Siamese neural networks in recommendation

Nicolás Serrano[1*] and Alejandro Bellogín[1*]

[1*]Computer Science Department, Universidad Autónoma de Madrid, Spain.

*Corresponding author(s). E-mail(s): nicolas.serranos@estudiante.uam.es; alejandro.bellogin@uam.es;

Abstract

Recommender systems are widely adopted as an increasing research and development area, since they provide users with diverse and useful information tailored to their needs. Several strategies have been proposed, and in most of them some concept of similarity is used as a core part of the approach, either between items or between users. At the same time, Siamese Neural Networks are being used to capture the similarity of items in the image domain, as they are defined as a subtype of Artificial Neural Networks built with (at least two) identical networks that share their weights. In this review, we study the proposals done in the intersection of these two fields, that is, how Siamese Networks are being used for recommendation. We propose a classification that considers different recommendation problems and algorithmic approaches. Some research directions are pointed out to encourage future research. To the best of our knowledge, this paper is the first comprehensive survey that focuses on the usage of Siamese Neural Networks for Recommender Systems.

Keywords: Recommender Systems, Siamese Networks, Architecture, Evaluation, Review

1 Introduction

Siamese Neural Networks (SNNs) emerged in 1994 as an artificial neural network architecture where two identical neural networks, then perceptrons, calculated the similarity between two elements [6]. This type of architecture is well-suited for situations where learning a similarity is key for an application; moreover, it is demonstrated to be quite scalable and efficient. Although they have been neglected for many years, thanks to advances in artificial neural network architectures, they are currently being highly used in the multimedia domain, where they take advantage of these improvements, obtaining a high precision in the calculation of the similarity between elements.

One of the research areas where SNNs have been applied is Recommender Systems (RSs). These systems play a very important role nowadays, where information overload and huge catalogs are prevalent. Ideally, recommendation algorithms should provide novel and interesting information to users, while showing diverse items that would improve the user experience with the system. To achieve this, there exist different strategies in terms of defining how a recommendation is generated, although they are usually categorized as content-based and collaborative filtering [41], but other types such as demographic or knowledge-based exist and are applied in the community. Most of these strategies have at their core the concept of *similarity*, so that *similar* items to those previously consumed by the user are assumed to match their (future) preferences.

Therefore, integrating SNNs into RSs is a natural step, as both deal and exploit similarities. In this review, we have revised the literature to understand the state-of-the-art in terms of how SNNs have been applied to the recommendation problem, as we have found no other work where they have been analyzed and categorized in detail. Given the flexibility of these approaches, and the prevalence of RSs, they have been used in several domains with different types of data. Hence, we propose a classification of the approaches and techniques found, while evidencing some gaps and challenges in this scenario. We also include experiments to further emphasize and visualize these issues.

Hence, the main contributions of this paper include:

- A detailed review of the state-of-the-art focused on methods that use Siamese Neural Networks for recommendation.
- A classification that covers the recommendation tasks addressed by these works, how the SNNs are configured, and their evaluation.
- A discussion that emphasizes the gaps and challenges in the area at the moment, both from a bibliographical and an experimental perspective.

The rest of the paper is structured as follows. Section 2 presents in detail the Siamese Networks and Recommender Systems. Section 3 reviews the literature and introduces the classification of the approaches we propose. Section 4 discusses the main issues found in the area by providing an overview of the reviewed techniques, both from a practical and a theoretical perspective. Finally, Section 5 concludes the paper.

2 Background

2.1 Siamese Networks

Similarity is a key concept not only on Recommender Systems, but in different fields of computer science. There exists different methods when measuring the similarity between elements, like the cosine similarity or the Pearson correlation coefficient. Nevertheless, these measurements are not useful when the elements to compare are lists of different features, having each of the features different meanings. In 1993, the Siamese Neural Networks (SNNs) were proposed as a solution of the problem, when solving the signature verification process, to measure the similarity between two signatures [6].

The Siamese Neural Network is an architecture of Artificial Neural Network [10] built from several and identical feedforward networks which shares weights, joined at the output. The elements to compare are processed at the same time, one for each network. Finally, the outputs are compared using a distance metric, such as Euclidean distance, determining whether they are similar (value close to 0) or different (value close to 1). In training, this result is compared to the labeling of the data to determine the efficacy of the model by means of a loss function.

Siamese Neural Networks can also be divided in two main models, based on the number of input parameters to the network: pairs and triplets.

2.1.1 SNNs based on Pairs

Siamese Neural Networks whose input is a pair of elements are the first one proposed in 1993 [6], sometimes also called *Twin Neural Networks*. The elements are paired and the system learns if they are similar or not through a loss function. A diagram of the architecture can be seen in Figure 1.

There are two loss functions mainly used in these models, the Binary Cross Entropy and the Contrastive loss. The Binary Cross Entropy determines if two elements are from the same or different class (Equation 1):

$$\text{Loss} = (Y)(-\log(Y_{pred})) + (1-Y)(-\log(1-Y_{pred})) \tag{1}$$

where

- Y is the label value. It will be 1 if both pairs belong to the same class, and 0 otherwise.
- Y_{pred} is the label value predicted by the Siamese network.

On the other hand, the Contrastive loss is a loss function that is better suited (in principle) to the problem addressed by Siamese Networks [12], as the objective of the network is to differentiate (and not classify) between two elements (Equation 2):

$$\text{Loss} = Y * D^2 + (1-Y) * \max(\alpha - D, 0)^2 \tag{2}$$

where:

- Y is the label value, as in Equation 1.
- D is the Euclidean distance between the outputs of both sister networks, components of the siamese network.
- α is the margin, a minimum distance that aims to discriminate between samples that are near and far away (in terms of D). By default, it is set to 1.

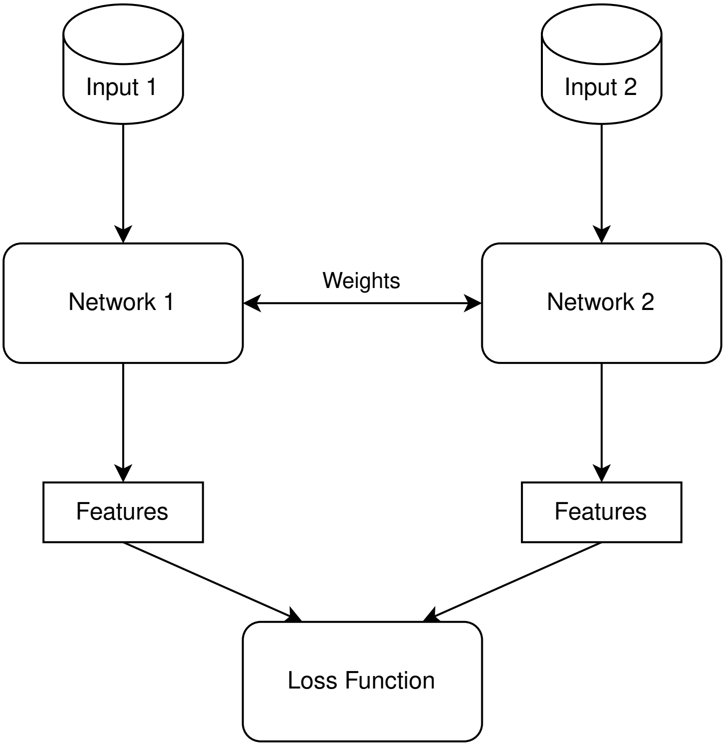

Fig. 1 Architecture of a Siamese Neural Network where the input is a pair of elements.

2.1.2 SNNs based on Triplets

Siamese Neural Networks whose input is a triplet of elements were proposed in 2015 [18], and are sometimes called *Triplet Networks*. On them, instead of having a pair of elements labeled from the same or different classes, there are three elements as input of the network. The first element is the anchor, it is the element to be compared against the other two. The other two elements are an element from the same class, called positive, and an element from a different class, called negative. In Figure 2, as in the Siamese Networks whose input is a pair of elements, the feedforward networks are identical, the weights between networks are shared, the distance between the outputs is calculated, and a loss function is used during training.

The loss function used in the Triplet Networks, by definition, needs to be different to the one used in Twin Networks. There, as explained before, the model uses the Binary Cross Entropy or the Contrastive losses, however, in Triplet Networks the so-called Triplet Loss was proposed to train the model:

$$\mathrm{L(A,P,N)} = \max(\|e(A) - e(P)\|^2 - \|e(A) - e(N)\|^2 + \alpha, 0) \tag{3}$$

where

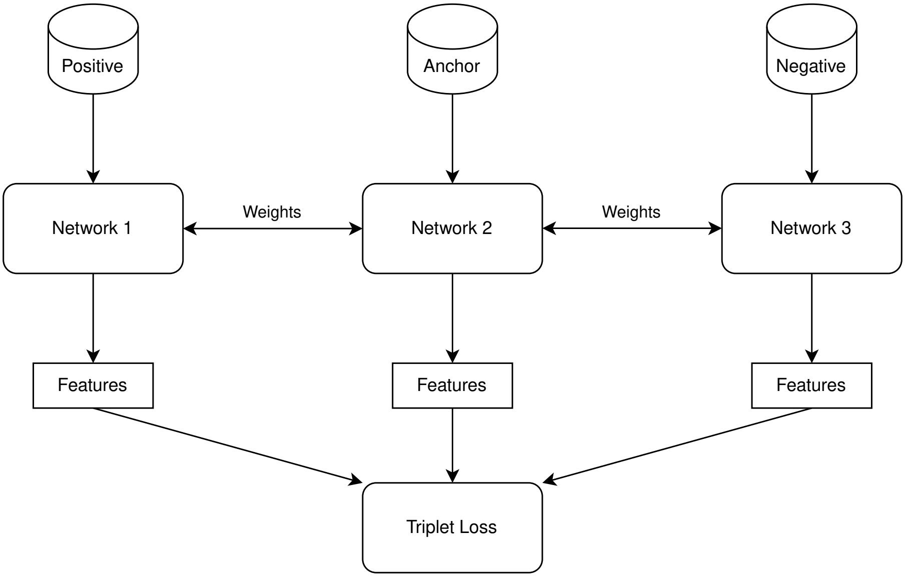

Fig. 2 Architecture of a Siamese Neural Network where the input is a triplet of elements.

- A, P, and N are the input parameters: anchor, positive and negative, respectively.
- α is the margin between the positive pairs and the negative pairs. By default, it is set to 1.
- $e()$ are the embeddings of each input parameter.

2.2 Recommender Systems

Recommender Systems (RSs) are used, as discussed in the introduction, as technological solutions to the information overload, since they help users to filter the most interesting items (in whatever domain the RS is being deployed) according to their preferences. Moreover, because of the prevalence of Internet, they have become indispensable due to their ability to process large amounts of information and make personalized recommendations to users by learning their interests and tastes [41].

Depending on the domain, items may have a different nature, either movies, books, electronic products, or touristic venues. At the same time, while the final objective for any of these systems is the same in any case, they are usually classified depending on how they work with the data, collaborative filtering and content-based being the two most popular and well-known categories.

Content-based (CB) recommender systems analyze the items and/or user features (content) and use them to create user and item profiles to recommend items to the target user that are similar to the ones she liked previously [29]. In order to make recommendations, this type of system uses three main components: a) the content analyzer that pre-processes the information available of the items in order to extract keywords, concepts, or other information; b) the profile learner that, using the content information of the items, builds a profile for every user in the system; and c) the filtering component that matches the user profile against the items in the system.

Collaborative Filtering (CF) techniques, on the other hand, analyze the interactions between users and items to establish patterns between them when making recommendations. These techniques are normally divided into two groups: memory-based that perform the recommendations using the interactions (usually represented as a user-item matrix) in a direct way by computing similarities between users and/or items [35], and model-based algorithms that build a predictive model by approximating the information stored in the preference or interaction matrix [24].

In the first case, the idea behind these algorithms is to recommend to the target user the most appropriate items by exploiting similarities between the rest of the users/items in the system. For this, they build neighborhoods by considering those users/items with the highest similarities, and predict the score for new items based on those similarities and the scores provided by such neighbors.

In the second case, the models approximate the user-item matrix by transforming both users and items into a latent factor space of low dimensionality so that the user-item interactions can be explained (or recovered) by applying dot products in that space. Whereas the concept of similarity is less explicit here, the recommendation is still based on those items that are closer (in the latent space) to the items previously consumed by the target user.

3 Siamese Networks for Recommendation

The Siamese Neural Networks and the Recommender Systems have temporarily coexisted as techniques in Artificial Intelligence research since the early 1990s. However, no study or proposal of the integration of SNNs with RSs were done until more than two decades later, as the first articles that took this approach date back to 2018 ([22, 26, 31, 46, 48]), where each author envisioned different strategies to tackle different problems.

In the next sections, we first present how we collected the papers to be analyzed in this review (Section 3.1), and then we categorize these works based on the recommendation task addressed in the proposal (Section 3.2), the algorithmic approaches considered when designing the neural network (Section 3.3), and how the methods were evaluated (Section 3.4).

3.1 Methodology

In this section, we present how the articles of the state-of-the-art have been retrieved to develop the analysis of the available approaches presented in this work.

We started with an initial study to extract the best key concepts to query the digital libraries. To collect all the articles, two general queries were developed to find all the articles related with the Siamese Networks and the

Table 1 Queries issued to the two digital libraries considered.

Source	**Query**
Scopus	(TITLE-ABS-KEY(”recommender systems”) OR TITLE-ABS-KEY(”recommendation system”) OR TITLE-ABS-KEY(”recommendation”)) AND (TITLE-ABS-KEY(”siamese network”) OR TITLE-ABS-KEY(”siamese neural network”) OR TITLE-ABS-KEY(”twin neural network”))

Web of Science	(
	TI=”recommender systems” OR TS=”recommender systems” OR AB=”recommender systems” OR AK=”recommender systems” OR
	TI=”recommendation system” OR TS=”recommendation system” OR AB=”recommendation system” OR AK=”recommendation system” OR
	TI=”recommendation” OR TS=”recommendation” OR AB=”recommendation” OR AK=”recommendation”
	)
	AND
	(
	TI=”siamese network” OR TS=”siamese network” OR AB=”siamese network” OR AK=”siamese network” OR
	TI=”siamese neural network” OR TS=”siamese neural network” OR AB=”siamese neural network” OR AK=”siamese neural network” OR
	TI=”twin neural network” OR TS=”twin neural network” OR AB=”twin neural network” OR AK=”twin neural network”
	)

Recommender Systems from Web of Science[1] and Scopus[2]. The requirement in both queries was to match at least one word with the same semantic meaning as Recommender System (like ’recommendation’ or ’recommender’) and to match at least one word with the same semantic meaning as Siamese Neural Network (like ’SNN’, ’twin neural network’, or ’Siamese Network’). The actual queries used are shown in Table 1.

A total of 55 articles were found, of which only 24 were classified as valid. This reduction of more than a half of the articles is due to the removal of duplicated articles (where some of them appeared individually and also as part of a conference) but, more importantly, to the fact that making the query so general (to avoid missing relevant papers) some results did not deal with Recommender Systems, but only included those terms in the abstract.

3.2 Recommendation tasks

Siamese Neural Networks and Recommender Systems tend to work in different domains and with different types of input data, depending on the problem to be addressed. When both are integrated, we observe that this remains true even though there are not so many examples in the literature. Table 2 shows the domains used in the analyzed articles, where we considered those that

[1] https://www.webofscience.com

[2] https://www.scopus.com/

Table 2 Categorization of articles according to domains and types of data. T stands for Text, A for Audio, I for Image, and V for Video.

Domain	Article	T	A	I	V
E-commerce	[3, 17, 26, 27]	✓			
Fashion	[15, 38, 46, 48]	✓		✓	
Films	[26, 27, 43, 49]				
Jobseeker	[23, 31]	✓			
Music	[9, 39]		✓		
News	[22]	✓			
Tourism	[44, 47, 49]			✓	
Other	[14, 19, 25, 28, 34, 36, 37, 45, 53]			✓	✓

appeared in [7] as a starting reference. The fashion domain was added to this list, both because of its presence in the surveyed works (as we shall analyze later), but also because of its growing importance in the field [21].

Depending on the domain, we observe different types of input data used by the models to learn when making recommendations. All articles except [23] and [27] make use of metadata for model training. Images stand out in all the articles in the fashion domain [15, 38, 46, 48] and in [19, 34] as input parameters of the Siamese Networks. On the other hand, the use of other multimedia elements, such as audio, is only observed for articles in the music domain [9, 39], whereas the use of video was used less frequently [28]. Finally, texts are also used in different domains [23, 46], where the news domain stands out [22]. However, in some articles in the fashion domain [3, 15] text is also exploited when training the model, but this is actually done in the recommendation part and not when training the Siamese Network, so it is not considered in our categorization.

This analysis of the application domains is consistent with others found in the literature, such as the one from [52] where the authors analyze the use of Deep Learning techniques in RS. There, text, images, audio, and videos (in that order) were the most popular data sources, corresponding to news/reviews, music, and video application fields.

3.3 Algorithmic approaches

In our context, Siamese Neural Networks are used as a tool to generate recommendations. This usage can vary between different problems, as well as the design of the network itself. By studying the literature, they can be classified in four categories: use of the network in the problem, number of input parameters, loss function, and feedforward network used in the SNN.

3.3.1 Use of the network in the problem

In Recommender Systems, it has been observed that there are different techniques to implement the functionality of identifying which items are useful for the user. Likewise, if Siamese Networks are treated as a black box that only extracts the information of the similarity between items, their contribution to

Table 3 Works categorized depending on how they used the network.

Feature Extraction		Prediction
Feedforward	[23, 25, 28, 34, 38, 45, 49]	[9, 14, 17, 22, 26, 31, 36, 43, 48, 53]
Clustering	[3, 19, 39, 44, 47]	
Learning to Rank	[15, 27, 37, 46]	

the recommendation algorithm can be divided into two categories: prediction and feature extraction.

In prediction, only the output of the network itself is used to calculate the similarity and, therefore, the importance of the item to be recommended. In these approximations, the N items with the highest score are returned to produce the predictions.

In feature extraction, on the other hand, the output of the network is used as intermediate data that is concatenated to a larger model that determines the importance of the items when generating the recommendations. There are also several subdivisions in this category:

- Feedforward: the output of the network is used together with other relevant data to make a prediction recommendation on a feedforward network.
- Clustering: the output of the network is used as a feature vector. This is used with any clustering technique like K-means to make the recommendation.
- Learning to Rank: the output of the network is used as a feature vector. This is used with any learning to rank technique (like Bayesian Personalized Ranking or BPR [40]) to generate the recommendations.

Table 3 shows a categorization of the reviewed articles by the different uses of the network.

3.3.2 Number of input parameters

Two distinct models of Siamese Networks are used in the literature based on the number of input parameters: pairs and triplets. This division is equivalent to the one that can be defined when integrating Siamese Networks in Recommender Systems.

It is important to highlight what a triple means in Siamese Networks for recommendation, since we have observed that in some articles they are used inaccurately. A triplet is made up of three elements: anchor, positive, and negative (A, P, N). The anchor is the item or user whose distance to the other two elements of the triplet is to be learned. The positive is an item of the same class as the anchor or an item that the anchor user has liked, and the negative is an item of a different class from the anchor or an item that the anchor user has not liked. Figure 3 shows how the distances change after training a triplet, where the distance between the anchor and the positive are reduced and likewise the anchor and the negative are separated.

In fact, data of the form (item 1, item 2, label) as (I_1, I_2, L), where the label takes the value 0 or 1 if the items belong to the same class or not, it is

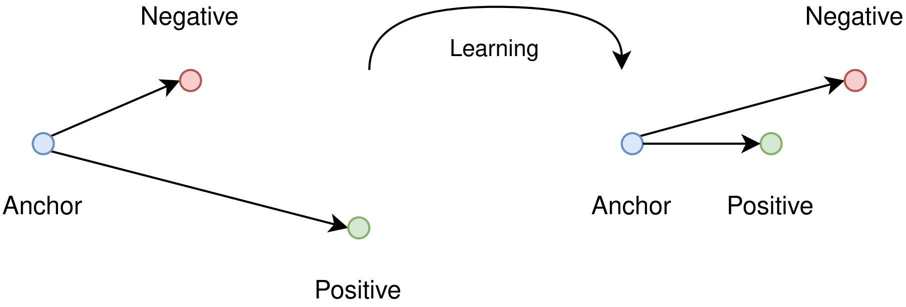

Fig. 3 Distance learning on the training of a Triplet Network.

Table 4 Classification of articles based on the different feedforward networks.

Network	Article
MLP	[14, 25, 37, 43, 45, 53]
CNN	[3, 9, 15, 19, 23, 28, 31, 34, 38, 39, 44, 46, 48]
RNN	[3, 17, 22, 23, 26, 47, 49]
GCN	[27]
Transformer	[36]

not a triplet but a pair. This confusion can be found at least in [49] and, thus, is not considered in our classification as if it is using triplets.

From the 26 articles that are collected in Table 2, 23 of them use pairs and only three of them ([9, 14, 44]) use triplets. In addition, one of them ([9]) besides using triplets, it uses a novel structure of input parameters of the form: an anchor, a positive item, and n negative items $(A, P, N_1, N_2, ..., N_n)$.

3.3.3 Loss function

Research works can also be classified depending on the loss function used to train the Siamese Network. However, we found there is a high correlation between the actual loss function used and the number of inputs to the Siamese Network. Every proposal using the Binary Cross Entropy ([17, 34, 39]) or the Contrastive loss ([19, 23, 27, 28, 31, 43, 48, 49]), use pairs as input data to the network. Similarly, all the works using the Triplet loss ([14, 44]) also use triplets as input.

It should be noted that not all articles describe which loss function was used, whereas in some cases the authors use other loss functions, including custom loss functions, tailored to the specific problem at hand, such as in [37, 47, 53]. Among the other loss functions, it includes the use of Softmax-cross-entropy [26], Point-wise loss [22], Multiple Negative Ranking Loss [36], and Max-margin hinge loss and Categorical cross-entropy loss [9].

3.3.4 Feedforward network used in the SNN

Finally, the articles can be categorized depending on the feedforward network that is used in the Siamese Network, as shown in Table 4. We observe that CNNs are the most popular type of network, although together with MLP and RNN are used almost uniformly throughout the surveyed articles. Graph Convolutional Networks (GCNs) and Transformers, on the other hand, have only been used once ([27, 36]), very recently, which may be an indication that more researchers will try this type of network in the future.

Nonetheless, it is important to note that there are more concrete approaches within each category of feedforward network, that allows for different levels of granularity when presenting this classification. More specifically, within the multilayer perceptron, the use of *multi-armed bandits* [25] as a reinforcement learning technique stands out. In convolutional networks, the use of fine-tuned pretrained models is remarkable: VGG-16 [19, 38], InceptionV3 [28, 46], AlexNet [15], and C3D [28] when using videos. In recurrent networks, different architectures are used: GRU [49], LSTM [3, 22, 23, 26, 47], and Bi-LSTM [17, 23]. Lastly, the transformer model used is SciBERT [36].

3.4 Evaluation settings

To measure the performance of artificial neural networks and Recommender Systems, several evaluation techniques are used in both areas. The main goal for these techniques is to compare different model approaches when they are used for the same problem, to detect which is the best algorithm to solve such problem, or to detect how well each algorithm performs.

In the reviewed articles, we have observed that not all the works share the same final goal for the recommender system. For example, some authors aim at dealing with the long-tail effect [43] or cold-start items [39]. Because of that, they measure different aspects and use different evaluation metrics. Nonetheless, some commonalities can be found in the evaluation methodologies and settings used.

Let us consider, as the first analysis level, the two main types of evaluation methodologies in RS [16]: offline and online. Offline methodologies are carried out with data already collected, by trying to simulate the behavior of users; online methodologies, on the other hand, compare the interaction of various RSs with real users, observing how they influence them. Performing an online evaluation is more expensive and, usually, it is not reproducible, hence not allowing the comparison against other algorithms not included originally in the experiment [4]. Probably for these reasons, in the bibliography of 24 articles analyzed herein, there are only two using online evaluation methodologies: [17, 39].

However, it is important to highlight that all works except [44] report experimental results where an offline evaluation methodology was used[3]. As

[3] The reason why [44] was included even though it does not perform a proper offline evaluation is because this is a position paper where the entire architecture is presented and tested, the only

we show in Table 5, there is no consensus among which evaluation metrics should be used, although this situation also occurs in the overall area of Recommender Systems [50]. While it is true that no metric is strictly better than any other (as it depends on the task the RS is intended to address and how it relates to the user experience), it is not surprising to note that all the metrics reported in these works are measuring something related to the accuracy of the system, in terms of how accurately it predicts the user's preferences. This, however, neglects recent efforts in the RS community to deal with beyond-accuracy evaluation dimensions, such as novelty, diversity, or fairness [2, 8]. In fact, among the metrics shown in this table, we identify those that measure ranking accuracy, either in binary form (like Precision, Recall, HR, MRR) or by considering both the relevance of item for the user and its position in the ranking (NDCG), against those that measure classification accuracy (where the actual rating provided by the user is expected to be predicted, like AUC and Accuracy) [16].

On the other hand, and considering the most commonly used evaluation metrics found in these works (i.e., Recall@K, Precision@K, Accuracy, AUC, F1, NDCG, MRR, and HR), these evaluations follow the trend in the RS community to favor ranking metrics over error metrics – where the former evaluate the quality of the recommendation list instead of the actual predicted value –, as it is well-established that those metrics correlate well with the user experience and satisfaction [32].

4 Discussion

In this section, we introduce some open issues and challenges we have identified after performing the presented analysis on the state-of-the-art of Siamese Neural Networks for Recommender Systems. We first focus this discussion on the analyzed bibliography (Section 4.1) and later (Section 4.2) we perform an experiment where more practical issues would become apparent.

4.1 Open issues and challenges from a bibliographical perspective

Considering that applying SNNs to RSs is a recent development, there are several opportunities that open up to improve this novel research area. First, in terms of algorithmic approaches, it is obvious that the latest techniques from the general application of SNNs need time to be adapted and translated into recommendation. For example, siamese networks with attention [54] or ensemble learning [20] might produce a large positive impact in the predictive accuracy of the recommendation algorithm. Similarly, as noted in Section 3.3.4, recent developments on transformer architectures (such as [55]) could be promising venues to explore when adapting these techniques for recommendation.

missing step is the integration of the SNN in the recommender system (in this case, an image-based travel recommender).

Table 5 Evaluation metrics used in the surveyed research works.

Article	**Evaluation Metric**
Year 2018	
[22]	HR@K, NDCG@K
[26]	Recall@K, MRR
[31]	Accuracy, Precision, Recall, F1
[46]	AUC
[48]	Mean Recall@K
Year 2019	
[15]	AUC
[28]	Recall, Betrayal Rate
[38]	Lift@K
[43]	Precision, Recall, F1, BPREF, LTC, WLTC, TTC
[53]	AUC, ERR, RD, FPR, TPR, FNR
Year 2020	
[9]	Precision, AUC
[19]	Accuracy
[23]	Accuracy
[34]	Precision, Recall, F1, ROC
[37]	HR, NDCG
[47]	Accuracy@K, macro-F1
[49]	Precision
Year 2021	
[3]	Accuracy, AUC
[17]	Precision, Recall, A/B Test
[39]	Accuracy, User Satisfaction
Year 2022	
[14]	Precision, Recall, HR and Average Reciprocal HR
[25]	Accuracy@K, Precision@K, Recall@K, F1@K, ROC
[36]	Precision, Recall, F1, MRR, MAP
[44]	(none)
[45]	AUC, NDCG, MRR, PR-AUC
Year 2023	
[27]	NDCG@K, Recall@K

Also related to the algorithmic approaches, the use of custom loss functions might be seen as a promising challenge in the future, where researchers could adjust or tweak those functions based on domain experts or the desired goal to be addressed by the SNN. Beyond those presented before (which have already been applied to recommendation), there are several examples recently where, for very specific problems, researchers propose custom functions for their problems, as in [51] or [33].

From the recommendation perspective, as mentioned in Section 3.4, it is worrying that none of the works analyzed have considered an evaluation dimension beyond accuracy. Alternative metrics such as novelty, diversity, serendipity, coverage, or fairness are being increasingly investigated in the user modeling and RS communities. They are critical to provide better experiences to the users, but also to make these systems useful while avoiding popularity biases, as this is what they tend to reproduce when focusing on accuracy [5].

Table 6 List of articles where some code was referenced in the paper.

Article	**Year**	**Framework**	**Link** (GitHub repositories)
[22]	2018	Keras	dhruvkhattar/RARE
[26]	2018	TensorFlow	PreferredAI/cbs
[3]	2021	TensorFlow, Keras	marinaangelovska/complementary_products_suggestions
[39]	2021	TensorFlow, Keras	michaelpulis/SnnForCbColdStartMusicRecommendation
[36]	2022	PyTorch	malteos/aspect-document-embeddings
[44]	2022	TensorFlow, Keras	samanthavaca/Travel_Recommender

An important aspect that arose from our analysis was the scarcity in application domains (see Section 3.2). Important domains for recommendation like web pages, social networks, or interfaces are seldom used. Moreover, their impact on critical attributes needed for a working recommender system such as its explainability, context awareness, or preference elicitation has not been considered (yet), and we believe that, taking into account the nature of the SNNs – i.e., based on similarities which are, quite frequently, easy to reason and argue about – they may result in a positive contribution for some of these attributes.

On top of these issues, we present now a reproducibility analysis we did on the reviewed works, in line with recent analyses in the area [4, 13]. As we show in Table 6, only six articles provide access to any type of code, so it can be studied or reused by other peers. Among them, none of them have any documentation or the use of each model is mostly attached to a specific dataset (which may not be known), making it very difficult to actually reuse these models. We observe that in all cases the code is provided through a GitHub repository, which makes it (to some extent) more reliable with respect to other options (such as private hosting or temporary links). It is also worth noting the use of TensorFlow [1] and Keras [11] as the primary machine learning frameworks to develop Recommender Systems with Siamese Networks.

In summary, the open issues and challenges according to the analysis done with the current state-of-the-art can be summarized as follows:

- Adapt latest SNNs techniques and custom loss functions for recommendation.
- Extend the application domains and, in general, how SNNs are applied, beyond recommendation prediction.
- Improve the evaluation methodologies considered, focusing on ranking approaches and reproducibility.

We, therefore, encourage the community to focus on these aspects to increase the impact Siamese Neural Networks may achieve in Recommender Systems in the near future.

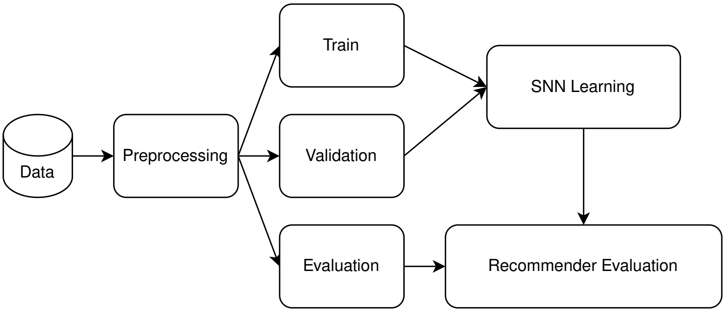

Fig. 4 Workflow followed to test a Recommender System with Siamese Network, where the network is used for prediction.

4.2 Open issues and challenges from an experimental perspective

In this section, we want to dig deeper in some of the assumptions and conclusions exposed in the literature by running an experiment. In particular, we want to test the somewhat established hypothesis that contrastive loss should work better than binary cross entropy with Twin Neural Networks. Other aspects such as how to integrate SNNs in RSs, which loss function to use, or the best evaluation metric to report, as already discussed, have no conclusive response from the community and probably depends on the nature of the problem or the domain and, hence, they are left out of this experimental study.

Moreover, for the sake of reproducibility, we make our code public (see here[4]) and specify the experimental settings in the next section, including how our experiment fits under the classification proposed throughout this review. Later on, we analyze the results and provide some discussion.

4.2.1 Experimental settings

In Figure 4, we show how the data was splitted to be used in our experiment. As we shall describe later, in our experiment the "Use of the network" refers as prediction, which guides how this process should be.

More specifically, in this process the data is partitioned into training, validation, and evaluation subsets. Each one of these data partitions is used in the following stages for learning the recommender system. The training and validation partitions are used when training the Siamese Network, whereas the evaluation partition is used together with the trained Siamese Network to evaluate the recommender system.

Based on this, the network is trained or adjusted (if a pretrained network model is being used) with the training data. The validation data is used during training to check that the network is not overfitting, i.e., that it is generalizing correctly. Furthermore, these validation data allow establishing an early stop condition of the training of the model, if it has not learned anything after a

[4] https://github.com/masternico97/Siamese-Recommender-Systems

number of iterations. Finally, the model is evaluated with the evaluation data. For the evaluation, we decided to use Recall@K and Precision@K metrics, as we observed previously that these are two of the most popular evaluation metrics. This means that we evaluate the first K recommendations offered by the model, and compare whether these suggestions are relevant for the user by checking against the evaluation subset.

The model we selected to benchmark is a content-based recommender system that uses siamese networks to compute item similarities. The objective of this experiment is to compare the use of different feedforward CNN networks (custom, VGG-19 pretrained of ImageNet, and Inception-Resnet-V2 pretrained of ImageNet) and the difference between using the Binary Cross Entropy loss and the Contrastive loss. Taking into account the classification presented before (see Section 3.3), this experiment fits under the *prediction category* for use of the network and *pairs* for the number of input parameters. The other two categories (loss function and feedforward networks) are the variables we want to test.

For this experiment, the input data are pairs of shoes images from the fashion domain. Specifically, the E-commerce Product Images dataset from [30] is used, extracting the images of man shoes. After studying the data, the pairs are created taking into account the *subcategory* of each item, by labeling as similar those images that share the same label and as different those that share no label in common. Moreover, we follow the approximation found in [42], where a statistically significant sample of the images from the dataset are retrieved, and then their K-best recommendations are examined. However, instead of computing the accuracy we report ranking-based metrics, as described before.

4.2.2 Results and Analysis

We show in Tables 7 and 8 the Recall@10 and Precision@10 of the 6 combinations of the experiment as described previously. Theoretically, and according to what the state-of-the-art has established, for the same model, the column showing the model trained using the Contrastive loss should give better results[5] than the one trained with the Binary Cross Entropy loss, because the Contrastive loss is a better suited function to the problem addressed by the Siamese Networks. This is because their objective is to differentiate and not to classify between two elements.

While this assumption is satisfied for two of the models, we observe that, for the VGG-19 model, both Recall@10 and Precision@10 are higher when using the Binary Cross Entropy loss function, indicating that it obtains better results than when Contrastive loss is used. Therefore, we conclude that it is not possible to determine which loss function is better when Siamese Networks are

[5] Note that the values presented in the table are performance values (in this case, Recall and Precision), not the output of the loss function; hence, the higher the value, the better the corresponding model.

Table 7 Recall@10 depending on the loss function and feedforward network, best result underlined.

	Binary Cross Entropy loss	**Contrastive loss**
Custom CNN	0.0114	0.0247
VGG-19	0.0551	0.0200
Inception-ResNet v2	0.0114	0.0230

Table 8 Precision@10 depending on the loss function and feedforward network, best result underlined.

	Binary Cross Entropy loss	**Contrastive loss**
Custom CNN	0.3025	0.3850
VGG-19	0.8100	0.2925
Inception-ResNet v2	0.1725	0.4050

integrated in Recommender Systems, as this may depend on the architecture of the feedforward network.

4.2.3 Summary

By testing variations of the loss function and feedforward networks on a single experiment under comparable conditions (all the algorithms were evaluated on the same data), we have detected an inconsistency with respect to what the literature claims: that Contrastive loss is always better when used in SNNs.

We argue this might be due to inherent properties of Recommender Systems, where users and items have very scarce interactions, which may produce some loss functions to work better than others depending on the recommendation domain or other conditions, such as information sparsity, number of items, quality of their attributes, and so on. Hence, we advocate to explore these aspects in the future and, in particular, to make the experiments as reproducible as possible, to maximize the possibilities to reuse and extend prior models.

5 Conclusions

In this survey, a comprehensive review of approaches where Siamese Neural Networks are integrated in Recommender Systems is presented. Even though the usage of these techniques for recommendation started few years ago, we believe this is a critical moment for this study, as Recommender Systems are widely used, and the data and computational capabilities allow for further and deeper extensions of these and related approaches. In fact, not even the terminology is completely established, as different authors started using *Twin Neural Networks* instead of the now more general term *Siamese Neural Networks*.

As a result of our survey, in our review, we have detected several issues and challenges that the research community could address in the future, such as the lack of beyond-accuracy evaluation dimensions, mostly focused currently on accuracy, and the difficulty to reproduce the results due to a lack of documentation and/or public implementations.

Another contribution of this work is the proposed classification of the literature, where we have categorized the papers on its application domains, the recommendation tasks they were applied, the algorithmic approaches considered (including how the neural network is used to address the problem, the number of input parameters, the loss function, and the feedforward network used in the Siamese Network), and how they were evaluated. A potential practical application of such classification is that researchers and practitioners could use it to, first, identify the gaps in the literature and work on them, and, second, to get an overview of the field and decide which technique is more appropriate for their situation.

Finally, in order to address some of the identified issues, we presented an experiment that tested whether the Contrastive loss function should perform better in a recommendation context. We observed that it actually depends on the feedforward network used for the Siamese Network, hence, opening up further opportunities to bring these communities closer and work together to improve the performance of these approaches for recommendation, where very promising directions lie ahead. One limitation of this experiment, though, is that it has only explored one domain (e-commerce), hence in the future it should be repeated for different and varied domains, such as fashion, films, and tourism.

Several directions of future research emerge from this work. On the one hand, and in line with recent efforts in both Machine Learning and Recommender Systems communities [4], how to improve reproducibility of research works when using Siamese Neural Networks. On the other hand, and evidenced by the analysis presented herein, not all the domains have been investigated equally, so it is worth considering why this occurs and whether some (novel) domains may benefit more of these techniques, in particular in the context of recommendation.

Acknowledgments. This work has been funded by the Ministerio de Ciencia e Innovación (reference PID2019-108965GB-I00). The authors thank the reviewers for their thoughtful comments and suggestions.

Declarations

Conflict of interest

The authors declare that they have no conflict of interest.

Ethical approval

This is a literature review article and does not involve human subject for data collection. There is no need for ethical approval.

Availability of data and materials

The dataset used in this paper is publicly available.

Code availability

The code is available in the following public repository https://github.com/masternico97/Siamese-Recommender-Systems.

References

[1] Abadi M, Agarwal A, Barham P, et al (2015) TensorFlow: Large-scale machine learning on heterogeneous systems. URL https://www.tensorflow.org/, software available from tensorflow.org

[2] Amigó E, Deldjoo Y, Mizzaro S, et al (2023) A unifying and general account of fairness measurement in recommender systems. Inf Process Manag 60(1):103,115. https://doi.org/10.1016/j.ipm.2022.103115, URL https://doi.org/10.1016/j.ipm.2022.103115

[3] Angelovska M, Sheikholeslami S, Dunn B, et al (2021) Siamese neural networks for detecting complementary products. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, pp 65–70

[4] Bellogín A, Said A (2021) Improving accountability in recommender systems research through reproducibility. User Model User Adapt Interact 31(5):941–977. https://doi.org/10.1007/s11257-021-09302-x, URL https://doi.org/10.1007/s11257-021-09302-x

[5] Bellogín A, Castells P, Cantador I (2017) Statistical biases in information retrieval metrics for recommender systems. Inf Retr J 20(6):606–634. https://doi.org/10.1007/s10791-017-9312-z, URL https://doi.org/10.1007/s10791-017-9312-z

[6] Bromley J, Guyon I, LeCun Y, et al (1993) Signature verification using a siamese time delay neural network. In: Cowan JD, Tesauro G, Alspector J (eds) Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]. Morgan Kaufmann, pp 737–744, URL http://papers.nips.cc/paper/769-signature-verification-using-a-siamese-time-delay-neural-network

[7] Burke R, Ramezani M (2011) Matching Recommendation Technologies and Domains, Springer US, Boston, MA, pp 367–386. https://doi.org/10.1007/978-0-387-85820-3_11, URL https://doi.org/10.1007/978-0-387-85820-3_11

[8] Castells P, Hurley N, Vargas S (2022) Novelty and diversity in recommender systems. In: Ricci F, Rokach L, Shapira B (eds) Recommender Systems Handbook. Springer US, p 603–646, https://doi.org/10.1007/978-1-0716-2197-4_16, URL https://doi.org/10.1007/978-1-0716-2197-4_16

[9] Chen K, Liang B, Ma X, et al (2021) Learning audio embeddings with user listening data for content-based music recommendation. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp 3015–3019

[10] Chicco D (2021) Siamese neural networks: An overview. Artificial Neural Networks pp 73–94

[11] Chollet F, et al (2015) Keras. URL https://github.com/fchollet/keras

[12] Chopra S, Hadsell R, LeCun Y (2005) Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA. IEEE Computer Society, pp 539–546, https://doi.org/10.1109/CVPR.2005.202, URL https://doi.org/10.1109/CVPR.2005.202

[13] Dacrema MF, Boglio S, Cremonesi P, et al (2021) A troubling analysis of reproducibility and progress in recommender systems research. ACM Trans Inf Syst 39(2):20:1–20:49. https://doi.org/10.1145/3434185, URL https://doi.org/10.1145/3434185

[14] Faroughi A, Moradi P (2022) Moocs recommender system with siamese neural network. In: 2022 9th International and the 15th National Conference on E-Learning and E-Teaching (ICeLeT), IEEE, pp 1–6

[15] Gao G, Liu L, Wang L, et al (2019) Fashion clothes matching scheme based on siamese network and autoencoder. Multimedia Systems 25(6):593–602

[16] Gunawardana A, Shani G, Yogev S (2022) Evaluating recommender systems. In: Ricci F, Rokach L, Shapira B (eds) Recommender Systems Handbook. Springer US, p 547–601, https://doi.org/10.1007/978-1-0716-2197-4_15, URL https://doi.org/10.1007/978-1-0716-2197-4_15

[17] Guo M, Yan N, Cui X, et al (2021) Deep learning-based online alternative product recommendations at scale. CoRR abs/2104.07572. URL https://arxiv.org/abs/2104.07572, https://arxiv.org/abs/2104.07572

[18] Hoffer E, Ailon N (2015) Deep metric learning using triplet network. In: Similarity-Based Pattern Recognition. Springer International Publishing, Cham, Lecture Notes in Computer Science, pp 84–92

[19] Holder CJ, Ricketts S, Obara B (2020) Convolutional networks for appearance-based recommendation and visualisation of mascara products. Machine Vision and Applications 31(1):1–13

[20] Huang H, Liu G, Zhang Y, et al (2022) Ensemble siamese networks for object tracking. Neural Comput Appl 34(10):8173–8191. https://doi.org/10.1007/s00521-022-06911-4, URL https://doi.org/10.1007/s00521-022-06911-4

[21] Jaradat S, Dokoohaki N, Pampín HJC, et al (2022) Fashion recommender systems. In: Ricci F, Rokach L, Shapira B (eds) Recommender Systems Handbook. Springer US, p 1015–1055, https://doi.org/10.1007/978-1-0716-2197-4_26, URL https://doi.org/10.1007/978-1-0716-2197-4_26

[22] Khattar D, Kumar V, Gupta S, et al (2018) Rare: A recurrent attentive recommendation engine for news aggregators. In: CIKM Workshops

[23] Khatua A, Nejdl W (2020) Matching recruiters and jobseekers on twitter. In: 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, pp 266–269

[24] Koren Y, Rendle S, Bell RM (2022) Advances in collaborative filtering. In: Ricci F, Rokach L, Shapira B (eds) Recommender Systems Handbook. Springer US, p 91–142, https://doi.org/10.1007/978-1-0716-2197-4_3, URL https://doi.org/10.1007/978-1-0716-2197-4_3

[25] Kumari T, Sharma R, Bedi P (2022) A contextual-bandit approach for multifaceted reciprocal recommendations in online dating. Journal of Intelligent Information Systems 59(3):705–731

[26] Le DT, Lauw HW, Fang Y (2018) Modeling contemporaneous basket sequences with twin networks for next-item recommendation. IJCAI

[27] Li B, Guo T, Zhu X, et al (2023) SGCCL: siamese graph contrastive consensus learning for personalized recommendation. In: Chua T, Lauw HW, Si L, et al (eds) Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM 2023, Singapore, 27 February 2023 - 3 March 2023. ACM, pp 589–597, https://doi.org/10.1145/3539597.3570422, URL https://doi.org/10.1145/3539597.3570422

[28] Li Z, Li S, Xue L, et al (2019) Semi-siamese network for content-based video relevance prediction. In: 2019 IEEE International Symposium on

Circuits and Systems (ISCAS), IEEE, pp 1–5

[29] Lops P, de Gemmis M, Semeraro G (2011) Content-based recommender systems: State of the art and trends. In: Ricci F, Rokach L, Shapira B, et al (eds) Recommender Systems Handbook. Springer, p 73–105, https://doi.org/10.1007/978-0-387-85820-3_3, URL https://doi.org/10.1007/978-0-387-85820-3_3

[30] Luhaniwal V (2021) E-commerce product images. URL https://www.kaggle.com/datasets/vikashrajluhaniwal/fashion-images

[31] Maheshwary S, Misra H (2018) Matching resumes to jobs via deep siamese network. In: Companion Proceedings of the The Web Conference 2018, pp 87–88

[32] McNee SM, Riedl J, Konstan JA (2006) Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: Olson GM, Jeffries R (eds) Extended Abstracts Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006. ACM, pp 1097–1101, https://doi.org/10.1145/1125451.1125659, URL https://doi.org/10.1145/1125451.1125659

[33] Nasiri A, Omid M, Taheri-Garavand A, et al (2022) Deep learning-based precision agriculture through weed recognition in sugar beet fields. Sustain Comput Informatics Syst 35:100,759. https://doi.org/10.1016/j.suscom.2022.100759, URL https://doi.org/10.1016/j.suscom.2022.100759

[34] Neve J, McConville R (2020) Imrec: Learning reciprocal preferences using images. In: Fourteenth ACM Conference on Recommender Systems, pp 170–179

[35] Nikolakopoulos AN, Ning X, Desrosiers C, et al (2022) Trust your neighbors: A comprehensive survey of neighborhood-based methods for recommender systems. In: Ricci F, Rokach L, Shapira B (eds) Recommender Systems Handbook. Springer US, p 39–89, https://doi.org/10.1007/978-1-0716-2197-4_2, URL https://doi.org/10.1007/978-1-0716-2197-4_2

[36] Ostendorff M, Blume T, Ruas T, et al (2022) Specialized document embeddings for aspect-based similarity of research papers. arXiv preprint arXiv:220314541

[37] Perera D, Zimmermann R (2019) Cngan: Generative adversarial networks for cross-network user preference generation for non-overlapped users. In: The World Wide Web Conference, pp 3144–3150

[38] Polanía LF, Gupte S (2019) Learning fashion compatibility across apparel categories for outfit recommendation. In: 2019 IEEE International Conference on Image Processing (ICIP), IEEE, pp 4489–4493

[39] Pulis M, Bajada J (2021) Siamese neural networks for content-based cold-start music recommendation. In: Fifteenth ACM Conference on Recommender Systems, pp 719–723

[40] Rendle S, Freudenthaler C, Gantner Z, et al (2009) BPR: bayesian personalized ranking from implicit feedback. In: Bilmes JA, Ng AY (eds) UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009. AUAI Press, pp 452–461, URL https://www.auai.org/uai2009/papers/UAI2009_0139_48141db02b9f0b02bc7158819ebfa2c7.pdf

[41] Ricci F, Rokach L, Shapira B (eds) (2022) Recommender Systems Handbook. Springer US, https://doi.org/10.1007/978-1-0716-2197-4, URL https://doi.org/10.1007/978-1-0716-2197-4

[42] Saxena N, Khosla V, Venkataraman A, et al (2013) Building an image-based shoe recommendation system

[43] Sreepada RS, Patra BK (2020) Mitigating long tail effect in recommendations using few shot learning technique. Expert Systems with Applications 140:112,887

[44] Vaca S (2022) Using siamese neural networks to generate image-based travel recommendations. https://doi.org/10.58445/rars.99

[45] Vijjali R, Bhageria D, Tamhane A, et al (2022) Foodnet: Simplifying online food ordering with contextual food combos. In: 5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD), pp 178–185

[46] Yang Z, Su Z, Yang Y, et al (2018) From recommendation to generation: A novel fashion clothing advising framework. In: 2018 7th International Conference on Digital Home (ICDH), IEEE, pp 180–186

[47] Yu Y, Tang H, Wang F, et al (2020) Tulsn: siamese network for trajectory-user linking. In: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, pp 1–8

[48] Yuan H, Liu G, Li H, et al (2018) Matching recommendations based on siamese network and metric learning. In: 2018 15th International Conference on Service Systems and Service Management (ICSSSM), IEEE, pp 1–6

[49] Yuan W, Wang P, Yuan M, et al (2020) N2one: Identifying coreference object among user generated content with siamese network. In: International Conference on Web Information Systems and Applications, Springer, pp 276–288

[50] Zangerle E, Bauer C (2023) Evaluating recommender systems: Survey and framework. ACM Comput Surv 55(8):170:1–170:38. https://doi.org/10.1145/3556536, URL https://doi.org/10.1145/3556536

[51] Zeng X, Chen H, Luo Y, et al (2019) Automated diabetic retinopathy detection based on binocular siamese-like convolutional neural network. IEEE Access 7:30,744–30,753. https://doi.org/10.1109/ACCESS.2019.2903171, URL https://doi.org/10.1109/ACCESS.2019.2903171

[52] Zhang S, Yao L, Sun A, et al (2019) Deep learning based recommender system: A survey and new perspectives. ACM Comput Surv 52(1):5:1–5:38. https://doi.org/10.1145/3285029, URL https://doi.org/10.1145/3285029

[53] Zhao Y, Qiao M, Wang H, et al (2019) Tdfi: Two-stage deep learning framework for friendship inference via multi-source information. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, IEEE, pp 1981–1989

[54] Zheng H, Gong M, Liu T, et al (2022) Hfa-net: High frequency attention siamese network for building change detection in VHR remote sensing images. Pattern Recognit 129:108,717. https://doi.org/10.1016/j.patcog.2022.108717, URL https://doi.org/10.1016/j.patcog.2022.108717

[55] Zhu Z, He X, Qi G, et al (2023) Brain tumor segmentation based on the fusion of deep semantics and edge information in multimodal MRI. Inf Fusion 91:376–387. https://doi.org/10.1016/j.inffus.2022.10.022, URL https://doi.org/10.1016/j.inffus.2022.10.022