IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.0122113

Improving Cold Start Stereotype-Based Recommendation Using Deep Learning

NOURAH A. AL-ROSSAIS AUTHOR[1]
[1]King Saud University, College of Computer and Information Sciences, Information Technology (IT) Department, KSA
(e-mail: nalrossais@ksu.edu.sa)

Corresponding author: Nourah A. Al-Rossais (e-mail: nalrossais@ksu.edu.sa).

ABSTRACT Recommendation engines constitute a key component of many online platforms. One of the most challenging problems facing a recommender system is that of cold start, namely the recommendation of items from the catalogue to a new unknown user, or the recommendation of newly injected content to existing users. It is established that incorporating metadata describing the item or the user leads to better cold start performance. Multiple independent findings point to the value of pre-processing the metadata to generate a new set of coordinates to aid the underlying recommendation algorithm; one of such pre-processing techniques, stereotyped features, have been shown to improve standard recommendation algorithms. Deep learning and complex neural networks have also been widely utilized in recent years in recommender systems, but their application and performance benchmarking in cold start scenarios is still a matter of ongoing research. This article reports on the application of deep learning neural networks to the stereotype driven framework for addressing cold start in recommender systems. We discuss the performance using a range of metrics, covering accuracy, value content of ranked lists but also serendipity and fairness of recommendations, with the latter becoming an important metric and risk factor for the online platform offering the recommendations. Our findings indicate that a multi-layer neural network substantially improves cold start accuracy performance metrics, despite the recommendations displaying worse fairness and serendipity traits. The work discusses for which metrics/scenarios stereotyping features may still be useful also for the class of more sophisticated deep learning recommender systems.

INDEX TERMS Cold start, deep learning, fairness of ranked lists, neural networks, new item problem, new user problem, recommender systems, recommender systems evaluation, serendipity of ranked lists, stereotypes.

I. INTRODUCTION

RECOMMENDER systems (later referred to as RS) are key to the success of online platforms and retailers. These systems suggest relevant items, such as products, songs, and content, to the users of the platform overcoming the information overload of scouting through vast catalogues. From e-commerce websites, to music streaming and social media platforms, a good (or poor) RS can affect not just the platform success and user experience, but also the firm's reputation. Netflix's success in the on-demand TV streaming services has been attributed to its innovative recommendation algorithm [4]. Amazon on the other hand, in 2018 received substantial criticism and negative press coverage for its recommendation algorithm reviewing job applicants' resumes, which was found to be biased [5]. Over the last decade of research in the field of RS the focus on the evaluation of recommendations has broadened: while accuracy of single recommendations or ranked lists remains a key component, other factors are being recognised almost as significant. For example serendipity, fairness and diversity. Serendipity and diversity consist in the ability of the recommendation algorithm to surprise the user with unexpected items, while variety and diversity measure the overspecialization of RS, see [6]. A RS with high accuracy but overspecialized will be unable to satisfy the user consistently over time, or to suggest items that may be enjoyed but that a user may not necessarily be looking for. Fairness of recommendations, a metric that only recently is starting to be employed (see [5], [7]) aims at mitigating risks to the firm hosting the RS, and also to instigate confidence to item providers that their products or content will not be over ranked by others unfairly.

One of the most challenging tasks that a RS needs to perform is to provide recommendations to new users joining the platform (users that have not expressed any preferences), or recommending new items injected in the platform (items that have not been rated by any user). These two problems known in the literature as cold starts have been discussed in detail by the author in [1], [2] and references within.

The recent advancement of artificial intelligence, and in particular deep learning, have been mirrored also in its applications to RS. Substantial work has been put in the use of AI in user profiling and preference discovery, in particular, [8], [9] reviews how different AI approaches have been explored to make recommendations, focusing mostly on deep learning and reinforcement learning. These techniques have gained popularity in the RS field due to their ability of finding complex and non-linear relationships between users and items. Nonetheless, deep learning models are usually non-interpretable, need a conspicuous amount of data, and are computationally expensive.

The rate of advancement of the application of deep learning to cold starts has been less pronounced and cold start remains an area where ad hoc approaches, often dataset dependent approaches, are developed. In this article we wish to bridge such gap by combining the stereotype based approach introduced in [1] with a deep learning framework. The stereotype based approach can be viewed as a "model free" manner to address cold start recommendations, and in this context also as a way to reduce the dimensionality and computational cost of a deep neural network.

A. SCOPE OF WORK

Recent multiple independent research findings have highlighted the advantages of stereotypes in enhancing out of sample performance of recommendation systems during extreme cold start. The research gap that this paper wishes to fill is founded upon two distinct pillars. The first involves extending the methodology of stereotypes in pure cold start scenarios, which were independently developed in recent works [1], [11], and integrating them into deep learning frameworks. In the conclusions and future work of [1], the author anticipated testing the cold start stereotype approach with more sophisticated algorithms. This paper serves to bridge that very gap.

The second source of novelty lies in the limited exploration of deep learning algorithms in pure cold start scenarios, despite the extensive research of such approach in fully warm recommendation systems. Notably, the performance that is reported in this paper of deep learning approaches in cold start scenarios exhibits lower serendipity and fairness, suggesting signs of overspecialization. This, in itself, is a noteworthy finding worth reporting, and that in principle corroborates the thesis of [12].

We present and discuss the performance of this combined approach in recommendations during cold start via experimental results obtained on a public dataset of movie reviews, movie features and user features merged from Movielens 1M and IMDb, [10]. We also present metrics that go beyond the simpler accuracy, and intend to demonstrate the effect on accuracy, serendipity and fairness of the methodology researched. In summary, the primary contributions of this paper are as follows: bridging the identified research gap as outlined in the future work of [1]; offering additional evidence regarding the advantages and limitations of employing stereotypes; demonstrating the practical benefits and risks associated with implementing the stereotype framework within a deep learning solver; and emphasizing the evidence, benefits, and potential risks linked with utilizing a deep learning framework in the context of cold start scenarios. These techniques are readily applicable in various real-world applications of recommendation engines that encounter cold start issues when user and item features are accessible.

II. BACKGROUND AND RELATED WORK

Cold start is one of the most challenging areas of operation of a RS, and several studies are dedicated to addressing it. [1] reviews the new user and new item problems, the existing literature, and introduces a novel way to address and improve cold start recommendations via stereotypes. The study suggests that the creation of general representations of users and items, known as stereotypes, from the available information, without relying on individual preferences or ratings, facilitates the development of a coordinate system that enhances two crucial aspects. The stereotyped features provide a substantial dimensionality reduction of the problem, and more importantly such generalized coordinate system enhances the ability of machine learning algorithms to discover robust preference patterns on a training dataset. The use of stereotypes lead to better out of sample predictions compared to the case in which the same machine learning algorithm is used with the original unstereotyped metadata.

Finally the framework of stereotypes can be applied to pretty much any technique where the RS formulation can be expressed in a functional form where the predicted user item interaction depends on the user features, the item features and previous ratings, for all users and all items. This is elaborated upon in mathematical detail in [20].

Recently and independently [11] reached similar conclusions, where the authors propose to use clustering via K-NN to generate stylized model features (i.e. stereotypes) and found that simple RS exhibit better performance out of sample in recommendation tasks when trained on such stereotyped features.

In a recent review by [21], the authors extensively discuss the application and evaluation of deep learning in recommendation systems. Among the 270 references cited, only two explicitly mention the issue of cold start, with no significant attention dedicated to the problem within the review itself. This observation highlights that although substantial research has been conducted on the application of deep learning in recommendation systems, the area remains relatively under-explored when specifically considering the challenges associated with new user and new item cold start problems.

The challenge of cold start continues to be a dynamic and pertinent area of research, as extensively discussed by [24]. Notably, the authors have proposed the development of a language model autoencoder to effectively tackle the new user problem.

[13] shows how recommendations, not just cold start ones, can be improved compared to standard RS by using a mixture of graph based clustering of users and deep learning, the experimental results are carried out on a private dataset. [14] coupled matrix factorization techniques, collaborative filtering, and neural networks to achieve better recommendations in the new item cold start problem compared to simpler RS. Advances in using deep learning during cold start have been less conclusive to date, in particular [12] reports some troubling findings, where sophisticated and often extremely computationally expensive algorithms perform no better than simpler methods on out of sample testing. Deep learning algorithms are prone to over-fitting, creating very accurate models on in-sample training data that often fail to perform as well out of sample. Using many layers in the neural network, each layer with many neurons, and training the network with vast dimensions of the input vector often leads to such problems.

The challenge of maintaining a balance between the accuracy and fairness of recommendations, which will be clearly defined in subsequent sections of this manuscript, has recently been investigated by [22]. Their proposed methodology focuses on training recommendation systems to achieve improved fairness without significantly compromising accuracy, emphasizing the use of deep learning and meta-learning architectures. In a recent study by [23], the authors acknowledge the predominant focus of previous works on fairness-aware recommendation systems on a predetermined set of users, typically those with established interactions (often warm-start users). Nonetheless, recommendation systems frequently encounter more complex fairness challenges concerning new or cold-start users, owing to their limited interaction history. The authors aimed to facilitate the transfer of knowledge from a fair model trained on warm users to the target cold users, with the aim of enhancing fairness performance.

III. PROBLEM FORMULATION

A. DATASET AND STEREOTYPED FEATURES

In the Movielens IMDb merged dataset [10] users express their liking of an item via an integer value rating on a scale from 1 to 5 to the movies they watched and wished to rate. The data is prepared by assembling a dataframe whose rows contain: the user id, the movie id, the rating that such user gave to such movie, all user features (either in their original shape encoded or stereotyped), all the item features either in their original shape encoded or stereotyped). The procedure to generate stereotypes is described in details in [1]. The number of columns describing an item and a user for this particular dataset go from over 150 entries of mixed types (categorical and numerical) to half of that number when stereotypes are used as encoded features; for instance one of the most descriptive features of the items is the item ‘genre‘ which comprises of 24 different categories in the original metadata and is encoded into just eight relevant stereotypes. Other relevant features like the budget of the movie, or popularity of the principal cast are stereotyped into three to four distinct classes. In the context of this work we will use the results obtained in [1] as a starting point, the reader is referred to such previous work for the procedure of generating stereotypes for different feature types (categorical versus numerical metadata), the validation of stereotypes and their representation/encoding.

B. NEW USERS AND NEW ITEMS EXPERIMENTS

The new user experiment is produced as follows: for each user left out in the test dataset of a given experiment each model generates recommendations and ranked recommendation lists as if the user had not rated any items (a new user). The resulting recommendations are compared to the actual preferences expressed by that user and accuracy metrics generated (rating value predicted vs actual, ranked preferences predicted vs actual, consumption non consumption of each item etc.). The new item experiment is produced in a similar manner. For each item left out in the test dataset the reviews of all users for such item are blanked out as if they never occurred. In any given experiment each model generates a rating prediction for each user for such left out items and in a similar manner as with the new-user experiment, we asses how far apart the system’s predictions are from the actual ratings.

All metrics presented in this article discussed in III-D, for each algorithm, are the result of a 6 fold validation experiment. In each of the 6 folds, we put aside a group of one third of the users (users that have at least 10 reviews) and a group of one third of the items (items that have been reviewed at least 10 times) to constitute the new user and new items test sets on which the models for that fold are assessed. This is done in a series of six successive experiments with a one sixth overlap across folds.

C. ALGORITHMS

Following the work in [1] we train simple machine learning models using the stereotyped coordinates as explanatory variables for the ratings as our established benchmarks. These benchmarks were proven to outperform both the same algorithms using unprocessed metadata features as well as standard matrix factorization methods enriched with metadata features such as the SVD with implicit feedback and metadata features of [15], [16]. In particular the findings of [1] become the current benchmarks in order of complexity:

1) Naive (NV): simply predicting that a new user will be rating an existing item with the item's in sample mean rating, or that a new item being rated by existing users will be rated via the in sample mean rating of the user.
2) Linear Regression (LR).
3) Neural Network Regression (NNR). A single layer neural network.
4) Extreme Gradient Boosting (XGB). The algorithm developed by [17].
5) Singular Value Decomposition with implicit feedback and metadata (SVD++).

This is a wide range of benchmarks, going from the simplest model free (NV) to what is often considered state of the art in RS and cold start (SVD++).

In this paper we introduce a deep neural network (DNN). In many literature contexts deep neural networks are often not specified enough to make the results easily reproducible, as highlighted also in [12]. There are very many details in a deep learning architecture that affect the final model, so in this article we will specify our proposed deep learning architecture in full. Our proposed DNN is composed by:

- A simple input and normalization layer that applies a series of weights to the input vector (i.e. the stereotyped coordinates of each item).
- Two consecutive hidden layers, each made up of: i-a dense fully connected layer, which is the most general purpose layer of a neural network tasked with a regression prediction, and ii- a dropout (or regularization) layer. The dense layers in our problem were chosen to have 128 neurons with linear activation function. This is the result of applying several rules of thumb available in the neural network configuration literature where the ideal number of active neurons should be chosen based on the dimension of the input vector, the output vector and whether some neurons may be deactivated during training. The most successful technique to alleviate the tendency of over-fitting of a neural network is to de-activate randomly the participation of a certain fraction of neurons of the dense layers during stages of the fitting process. This technique suggested by [18], and implemented via the dropout layers, prevents the network from overspecializing its neurons, by forcing neurons within a layer to display more adaptability to changing inputs. This approach also prevents certain layers to co-adapt to correct mistakes from prior layers, as discussed in [18].
- An output layer that condense the prediction into a single real valued number within an upper and lower bounds. In the context of the dataset at hand, and for all models tested, we treat the predicted rating as a continuous real variable between 1 and 5 rather than the alternative approach of generating a label from 1 to 5.

For readers who prefer a visual representation, Figure 1 illustrates the schematic flow chart of the modeling pipeline, presenting the stages of stereotyped and full model preparation. Both models are subsequently tested in the new user/item experiment.

In the present paper, we conduct a comprehensive analysis of the cold start performance spectrum (a diverse range of metrics and properties). Our benchmarking specifically emphasizes the comparison between the use of stereotypes and original metadata features for the models introduced earlier. While this study addresses the relative benefits and limitations of these specific approaches, it is important to note that the exploration of cold start solutions extends beyond the scope of this research. Future work will aim to assess the effectiveness of our findings in the context of other recommendation approaches that employ alternative methodologies to address cold start issues. This comparison will provide further valuable insights into the applicability of different strategies for handling cold start problems.

D. METRICS

When assessing the performance of a RS there are many metrics that are relevant. As discussed in section I accuracy metrics of single recommendations have been historically the most published, and for that reason we will show in our evaluation the root mean squared error (RMSE) as well as the mean absolute error (MAE) of single recommendations.

In [1], [2] we also focused on item consumption, i.e. in the new user problem try to predict which items in the full catalog will be consumed (in the case of movies watched and rated), and in the case of a new item try to predict which of the existing users will consume that item. While these are important metrics in the

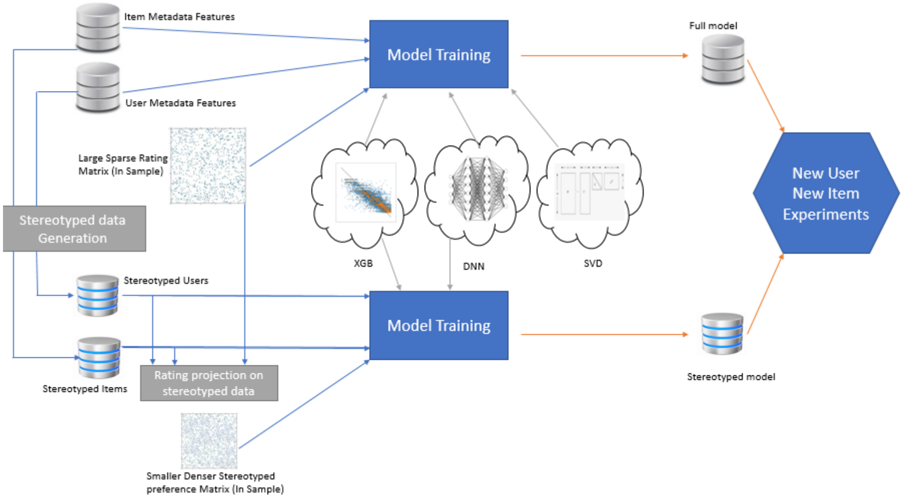
Item Metadata Features
User Metadata Features
Large Sparse Rating Matrix (In Sample)
Stereotyped data Generation
Stereotyped Users
Stereotyped Items
Rating projection on stereotyped data
Smaller Denser Stereotyped preference Matrix (In Sample)
Model Training
XGB
DNN
SVD
Model Training
Full model
New User New Item Experiments
Stereotyped model

FIGURE 1. Visual representation of the full and stereotyped modeling pipeline, Visual representation of the complete stereotyped modeling pipeline, illustrating the approach's capacity to facilitate the interchangeable use of models in a standardized manner.

business application of RS they are less utilized in the RS academic literature and in this context we will omit such metrics, also because the Hit Rate introduced next is a proxy for such consumption metrics applied to ranked lists.

Going past the simplest predictive accuracy metrics, top-N rank accuracy metrics measure the ability of a recommendation algorithm to produce a list of N items whose content is of interest to the user and also the ordering of items close to the rank that the user would express. In this context we will focus on two of the most descriptive top-N metrics: the Hit Rate (HR) and the Normalised Discounted Cumulative Gain (NDCG). The HR measures the proportion of successfully recommended items in top-N list. In other words, in the new user experiment if a user rated one of the top-N we recommended among his/her top-N, that is considered a "hit"; in the new item experiment, if the users that are selected as top users for the item also consumed the item in their actual top-N, then it is considered a "hit". The second metric of accuracy for a top N list is aimed at representing both whether a recommendation constitutes a hit and also the value of such hit. In a non-binary rating context, a recommendation that ranks higher in the top-N list holds more value than one that leads to a lower expressed rating and appears lower in the list. This valuation is compounded by the tendency of many users to avoid delving deep into lengthy recommended lists. The metric that serves this purpose is the normalised discounted cumulative gain (NDCG) as introduced in [19] and expanded in details in [2]. In this context we will not repeat the various equations that lead to the definition of the NDCG, which can be found in [2], [19], it is sufficient to remind that the NDCG measures how effective a ranking system is in determining the position of relevant items. It leads to a real valued quantity in the interval [0.0, 1.0]; with zero being the limiting case of a top-N list with no hits, and 1.0 the case of all entries being a hit and all correctly ranked as per the user's expressed rating.

Serendipity is a desired property of a list of recommendations, it occurs when the list contains items that are relevant and somehow unexpected, in other words when the RS does not overspecialize its recommendations. For example, if a user rated positively a war movie and as a result the top-N recommendations generated by the RS were all famous war movies, we may have a good NDCG in this list for this user, but the RS would not be serendipitous. The list would lack variety, will not open up the user to other branches of the content catalogue. In the literature several definitions of metrics that can represent serendipity have been advanced, often these are tied up to the particular dataset analyzed, but to the author's knowledge there is not a widely accepted metric as discussed in [2]. Instead we will resort to the metric introduced in [1], [2]. Through a Lasso regression analysis (or a similar feature ranking methodology), we can focus on the top metadata features that describe an item and determine the relative importance of stereotyped features. We can then introduce a proxy for serendipity by measuring the diversity and variety of the recommended list in comparison to the full range of potential feature values for such identified list of top features. In particular for the Movielens/IMDb data we discovered that among the most representative features sits the 'genre' of the item. So counting the ratio of the total number of genres represented in a ranked list to the total genre available will allow us to obtain a proxy for serendipity and variety. This operative definition fits perfectly with the example of the war movies above, if the ranked list is overwhelmingly populated of the same genre that may be a one off highly accurate list, but will rank low according to our serendipity proxy (SER). When dealing with a complex categorical feature, i.e. a feature that may assume a large number of different labels in a non-predefined non-unique manner, we need to address the fact that the same label may have a different weight for different items. As discussed in [1] we have to introduce a cutoff constant k that determines the lowest weight each label has in a ranked list to be considered significant. For example in the Movielens IMDb dataset the genre feature is a complex categorical one, each item can be labelled with one or many genres labels. The limiting case of $k = 0$ means that all possible genre labels encountered in the ranked list will make it in the feature variety metrics that is then proxied to SER. The higher the k the higher the minimum weight required for a label to be added to the feature variety count.

Finally, as discussed in I, academics and businesses are becoming more aware about the negative impacts to content providers and reputation risks of an unfair RS. A simple but effective way in which one could approach the fairness problem is by looking at the variety of the top-N lists produced over a given new users or new items test set. A RS that overspecializes its accuracy to the detriment of fairness would show the tendency to always offer the same popular items. This can be quickly assessed by taking the total number of items in the union of all top-N lists and dividing by N, we indicate this metric as $\mathrm{L_{UI}N}$. While measuring the number of different items that are proposed in the top-N lists give a simple operative way to establish relative fairness between recommendation algorithms, [7] provides a more general operative definition for the fairness of a RS; [7]'s definition of fairness focuses on the items that appear the least among the top-N recommendations for all users, it isolates the t% such items over the global new items population and calculates a proxy of their score in the ranked list. This can be done by computing for each item the Mean Discounted Gain (MDG) for item i at relevance N:

$$\mathrm{MDG_i} = \frac{1}{||\mathrm{U}||} * \sum_{\mathrm{u} \in \mathrm{U}} \frac{\delta(\mathrm{z_i^{(u)}} <= \mathrm{N})}{\log(1 + \mathrm{z_i^{(u)}})} \tag{1}$$

In equation (1) $\mathrm{z_i^{(u)}}$ is the ranking position obtained by item i in the ranked list of user u, the function $\delta(\mathrm{x})$ is the indicator function taking value of 1 if x is true and 0 otherwise. The summation is across all users u in the test set U. By taking a statistics, for example the simple mean, of the $\mathrm{MDG_i}$ across all items i in the t% of least recommended items [7] states that we can obtain a metric of fairness. We argue that such metric could only be used to compare RS algorithms exhibiting similar $\mathrm{L_{UI}N}$. When different algorithms generate substantially different size of unions of unique items in their top-N lists we argue that such a metric should be scaled accordingly. We therefore propose a fairness metric as follows:

$$\mathrm{FRN} = \mathrm{L_{UI}N} * < \mathrm{MDG_i} > \tag{2}$$

Where $< . >$ stands for a statistical operator, for example the sample mean. The higher the value of FRN the higher the fairness of the RS, i.e. the least recommended items have a higher chance of showing in up in some ranked lists, and there are more of such items.

IV. RESULTS

A. SINGLE RECOMMENDATION ACCURACY

Tables 1 and 2 display the RMSE and MAE accuracy metrics of the new user and new item experiments discussed in III-B for all algorithms introduced in III-C. The NV model is featureless and therefore there is no difference in its predictions using base features vs stereotyped features. With regard to the SVD++ with metadata we report only its standard implementation on original metadata and not an extension on stereotypes that would have to be formulated carefully and it's outside the scope of this work, and hence only the metrics for base features are populated for such algorithm.

From these simple accuracy metrics it is clear that the DNN is the best performing algorithm; in the new user problem it can achieve the same level of accuracy on standard base features as simpler algorithms can via the aid of stereotyped features. When the DNN is trained on stereotyped features it achieves an even larger improvement. Slightly different situation for the new item problem, where the DNN is still the best performing algorithm by a large margin, but the extra improvement obtained via the stereotypes is minimal.

Some might argue that stereotypes could be perceived as an intelligent and sophisticated data pre-processing tool, akin to an embedding. A deep neural network, when sufficiently intricate, might be capable of extracting comparable feature transformations in its hidden layers, yielding similar outcomes. However, this hypothesis is challenging to validate due to the opaque nature of deep neural networks and the inherent difficulty in interpreting their concealed layers.

TABLE 1. Accuracy performance metrics for the new user problem.

	Model	Base Features	Stereotype Features
RMSE	NV	0.963	-
	LR	0.940	0.938
	NNR	0.918	0.906
	XGB	0.913	0.901
	SVD++	0.924	-
	DNN	0.901	0.887
MAE	NV	0.772	-
	LR	0.743	0.742
	NNR	0.724	0.712
	XGB	0.721	0.710
	SVD++	0.736	-
	DNN	0.709	0.691

TABLE 2. Accuracy performance metrics for the new item problem.

	Model	Base Features	Stereotype Features
RMSE	NV	1.01	-
	LR	0.939	0.934
	NNR	0.928	0.917
	XGB	0.926	0.918
	SVD++	0.905	-
	DNN	0.894	0.891
MAE	NV	0.812	-
	LR	0.741	0.736
	NNR	0.735	0.727
	XGB	0.738	0.729
	SVD++	0.727	-
	DNN	0.714	0.711

B. RANKED LISTS ACCURACY

When moving to the ranking accuracy metrics (HR and NDCG), the variety and serendipity (SER) as well as the fairness (FRN) metric we can restrict our attention to the three most diverse algorithms, namely the XGB, SVD++ and DNN; the standard models LR, NNR, XGB are very close to each other and represent different statistical approaches from the field of regression in machine learning and their performance cluster substantially around similar values. Also in [1] we demonstrated that XGB with stereotyped features was the one with the best behaviour among them, we can therefore without any loss of generality omit the LR and NNR models and keep the best performing of the three, namely XGB using stereotyped features as our stereotype based model benchmark.

In rank driven experiments we focus on a range of top-N lists to study the effect of the depth of the list on the metric and algorithm performance, in particular we show experiments with ranked lists spanning from 10 items to 20, 30, 50 and 100.

Figure 2 shows the HR of the ranked lists produced by our different RS algorithms for the new user and new item experiments. The performance of the matrix factorization-driven SVD++ consistently falls short in both new item and new user experiments across various ranked list sizes. In contrast, the deep neural network (DNN) displays nuanced behavior concerning the utility of stereotypes. The experimental results indicate distinct dynamics between the new user and new item cold start scenarios. For smaller ranked lists, the inclusion of stereotypes appears to be of relevance, as even the XGB with stereotypes marginally outperforms the DNN based on the base features for the shortest list. However, as the list length increases, the effectiveness of stereotypes in boosting performance diminishes. Notably, the DNN performs well with only minor variations, suggesting that the use of stereotypes or base metadata features has a limited impact. This conclusion is especially pertinent for the performance of the DNN models in the new item cold start scenario.

The same findings are confirmed with the analysis of the normalized discounted cumulative gain, NDCG, represented for both the new user and new item experiments in figure 3 as a function of the list size and of the RS algorithm. The DNN algorithm outperforms all others independently of the length of the ranked list. While in the new user case the extra ranking accuracy of the DNN is of the order of a few percentage points compared to that of a simpler XGB driven RS, and the use of stereotyped features provides an extra small boost in predicting ranked lists, for the new item case the situation is substantially more complex. The DNN outperforms the other algorithms by a large margin, especially at the top-30 list, and its NDCG of ranked lists when trained on base features even outperforms the same model trained on stereotyped features.

These observations align with the trends identified in the analysis of simpler accuracy metrics, suggesting that advanced deep learning networks might assimilate similar patterns of stereotyped features through training on base metadata. Consequently, the role of stereotypes in such models could become potentially redundant, particularly when confronted with larger lists. This implies that the increased capacity and learning capability of deep neural networks allow them to grasp and integrate complex feature interactions, minimizing the distinct advantage provided by explicitly included stereotyped features.

C. SERENDIPITY AND FAIRNESS

When moving to the serendipity metric (SER), in addition to the variability due to the length of the top-N list considered, and the algorithm and base features considered, we have one further parameter for the experiment, the constant k that determines the cut-off of the classes considered for complex categorical features. In this context we will limit our results to no filter ($k = 0$) and to the more restrictive case where the label has to account for at least a weight of fifty percent for at least one item (e.g $k = 0.5$).

Figure 4 illustrates the serendipity of ranked lists by showing the percentage of the feature's labels covered by the average top-N ranked list. In the new user experiment, the SER is computed across all top-N ranked lists recommended to new users, comprising existing items. For the new item experiment, all the ranked lists consist of potential new items recommended to existing users, with the metric computed accordingly.

The graphs depict how quickly a ranked list increases its SER, for example in the new user case and $k = 0$ we have typically top-10 lists that span 60% of the possible labels of the complex categorical feature and it takes a list of length 100 to cover the full catalogue of labels. In the case of this dataset labels are not equally frequent, some are substantially more infrequent than others; for example only a very small percentage of items has been labeled as 'documentary' and so it takes a larger list

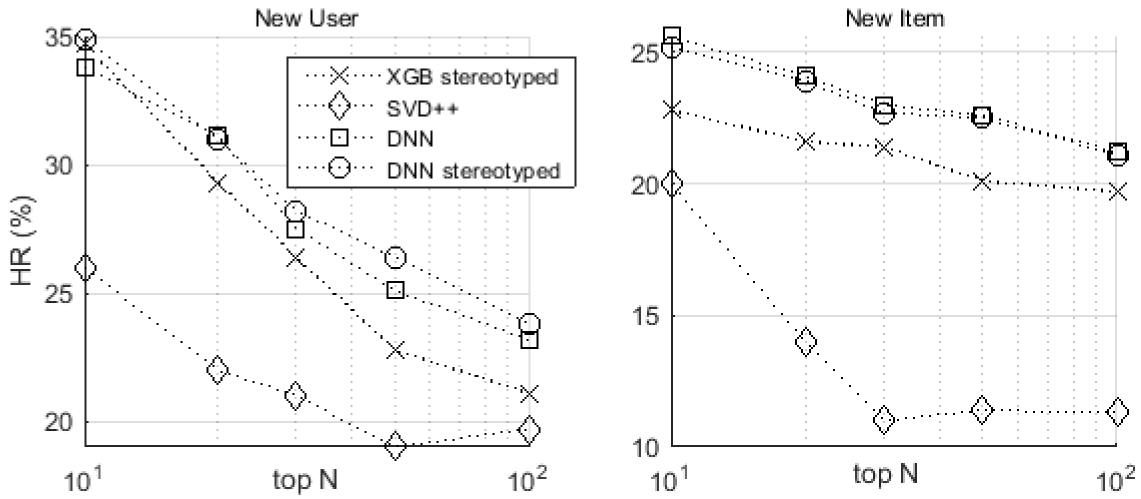

FIGURE 2. Hit Rate in percentage terms for the new user and new item experiments as a function of the length of the top-N ranked lists. The figure highlights the superiority of the DNN algorithm and the smaller effect due to the use of stereotyped features instead of base features.

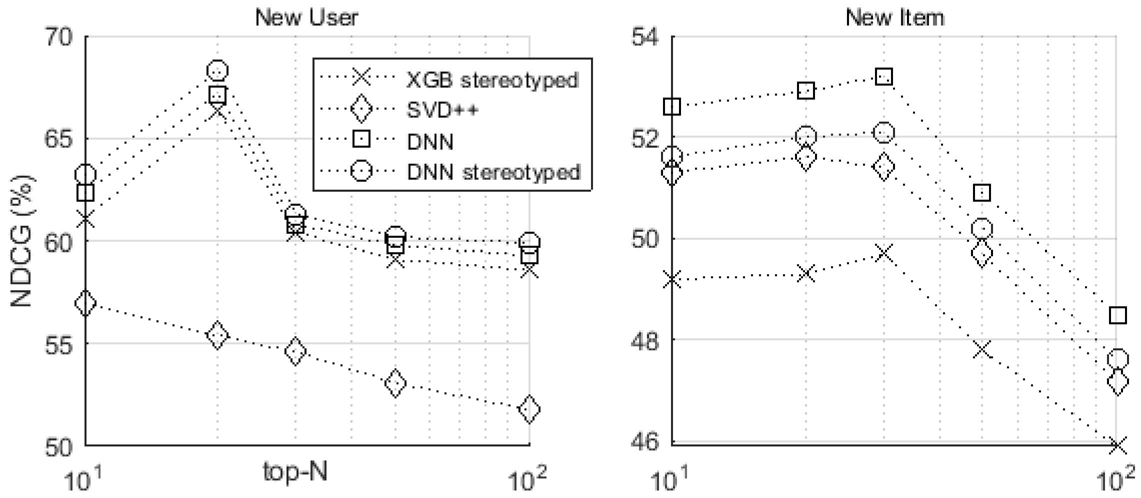

FIGURE 3. NDCG in percentage terms for the new user and new item experiments as a function of the top-N ranked lists. The figure highlights the superiority of the DNN algorithm under all circumstances, and the fact that while stereotypes improve performance in the new user case, the DNN trained on base features is overperforming any other algorithm in the new item case.

to have some representation of that label. At higher k filters the same label may have a weight that is too small to be included in the SER count. In the new item case the number of labels covered by the typical ranked lists are lower than those of the new user case, and even with the top-100 lists and k = 0 they do not reach the full set of available labels, this is clearly because in the test set of new item there are fewer interesting items with infrequent labels and all the algorithms trained have been trained to maximise the accuracy metrics, hence they are somewhat reluctant about placing niche items (like documentaries for example) in the top-N lists, something that would marginally improve SER but potentially damage accuracy.

Figure 4 also highlights that there is little material difference in the SER metric across the three algorithms: XGB, DNN with base features, DNN with stereotyped features; for completeness and to illustrate this point further we show the values for the new user case and new items case for the top-20 list with k = 0 in table 3. This is the combination with the largest variability in SER among algorithms, and it confirms that while the effect of the model accounts for a small percentage improvement in the SER of this average list (i.e. moving from XGB to DNN), the effect of stereotypes is much weaker on this metric, and while it still has some effect for XGB for the DNN it appears to be detrimental. Also in all cases at small lists and k = 0 the DNN appears to have less SER than the standard XGB algorithm, this is also explained when we look at the proxy of list variety

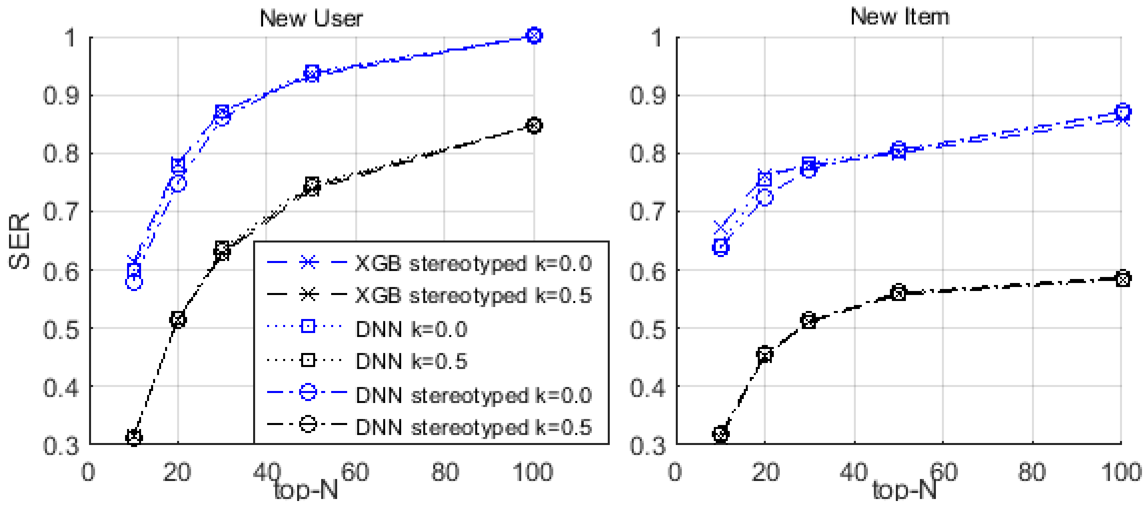

FIGURE 4. SER as percentage of feature's labels covered by a top-N recommended list. Effect of top-N list length and *k* label filter. The algorithm used only accounts for a small percentage change in SER.

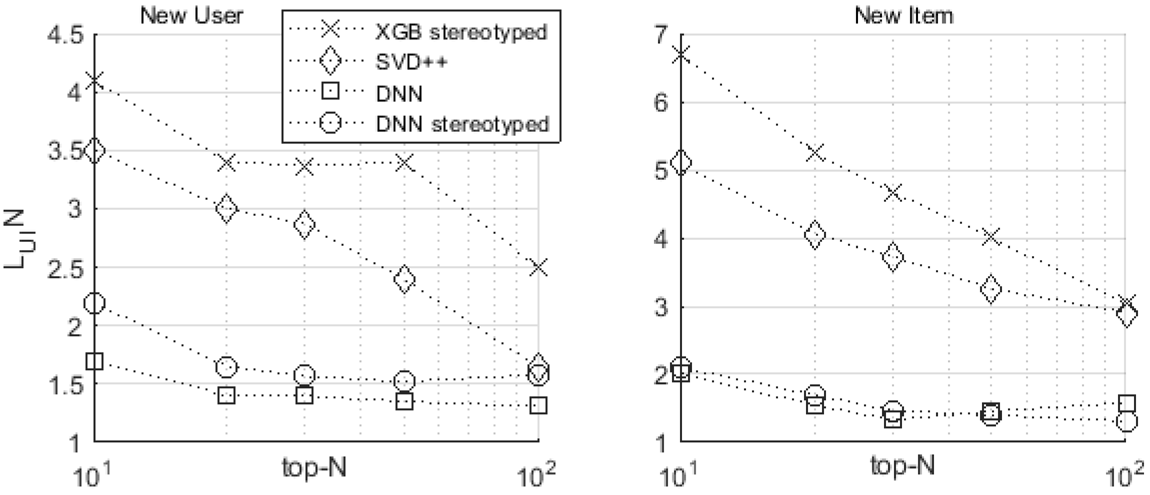

FIGURE 5. Total unique items covered by all the top-N recommended lists ($L_{UI}N$) as a function of N and of the algorithm. The DNN all show very low number of distinct items that compose the lists, hence high specialization.

$\mathrm{L_{UI}N}$ next.

With regard to the fairness of our RS algorithms we show both the simple naive count of distinct items in the top-N list, as a multiple of N, $\mathrm{L_{UI}N}$, as well as the FRN metric defined earlier in section III-D. The $\mathrm{L_{UI}N}$ shows which RS algorithm have the tendency to narrow their lists more; for example a RS could achieve excellent accuracy if to any new unknown user it recommended always the same top items, for example the items with the best ever ratings (e.g. in the movie case The God-Father, Star Wars etc). A low $\mathrm{L_{UI}N}$ is an indication of the lack of variety of the lists and as list grows of their fairness. Figure 5 shows that the DNN's top-N lists tend to be overspecialized offering in total much fewer items from the catalogue compared to the non neural network algorithms. The overspecialization could also be seen in the SER, and in light of figure 5 we have an explanation for the smaller SER of the low top-N lists of the DNN. The DNN algorithms keep their top-N lists narrow, which improves their accuracy metrics to the detriment of serendipity and fairness, as it can be concluded from figure 6. Figure 6 highlights that fairness, as represented by the FRN metric of the top-N lists, is substantially lower for the DNN algorithms compared to the simpler XGB algorithm at every N. The adoption of stereotyped features in the DNN improves the FRN for every N in both the new user and new item cases, with the largest lists showing better fairness metrics. This observation introduces an intriguing problem that we delve into more deeply in the conclusions, pertaining to the trade-off between accuracy and fairness that RS developers often face.

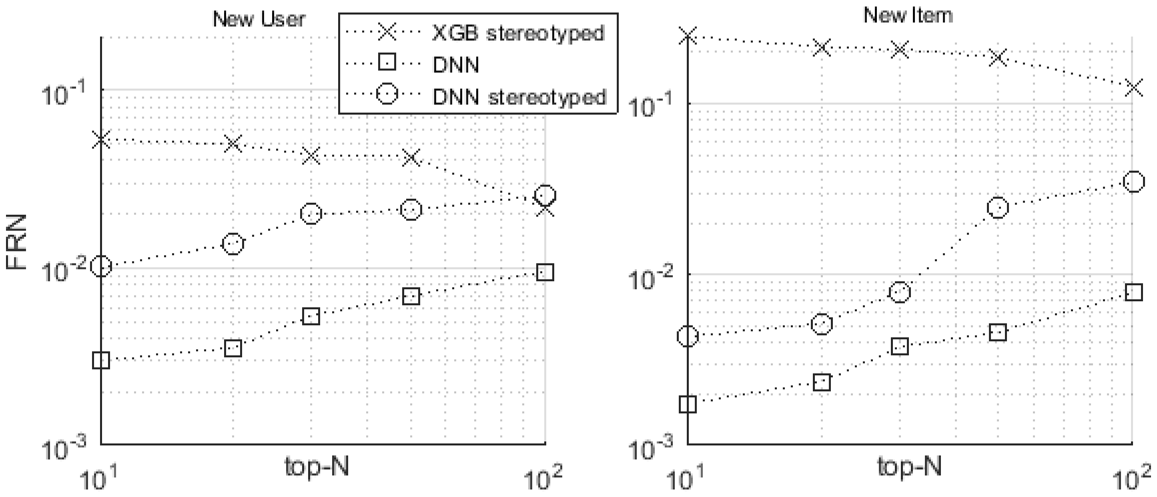

FIGURE 6. Fairness FRN metric of the worst 20 percent items in the union of all top-N lists as a function of N and of the algorithm. The DNN's high specializations impact negatively the fairness of the recommendations, with stereotypes improving substantially over the base features.

TABLE 3. SER values for the new user and new items problems and top 30 list with filter $k = 0$.

	Model	Base Feat.	Stereotyped Feat.
New User	XGB	0.778	0.783
	DNN	0.777	0.748
New Items	XGB	0.755	0.762
	DNN	0.753	0.724

V. CONCLUSION, KEY STRENGTHS, LIMITATIONS AND FUTURE WORK.

In this research article we have presented the application of a deep learning neural network for cold start recommendations, using both base and stereotyped metadata features. At a time where deep neural networks are receiving ever more attention due to their prediction abilities, we have documented several important findings that can be summarized as follows:

- the deep neural network driven RS displays much higher accuracy of single recommendations, as well as of ranked lists compared to other RS algorithms.
- When focusing solely on such accuracies, we find that stereotyping metadata features adds little value to a deep neural network, giving small improvements in the new user case, and potentially none to little degradation in the new item case. We have suggested that this observation, which was consistently present in the accuracy metrics of single recommendations and ranked lists, may be due to the ability of a complex neural network to absorb from the training data the same information content embedded in the stereotypes. This is contrary to what was previously documented, where stereotypes uniquely improved recommendations when simpler underlying algorithms were considered.
- On the other hand, we document that deep neural network driven recommendations suffer from overspecialization in their top-N ranked lists. The overspecialization leads to a lower serendipity for these systems but most importantly to a much reduced fairness of the recommendations. In this case stereotyping metadata features appears to improve substantially on the fairness metrics of a deep neural network using simple metadata.

This work shows how deep learning can become an important source of extra accuracy compared to standard recommendation methodologies, and how at the same time the RS designers need to put in place a series of heuristics to ensure the higher accuracy achieved does not come to the full detriment of the RS fairness. For an institution or business adopting a new RS it is an interesting and potentially challenging problem to balance the desire to reach a high level of accuracy, especially when presenting users with ranked lists, with the potential risks of operating an unfair RS. In this paper we have also revised and modified a recently proposed metric to asses fairness of ranked lists, and shown how stereotypes may be used to improve fairness by damaging very little the extra accuracy arising from the adoption of deep learning.

We can summarize the key strengths and limitations of the proposed approach, with a view toward real-life and business applications, as follows.

Key strengths:

- The incorporation of stereotyped constructs has demonstrably provided an additional source of accuracy and improved fairness/serendipity characteristics. This benefit would particularly aid platforms that are unable to invest extensively in re-

searching and developing a new recommendation system based on deep learning.
- The utilization of deep learning has led to enhanced prediction accuracy for both single recommendations and ranked lists during the cold start phase. This paper highlights that such an approach would be a compelling choice for businesses looking to develop a robust recommendation system, especially when compared to more traditional approaches like SVD++, which are now widely used in practical applications.

Limitations:

- A common critique from practitioners in the industry is directed at deep learning approaches due to the inherent complexity, which often limits the interpretability of the recommendations. This lack of transparency can potentially hinder the understanding of the system, opening businesses and stakeholders to risks that are challenging to quantify, such as reputation and fairness concerns.
- While the approach demonstrates improved prediction accuracy, the complexity associated with deep learning models may require considerable computational resources, making implementation more challenging for organizations with limited infrastructure or expertise.

The impact and significance of this work in real-life applications are multi-faceted. It extends beyond providing an enhanced user experience during the cold start for both new users and new item providers, ensuring a balance between maintaining adequate levels of fairness and serendipity in the generated ranked lists. Moreover, the findings underscore the potential for customization of the underlying recommendation model, catering to the specific needs and preferences of diverse user groups.

In the context of businesses seeking to improve, modernize, or develop a recommendation engine to address the challenges associated with cold starts, the distinctive feature of this work lies in its provision of a comprehensive view of recommendation system development with easy to replace internal models. This comprehensive approach is especially beneficial compared to other works where each model is developed in isolation, making it a feasible and valuable resource for medium and small businesses seeking to enhance their recommendation capabilities without the risk of investing in a single model whose results may not be reproducible.

This work opens up several promising avenues for future research. One of the most pressing lines of inquiry involves benchmarking the performance of the proposed approach against alternative methodologies designed to address cold start challenges, as presented in existing literature. With the established understanding that accuracy and ranked accuracy alone should not dictate the focus of a recommendation system, future benchmarking efforts should prioritize the careful examination of the trade-off between accuracy/ranked accuracy and the serendipity/fairness of a recommendation engine. Our observations suggest that deep learning models might skew recommendations towards accuracy at the expense of other essential metrics. We posit that models built on predicting ratings and minimizing prediction errors may inherently prioritize accuracy, highlighting the need for novel objective functions that can strike a better balance among various key metrics. Exploring alternative objective functions for the in-sample optimization of the recommendation algorithm remains a relatively unexplored area, one that we intend to pursue in our future research endeavors. Finally the performance of the various models should be assessed in the context of a warming up environment, such as when new users engage with the system or new items receive feedback from users. Assessing the performance in these warming up phases that move past the cold start can provide valuable insights into how the models perform during the initial stages of user-item interactions.

ACKNOWLEDGMENT

This research project was supported by a grant from the "Research Center of the Female Scientific and Medical Colleges", Deanship of Scientific Research, King Saud University.

REFERENCES

[1] AlRossais, N., Kudenko, D. & Yuan, T., "Improving cold-start recommendations using item-based stereotypes," in User Model User-Adap Inter, 31, 867–905, 2021, https://doi.org/10.1007/s11257-021-09293-9.

[2] N. Al-Rossais, Intelligent, Item-Based Stereotype Recommender System. Ph.D. dissertation, CS, University of York, York,2021.

[3] Zangerle E., Bauer C. Evaluating Recommender Systems: Survey and Framework. ACM Comput. Surv. 55, 8, Article 170 (August 2023), 38 pages. https://doi.org/10.1145/3556536.

[4] Gomez-Uribe C.A., Hunt N. The Netflix Recommender System: Algorithms, Business Value, and Innovation.. ACM Trans. Manage. Inf. Syst. 6, 4, Article 13 (January 2016), 19 pages. https://doi.org/10.1145/2843948.

[5] Kodiyan, A. A. An overview of ethical issues in using AI systems in hiring with a case study of Amazon's AI based hiring tool.. Researchgate Preprint (2019): 1-19.

[6] Bushra A., Awajan. A., Fraihat S. Survey on the Objectives of Recommender Systems: Measures, Solutions, Evaluation Methodology, and New Perspectives.. ACM Comput. Surv. 55, 5, Article 93 (May 2023), 38 pages.

[7] Zhu Z., Jingu. K., Trung N., Aish F, Caverlee J. Fairness among New Items in Cold Start Recommender Systems.. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021). Association for Computing Machinery, New York, NY, USA, 767–776.

[8] Zhang, Q., Lu, J. Jin, Y. Artificial intelligence in recommender systems.. Review Article. Complex Intell. Syst.7, 439–457 (2021).

[9] M. M. Afsar, Trafford C., Behrouz F. Reinforcement Learning based Recommender Systems: A Survey.. ACM Computing Surveys 1, 1 (June 2022), 37 pages.

[10] ALRossais, N. Kudenko, D. iSynchronizer: A Tool for Extracting, Integration and Analysis of MovieLens and IMDb Datasets.. UMAP'18 Adjunct: 26th Conference on User Modeling, Adaptation and Personalization Adjunct, July 8-11, 2018, Singapore.

[11] Khodabandehlou, S., Hashemi Golpayegani, S.A. and Zivari Rahman, M. An effective recommender system based on personality traits, demographics and behavior of customers in time context..Data Technologies and Applications. 2021, Vol. 55 No. 1, pp. 149-174.

[12] Dacrema, M.F., Boglio, S., Cremonesi, P. and Jannach D. A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research.. ACM Trans. Inf. Syst. 39, 2, Article 20 (April 2021), 49 pages.

[13] M. Rostami, M. Oussalah and V. Farrahi. A Novel Time-Aware Food Recommender-System Based on Deep Learning and Graph Clustering.. IEEE Access, vol. 10, pp. 52508-52524, 2022, doi: 10.1109/ACCESS.2022.3175317.

[14] Wei, J., He, J., Chen, K., Zhou, Y., Tang, Z. Collaborative filtering and deep learning based recommendation system for cold start items.. (2017). Expert Systems with Applications, 69, 29-39.

[15] Frolov, E., Oseledets, I. Hybrid SVD: when collaborative information is not enough. In: Proceedings of the 13th ACM Conference on Recommender Systems, Copenhagen, Denmark, pp 331–339 (2019).

[16] Hadash, G., Shalom, O.S., Osadchy, R. Rank and rate: multi-task learning for recommender systems. In: Proceedings of the 12th ACM Conference on Recommender Systems, Vancouver, British Columbia, Canada, pp 451–454 (2018)

[17] Chen, T. Scalable and flexible gradient boosting. Online resource. https://xgboost.ai/(2016).

[18] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. . Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), (2014), 1929-1958.

[19] Jarvelin, K., Kekalainen, J. Cumulated gain-based evaluation of IR techniques. ACM Trans. Inform. Syst. (TOIS) 20(4), 422–446 (2002).

[20] AlRossais, N., "Warming up from extreme cold start using stereotypes with dynamic user and item features", submitted to ACM International Conference on Recommender Systems (RecSys 2023).

[21] Wu, L., He, X., Wang, X., Zhang, K. and Wang, M."A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation". IEEE Transactions on Knowledge and Data Engineering, 35(5), pp.4425-4445. (2022)

[22] Wei, T. and He, J. "Comprehensive fair meta-learned recommender system." In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 1989-1999). 2022.

[23] Li, Y., Wang, D., Chen, H. and Zhang, Y. "Transferable Fairness for Cold-Start Recommendation." arXiv preprint arXiv:2301.10665. 2023.

[24] Heidari, N., Moradi, P. and Koochari, A. "An attention-based deep learning method for solving the cold-start and sparsity issues of recommender systems. Knowledge-Based Systems, 256, p.109835. 2022.

NOURAH A. AL-ROSSAIS received the Ph.D. degree in computer science from the University of York, York, UK, in 2021.

Since 2021, she has been an Assistant Professor with College of Computer Sciences at King Saud University. In 2022, she has been assigned head of business unit in computer science college at King Saud University.

Her primary field of research interest is Artificial Intelligence (AI). Within AI, she is interested in problems related to machine learning and data mining, and their interdisciplinary applications to recommendations especially in business field.