian Zhao, and Jari Veijalainen. 2018. Investigating serendipity in recommender systems based on real user feedback. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing.* 1341–1350.
[16] Qingchen Zhang, Laurence T. Yang, Zhikui Chen, and Peng Li. 2018. A survey on deep learning for big data. *Information Fusion* 42 (2018), 146–157.
[17] Marius Kaminskas and Derek Bridge. 2016. Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems* 7, 1 (2016), 1–42.
[18] Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen. 2016. A survey of serendipity in recommender systems. *Knowledge-Based Systems* 111 (2016), 180–192.
[19] Reza Jafari Ziarani and Reza Ravanmehr. 2021. Serendipity in recommender systems: A systematic literature review. *Journal of Computer Science and Technology* 36, 2 (2021), 375–396.
[20] Fakhri Abbas and Xi Niu. 2019. Computational serendipitous recommender system frameworks: A literature survey. In *Proceedings of the 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA'19).* IEEE, Los Alamitos, CA, 1–8.
[21] Zhe Fu, Xi Niu, and Li Yu. 2023. Wisdom of crowds and fine-grained learning for serendipity recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval.*
[22] Ahmad Hassan Afridi. 2018. User control and serendipitous recommendations in learning environments. *Procedia Computer Science* 130 (2018), 214–221.
[23] Jizhou Huang, Shiqiang Ding, Haifeng Wang, and Ting Liu. 2018. Learning to recommend related entities with serendipity for web search users. *ACM Transactions on Asian and Low-Resource Language Information Processing* 17, 3 (2018), 1–22.
[24] Pan Li, Maofei Que, Zhichao Jiang, Yao Hu, and Alexander Tuzhilin. 2020. PURS: Personalized unexpected recommender system for improving user satisfaction. In *Proceedings of the 14th ACM Conference on Recommender Systems.* 279–288.
[25] Satoko Inoue and Masataka Tokumaru. 2020. Serendipity recommender system for academic disciplines. In *Proceedings of the 2020 Joint 11th International Conference on Soft Computing and Intelligent Systems and the 21st International Symposium on Advanced Intelligent Systems (SCIS-ISIS'20).* IEEE, Los Alamitos, CA, 1–4.
[26] Ningxia Wang and Li Chen. 2021. User bias in beyond-accuracy measurement of recommendation algorithms. In *Proceedings of the 15th ACM Conference on Recommender Systems.* 133–142.
[27] Mingwei Zhang, Yang Yang, Rizwan Abbas, Ke Deng, Jianxin Li, and Bin Zhang. 2021. SNPR: A serendipity-oriented next POI recommendation model. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management.* 2568–2577.
[28] Fuzheng Zhang, Kai Zheng, Nicholas Jing Yuan, Xing Xie, Enhong Chen, and Xiaofang Zhou. 2015. A novelty-seeking based dining recommender system. In *Proceedings of the 24th International Conference on World Wide Web.* 1362–1372.
[29] Wei Lu, Fu-Lai Chung, Wenhao Jiang, Martin Ester, and Wei Liu. 2018. A deep Bayesian tensor-based system for video recommendation. *ACM Transactions on Information Systems* 37, 1 (2018), 1–22.
[30] Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. 2018. Attention-based transactional context embedding for next-item recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.
[31] Ludovik Coba, Panagiotis Symeonidis, and Markus Zanker. 2018. Novelty-aware matrix factorization based on items' popularity. In *Proceedings of the International Conference of the Italian Association for Artificial Intelligence.* 516–527.
[32] Yuanbo Xu, Yongjian Yang, En Wang, Jiayu Han, Fuzhen Zhuang, Zhiwen Yu, and Hui Xiong. 2020. Neural serendipity recommendation: Exploring the balance between accuracy and novelty with sparse explicit feedback. *ACM Transactions on Knowledge Discovery from Data* 14, 4 (2020), 1–25.

[33] Pranita Shrestha, Min Zhang, Yiqun Liu, and Shaoping Ma. 2020. Curiosity-inspired personalized recommendation. In *Proceedings of the 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'20)*. IEEE, Los Alamitos, CA, 33–40.

[34] Kachun Lo and Tsukasa Ishigaki. 2021. PPNW: Personalized pairwise novelty loss weighting for novel recommendation. *Knowledge and Information Systems* 63, 5 (2021), 1117–1148.

[35] Jungkyu Han and Hayato Yamana. 2019. Geographic diversification of recommended POIs in frequently visited areas. *ACM Transactions on Information Systems* 38, 1 (2019), 1–39.

[36] Xiaoyu Ge, Panos K. Chrysanthis, Konstantinos Pelechrinis, Demetrios Zeinalipour-Yazti, and Mohamed A. Sharaf. 2020. Serendipity-based points-of-interest navigation. *ACM Transactions on Internet Technology* 20, 4 (2020), 1–32.

[37] Jianing Sun, Wei Guo, Dengcheng Zhang, Yingxue Zhang, Florence Regol, Yaochen Hu, Huifeng Guo, et al. 2020. A framework for recommending accurate and diverse items using Bayesian graph convolutional neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2030–2039.

[38] Zhihua Cui, Xianghua Xu, Fei Xue, Xingjuan Cai, Yang Cao, Wensheng Zhang, and Jinjun Chen. 2020. Personalized recommendation system based on collaborative filtering for IoT scenarios. *IEEE Transactions on Services Computing* 13, 4 (2020), 685–695.

[39] Wanyu Chen, Pengjie Ren, Fei Cai, Fei Sun, and Maarten De Rijke. 2021. Multi-interest diversification for end-to-end sequential recommendation. *ACM Transactions on Information Systems* 40, 1 (2021), 1–30.

[40] Jongsoo Lee, Jung Ah Shin, and Dong-Kyu Chae. 2022. A diversity personalization approach towards recommending POIs for Jeju island. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*. 1804–1807.

[41] Marco De Gemmis, Pasquale Lops, Giovanni Semeraro, and Cataldo Musto. 2015. An investigation on the serendipity problem in recommender systems. *Information Processing & Management* 51, 5 (2015), 695–717.

[42] Li Chen, Yonghua Yang, Ningxia Wang, Keping Yang, and Quan Yuan. 2019. How serendipity improves user satisfaction with recommendations? A large-scale user evaluation. In *Proceedings of the World Wide Web Conference*. 240–250.

[43] Xi Niu and Fakhri Abbas. 2019. Computational surprise, perceptual surprise, and personal background in text understanding. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*. 343–347.

[44] Xi Niu, Wlodek Zadrozny, Kazjon Grace, and Weimao Ke. 2018. Computational surprise in information retrieval. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1427–1429.

[45] Paul Hugh Cleverley and Simon Burnett. 2015. Creating sparks: Comparing search results using discriminatory search term word co-occurrence to facilitate serendipity in the enterprise. *Journal of Information & Knowledge Management* 14, 01 (2015), 1550007.

[46] Xi Niu and Fakhri Abbas. 2017. A framework for computational serendipity. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation, and Personalization*. 360–363.

[47] Xi Niu, Fakhri Abbas, Mary Lou Maher, and Kazjon Grace. 2018. Surprise me if you can: Serendipity in health information. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[48] Xiangyu Fan and Xi Niu. 2018. Implementing and evaluating serendipity in delivering personalized health information. *ACM Transactions on Management Information Systems* 9, 2 (2018), 1–19.

[49] Benard Maake, Tranos Zuva, and Sunday O. Ojo. 2019. A serendipitous research paper recommender system. *International Journal of Business and Management Studies* 11, 1 (2019), 39–53.

[50] P. Moreira Gabriel De Souza, Dietmar Jannach, and Adilson Marques Da Cunha. 2019. Contextual hybrid session-based news recommendation with recurrent neural networks. *IEEE Access* 7 (2019), 169185–169203.

[51] Jacek Wasilewski and Neil Hurley. 2019. Bayesian personalized ranking for novelty enhancement. In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation, and Personalization*. 144–148.

[52] Marwa Hussien Mohamed, Mohamed Helmy Khafagy, Heba Elbeh, and Ahmed Mohamed Abdalla. 2019. Sparsity and cold start recommendation system challenges solved by hybrid feedback. *International Journal of Engineering Research and Technology* 12, 12 (2019), 2735–2742.

[53] Yashar Deldjoo, Maurizio Ferrari Dacrema, Mihai Gabriel Constantin, Hamid Eghbal-Zadeh, Stefano Cereda, Markus Schedl, Bogdan Ionescu, and Paolo Cremonesi. 2019. Movie genome: Alleviating new item cold start in movie recommendation. *User Modeling and User-Adapted Interaction* 29, 2 (2019), 291–343.

[54] Pramit Mazumdar, Bidyut Kr. Patra, and Korra Sathya Babu. 2020. Cold-start point-of-interest recommendation through crowdsourcing. *ACM Transactions on the Web* 14, 4 (2020), 1–36.

[55] Peizhe Cheng, Shuaiqiang Wang, Jun Ma, Jiankai Sun, and Hui Xiong. 2017. Learning to recommend accurate and diverse items. In *Proceedings of the 26th International Conference on World Wide Web*. 183–192.

[56] Jorge Díez, David Martínez-Rego, Amparo Alonso-Betanzos, Oscar Luaces, and Antonio Bahamonde. 2019. Optimizing novelty and diversity in recommendations. *Progress in Artificial Intelligence* 8, 1 (2019), 101–109.

[57] Pablo Castells, Neil J. Hurley, and Saul Vargas. 2015. Novelty and diversity in recommender systems. In *Recommender Systems Handbook*. Springer, 881–918.
[58] Xueqi Li, Wenjun Jiang, Weiguang Chen, Jie Wu, and Guojun Wang. 2019. HAES: A new hybrid approach for movie recommendation with elastic serendipity. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1503–1512.
[59] Tuck Wah Leong, Peter Wright, Frank Vetere, and Steve Howard. 2010. Understanding experience using dialogical methods: The case of serendipity. In *Proceedings of the 22nd Conference of the Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction*. 256–263.
[60] Ataur Rahman and Max L. Wilson. 2015. Exploring opportunities to facilitate serendipity in search. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 939–942.
[61] Tao Wang, Xiaoyu Shi, and Mingsheng Shang. 2020. Diversity-aware top-*n* recommendation: A deep reinforcement learning way. In *Proceedings of the CCF Conference on Big Data*. 226–241.
[62] Shoshana Loeb, Ralph Hill, and Tom Brinck. 1992. Lessons from LyricTime: A prototype multimedia system. *ACM SIGCOMM Computer Communication Review* 22, 3 (1992), 35–36.
[63] José Campos and Antonio Dias de Figueiredo. 2002. *Programming for Serendipity*. AAAI Technical Report FS-02-01. AAAI Press.
[64] Leo Iaquinta, Marco De Gemmis, Pasquale Lops, Giovanni Semeraro, Michele Filannino, and Piero Molino. 2008. Introducing serendipity in a content-based recommender system. In *Proceedings of the 2008 8th International Conference on Hybrid Intelligent Systems*. IEEE, Los Alamitos, CA, 168–173.
[65] Kensuke Onuma, Hanghang Tong, and Christos Faloutsos. 2009. TANGENT: A novel, 'Surprise me', recommendation algorithm. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 657–666.
[66] Yuan Cao Zhang, Diarmuid Ó. Séaghdha, Daniele Quercia, and Tamas Jambor. 2012. Auralist: Introducing serendipity into music recommendation. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*. 13–22.
[67] Panagiotis Adamopoulos and Alexander Tuzhilin. 2014. On unexpectedness in recommender systems: Or how to better expect the unexpected. *ACM Transactions on Intelligent Systems and Technology* 5, 4 (2014), 1–32.
[68] Yejin Kim, Kwangseob Kim, Chanyoung Park, and Hwanjo Yu. 2019. Sequential and diverse recommendation with long tail. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*, Vol. 19. 2740–2746.
[69] Mengmeng Li, Tian Gan, Meng Liu, Zhiyong Cheng, Jianhua Yin, and Liqiang Nie. 2019. Long-tail hashtag recommendation for micro-videos with graph convolutional network. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 509–518.
[70] Max Welling and Thomas N. Kipf. 2016. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations (ICLR'17)*.
[71] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1532–1543.
[72] Kazjon Grace, Mary Lou Maher, Nicholas Davis, and Omar Eltayeby. 2018. Surprise walks: Encouraging users towards novel concepts with sequential suggestions. In *Proceedings of the 9th International Conference on Computational Creativity (ICCC'18)*.
[73] Younghoon Lee. 2020. Serendipity adjustable application recommendation via joint disentangled recurrent variational auto-encoder. *Electronic Commerce Research and Applications* 44 (2020), 101017.
[74] Xueqi Li, Wenjun Jiang, Weiguang Chen, Jie Wu, Guojun Wang, and Kenli Li. 2020. Directional and explainable serendipity recommendation. In *Proceedings of The Web Conference 2020*. 122–132.
[75] John Boaz Lee, Ryan A. Rossi, Sungchul Kim, Nesreen K. Ahmed, and Eunyee Koh. 2019. Attention models in graphs: A survey. *ACM Transactions on Knowledge Discovery from Data* 13, 6 (2019), 1–25.
[76] Shaina Raza and Chen Ding. 2021. Deep neural network to tradeoff between accuracy and diversity in a news recommender system. In *Proceedings of the 2021 IEEE International Conference on Big Data (Big Data'21)*. IEEE, Los Alamitos, CA, 5246–5256.
[77] Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1063–1072.
[78] Ruobing Xie, Qi Liu, Shukai Liu, Ziwei Zhang, Peng Cui, Bo Zhang, and Leyu Lin. 2021. Improving accuracy and diversity in matching of recommendation with diversified preference network. *IEEE Transactions on Big Data* 8, 4 (2021), 955–967.

[79] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS'17)*.

[80] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. 173–182.

[81] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative memory network for recommendation systems. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. 515–524.

[82] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 165–174.

[83] Ruomu Zou. 2019. A deep, forgetful novelty-seeking movie recommender model. *arXiv Preprint arXiv:1909.01811* (2019).

[84] Sindhu Padakandla. 2021. A survey of reinforcement learning algorithms for dynamically varying environments. *ACM Computing Surveys* 54, 6 (2021), 1–25.

[85] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*. 167–176.

[86] Rupert G. Miller Jr. 2011. *Survival Analysis*. John Wiley & Sons.

[87] Joseph G. Ibrahim, Ming-Hui Chen, Danjie Zhang, and Debajyoti Sinha. 2013. Bayesian analysis of the Cox model. In *Handbook of Survival Analysis*, John P. Klein, Hans C. van Houwelingen, Joseph G. Ibrahim, and Thomas H. Scheike (Eds.). Chapman & Hall/CRC Handbooks of Modern Statistical Methods. Chapman & Hall/CRC, Boca Raton, FL, 27–48.

[88] Keigo Sakurai, Ren Togo, Takahiro Ogawa, and Miki Haseyama. 2020. Music playlist generation based on reinforcement learning using acoustic feature map. In *Proceedings of the 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE'20)*. IEEE, Los Alamitos, CA, 942–943.

[89] Dusan Stamenkovic, Alexandros Karatzoglou, Ioannis Arapakis, Xin Xin, and Kleomenis Katevas. 2022. Choosing the best of both worlds: Diverse and novel recommendations through multi-objective reinforcement learning. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*. 957–965.

[90] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing popularity bias in recommender systems with personalized re-ranking. In *Proceedings of the 32nd International Florida Artificial Intelligence Research Society Conference (FLAIRS'19)*.

[91] Chung-Ming Huang and Chen-Yi Wu. 2021. The point of interest (POI) recommendation for mobile digital culture heritage (M-DCH) based on the behavior analysis using the recurrent neural networks (RNN) and user-collaborative filtering. *Journal of Internet Technology* 22, 4 (2021), 821–833.

[92] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems* 5, 4 (2015), 1–19.

[93] Nabiha Asghar. 2016. Yelp dataset challenge: Review rating prediction. *arXiv Preprint arXiv:1605.05362* (2016).

[94] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*. 22–32.

[95] James Bennett and Stan Lanning. 2007. The Netflix Prize. In *Proceedings of the KDD Cup and Workshop*, Vol. 2007. ACM, New York, NY, 1–4.

[96] Jinoh Oh, Sun Park, Hwanjo Yu, Min Song, and Seung-Taek Park. 2011. Novel recommendation based on personal popularity tendency. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*. IEEE, Los Alamitos, CA, 507–516.

[97] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, and Xing Xie. 2014. Mining novelty-seeking trait across heterogeneous domains. In *Proceedings of the 23rd International Conference on World Wide Web*. 373–384.

[98] Yong Wang, Jingyuan Wang, and Li Li. 2018. Enhancing long tail recommendation based on user's experience evolution. In *Proceedings of the 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design (CSCWD'18)*. IEEE, Los Alamitos, CA, 25–30.

[99] Gaurav Pandey, Denis Kotkov, and Alexander Semenov. 2018. Recommending serendipitous items using transfer learning. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1771–1774.

[100] Reza Jafari Ziarani and Reza Ravanmehr. 2021. Deep neural network approach for a serendipity-oriented recommendation system. *Expert Systems with Applications* 185 (2021), 115660.

[101] Maximilian Jenders, T. Lindhauer, Gjergji Kasneci, Ralf Krestel, and Felix Naumann. 2015. A serendipity model for news recommendation. In *KI 2015: Advances in Artificial Intelligence*. Lecture Notes in Computer Science, Vol. 9324. Springer, 111–123.

[102] Ningxia Wang, Li Chen, and Yonghua Yang. 2020. The impacts of item features and user characteristics on users' perceived serendipity of recommendations. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation, and Personalization*. 266–274.

[103] Todd B. Kashdan, Matthew W. Gallagher, Paul J. Silvia, Beate P. Winterstein, William E. Breen, Daniel Terhar, and Michael F. Steger. 2009. The Curiosity and Exploration Inventory-II: Development, factor structure, and psychometrics. *Journal of Research in Personality* 43, 6 (2009), 987–998.

[104] Fuzhen Zhuang, Yingmin Zhou, Fuzheng Zhang, Xiang Ao, Xing Xie, and Qing He. 2017. Sequential transfer learning: Cross-domain novelty seeking trait mining for recommendation. In *Proceedings of the 26th International Conference on World Wide Web Companion*. 881–882.

[105] Muhammad Murad Khan, Roliana Ibrahim, and Imran Ghani. 2017. Cross domain recommender systems: A systematic literature review. *ACM Computing Surveys* 50, 3 (2017), 1–34.

[106] Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, Yang Yang, and Zi Huang. 2019. From zero-shot learning to cold-start recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4189–4196.

[107] Zhaoxin Huan, Gongduo Zhang, Xiaolu Zhang, Jun Zhou, Qintong Wu, Lihong Gu, Jinjie Gu, Yong He, Yue Zhu, and Linjian Mo. 2022. An industrial framework for cold-start recommendation in zero-shot scenarios. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3403–3407.

[108] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proceedings of the IEEE* 109, 1 (2020), 43–76.

Received 11 September 2022; revised 22 March 2023; accepted 31 May 2023