Migration Letters
Volume: 20, No: S6(2023), pp. 499-508
ISSN: 1741-8984 (Print) ISSN: 1741-8992 (Online)
www.migrationletters.com

Cold Start Reduction Using Residual Neural Networks in Recommender Systems

Layth AL-Mahadeen[1], Ghayth AlMahadin[2], Ahmad Sami Al-Shamayleh[3]

Abstract
Recommender Systems (RS) are extremely important in some industries because they can generate significant revenue when used efficiently or serve as a significant way to differentiate from competitors. There are three major categories of models for accomplishing the RS: collaborative filtering methods (CFM), and content-based methods (CBM). The cold start problem (CSP) is one of the most problems still facing many researchers. where the CSP describes the difficulty in making recommendations when the users or items are new, continues to be a significant challenge for CF. Traditionally, this problem has been addressed by conducting an additional interview process to determine the user (item) profile before making any recommendations. This paper presents a hybrid model for CSP reduction based on the Residual Neural Networks (ResNets). We used the ULMFiT dataset based on the Arabic Wikipedia corpus with (30k) vocabulary. Different scenarios are used for evaluating the experiments using a supervised machine learning algorithm. The experiment results showed the ability of the proposed method to reduce the CSP, with an F1 score reaching (96.96 %).

Keywords: *Cold Start Problem, Neural Collaborative Filtering, Content-based models, Residual Neural Networks (ResNets), CAMeLBERT embedding, Machine Learning.*

1. INTRODUCTION

With the rise of social media and the spread of e-commerce sites over the last few decades, RS has grown in importance in our lives [1]. From e-commerce like suggesting interest articles, to online advertising based on their preferences, RS is now unavoidable in our daily online journeys [2]. In other words, the RS are algorithms that suggest relevant objects (books, movies, hotels, and so on) to users in different industries like suggesting movies, books to read, and products to buy [3]. RS is crucial in several sectors because it increases interactive activities and generates significant revenue when it is efficient, and also serves as a significant way to differentiate from competitors [4]. There are three major categories of models for accomplishing the RS [5]: Collaborative Filtering Methods (CFM) [6], Content-Based Methods (CBM) [7], and hybrid methods.

CFM is a method that generates new recommendations solely based on previous interactions between users and objects [8]. The "User-Item Interactions Matrix" (U-IIM) is where these interactions are saved (The central idea behind CFM rules is that previous U-IIMs are sufficient for detecting comparable consumers and/or comparable objects and making predictions based on these estimated proximities [9], [10]. The CFM can be categorized into two classes: Memory-Based models (ME-B) and Model-Based (MO-B)

[1] Full Time Lecturer, Al- Balqa' Applied University, Department of Financial and Administrative Sciences, Al-Karak, Jordan, layth.mahadeen@bau.edu.jo
[2] Assistant Professor. Al Ahliyya Amman University, Department of Networks and Cybersecurity, Amman, Jordan
[3] Assistant Professor. Al Ahliyya Amman University, Department of Networks and Cybersecurity, Amman, Jordan

models [11]. ME-B models, which rely on finding the nearest neighbors, work directly with recorded interaction values, assuming no model [12]. ME-B models are again categorized into two sub-models: The User-Based (UB) model and the Item-Based (IB) model [13]. The UB model is working for rating a new object by locating other users in the user's neighborhood who have previously rated the same object. If a new object receives positive ratings from the user's surroundings, it is recommended to the user, while the IB model creates an object-neighborhood out of all similar objects that the user has previously rated. The user's rating for a different new object is then predicted by calculating the weighted average of all ratings present in a similar object neighborhood.

While the MO-B models are predicated on the assumption of an underlying "generative" model that explains U-IIM and attempts to discover it to make new predictions [4]. Using various data mining and machine learning algorithms, MO-B systems create a model for predicting the user's rating for an unrated object. They do not use the entire dataset when making recommendations but rather extract features from it to build a model. The MO-B technique was born as a result. These techniques also necessitate two steps for prediction: the first is to build the model, and the second is to predict ratings using a function (f) that takes the model defined in the first step and the user profile as input [14]. The primary advantage of CFM is that no information about users or objects is required, allowing them to be used in a variety of situations [15]. Furthermore, for a fixed set of users and objects, new interactions recorded over time bring new information and improve system effectiveness [16]. But on the other hand, the CFM suffers from the cold start problem (CSP) because it only considers past interactions to make recommendations [17]: it is impossible to recommend anything to new users or a new object to any users, and many users or objects have too few interactions to be efficiently handled [18]. When the RS is unable to draw any conclusions from the available data, which is insufficient, the CSP occurs. CSP is a condition in which the system is unable to generate efficient recommendations for cold (or new) users who have rated no or only a few objects. When a new user joins the system or when new objects (or products) are added to the database, this usually happens [14]. CBM, as opposed to CFM, uses additional information about users and/or objects [3].

The cold start problem (CSP) in CBM is much less severe than in CFM: new users or objects can be described by their characteristics (content), and thus relevant suggestions for these new entities can be made [19]. This disadvantage will logically affect only new users or objects with previously unseen features, but once the system is old enough, this has little to no chance of occurring.

Many works and studies on RSs have been conducted to provide some models that obtain accurate predictions and recommendations and to reduce the CSP using machine learning and deep learning, where in the [21] study, the authors proposed a classification-based model based on the neural deep learning architecture, returning both rating predictions and their reliabilities, where the extra information (prediction reliabilities) is used in a variety of relevant CFM areas such as detection of shilling attacks, recommendations explanation, or navigational tools to show users and object dependencies. Furthermore, users were graciously provided with recommendation reliabilities such as "probably you will like this film", "almost certainly you will like this song", and so on. Experiments were carried out using four well-known public datasets (Movielens 100K, MovieLens 1M, MyAnimeList, and Netflix). The experimental results demonstrated an improvement in the quality of individual rating predictions, which maintains recommendation results and opens the door to a set of relevant CFM fields. While the authors of the [22] study proposed a product RS based on the "Neural Collaborative Filtering (NCF) algorithm". Customer purchase data was used as implicit feedback data in the proposed product RS. They gathered the dataset from one of Indonesia's marketplaces in (2019). The data set consists of (11.918) implicit feedback transaction data in the form of customer purchase data. Customer data, product data, and customer purchase data have all been used.

Implicit feedback provides reliable data for the development of recommendation systems. The experiment results showed that NCF outperforms the other CFM in terms of performance.

In the same context, in the study [23], the authors proposed a model called BENEFICT to address two issues that most review-based RSs face. Whereas the use of traditional word embeddings may influence prediction performance due to their inability to model the dynamic nature of word semantics, the second disadvantage is the black-box nature, which obscures the explanations behind each prediction. They used the BERT, multilayer perceptron, and maximum subarray problems, respectively, to derive contextualized review features, model user-object interactions, and generate explanations. They ran experiments with the datasets "Toys and Games, Digital Music, Yelp-Dense, and Yelp-Sparse." According to the experimental results, the BENEFICT consistently outperforms other cutting-edge models by a margin of nearly (7%). Finally, the authors [24] proposed a hybrid CFM that uses both models to learn knowledge from implicit feedback data in parallel. Where data was used to map the embedding vectors representing user and object information. The generalized matrix factorization of these embeddings element-wise product, whereas the neural network takes as input a (2-D) interaction map formed by stacking two vectors. The element outputs are then concatenated to produce an accurate estimate of the user-object correlation. The results of experiments on standard datasets such as "MovieLens, Yelp, and Pinterest" showed that it outperformed several baselines.

Several studies attempted to solve the cold start problem (CSP) by recommending random objects (RRI) to new users or new objects to new users (random strategy), common objects to new users, or new objects to most active users (maximum expectation strategy), a set of different objects to new users or a new object to a set of different users (exploratory strategy) [20]. Even so, the most of related studies faced that the systems cannot draw enough inferences for users or items about which it has not yet gathered sufficient information. The major contributions of this paper are summarized as follows:

- We reduce the impact of the cold start problem (CSP) in Collaborative Filtering Methods (CFM).
- We constructing a hybrid model between each of the Memory-Based models (ME-B) and Model-Based (MO-B) models.
- We used the Residual Neural Networks (ResNets) framework with the CAMeLBERT to generate contextualized word embedding.
- We conducted extensive experiments on the ULMFiT dataset based on Ar Wikipedia corpus with (30k) vocabulary.
- We evaluated the results based on the supervised machine learning using the KNIME Analytics (KA) platform, where we used each of the (“Support Vector Machine (SVM), Decision Tree (DT), Probabilistic Neural Network (PNN), and the K Nearest Neighbor”) classifiers.

The remainder of this paper is organized as follows. Section (2) demonstrates the proposed method. Section (3) describes the details of the dataset used in this study, and the experiments that were used for testing the performance of the proposed method. The experimental results are presented in Section (4). Section (5) contains the discussion. Finally, Section (6) concludes the paper.

2. ABOUT THE DATASET USED FOR EXPERIMENTATION

Arabic is a major world language yet is underrepresented on the Internet and there is a lack of resources for Arabic NLP work. The proposed method used the ULMFiT dataset based on Ar Wikipedia corpus with (30k) vocabulary. This Arabic ULMFiT dataset was built using fastai based on selected articles from Arabic Wikipedia with (100) tokens or more. The included dataset for classification is from the study of [25] for Hotel Arabic-Reviews Dataset Construction for Sentiment Analysis Applications.

3. METHOD AND PROPOSED WORK

For reducing the impact of the CSP in the recommendation systems, the proposed method used the ResNet algorithm, which is a Convolution Neural Network (CNN) architecture that, unlike other architectural models, does not degrade in performance as the architecture becomes more complex. Convolutional layers are present in all layers. Its architecture includes a building block known as the residual block. Figure (1) depicts the overall layout of our project.

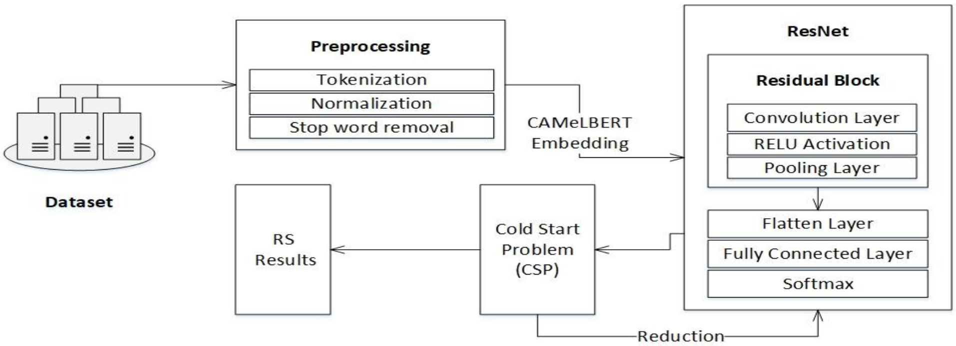

Figure 1. General Framework

3.1. Preprocessing

In the proposed method, the data is preprocessed to make it suitable for machine learning and to improve the model's accuracy and efficiency. Preprocessing is a critical step in NCF for reducing the CSP. Table (1) shows the all stages in the Preprocessing level:

Table 1. Preprocessing stages

Phase	Description
Tokenization	The process of dividing a text into smaller units is called tokens. Words or characters can be used as tokens. The task is to divide the entire text string into distinct terms.
Normalization	The standard form is applied to all of a word's conjugations. Normalizer is given a set of guidelines to follow to produce a uniform format for the proposed dataset.
Stop word removal	Removing stop words that are commonly used in the language but are unrelated to the context in which they are found. These words can be removed from the text without affecting the work's performance.

3.2. Embedding Module

Word representation or word embedding is used as input features to represent text data in the model when using machine learning or deep learning algorithms for NCF. Because computers cannot detect and analyze raw text data, words, phrases, and documents are represented numerically in natural language processing. TF-IDF, Glove, Bag of Words, Word2Vec, FastText, and other embedding methods exist. In this study, we used the CAMeLBERT to generate contextualized word embedding, which is consistent with BERT's goal. In this paper, the embedding modules were trained on a large amount of text data, allowing them to comprehend the context of the input sentence and produce a dense [26]. CAMeLBERT has (37) different pre-trained models for various tasks, each trained on a different dataset. The model used in this study is bert-base-arabic-camelbertmix-sentiment, which was trained on (17.3B) words written in CA, DA, and MSA.

3.3. Residual Neural Network (ResNet)

ResNet is a well-known deep neural network model introduced by [27]. The degradation problem, which occurs when network performance declines as network depth increases, is addressed by the ResNet model by skipping connections between two or three layers that add input x to the output. The residual block is a special building block in the ResNet architecture. According to [27], the residual block is defined as follows:

$$\mathrm{y} = \mathrm{F}(\mathrm{x}, \{\mathrm{Wi}\}) + \mathrm{H}(\mathrm{x}) \tag{1}$$

Where:

- x and y are the residual block's input and output.
- F (x, WI) represents the residual mapping to be learned,
- H(x) represents identity mapping.
- All convolutional layers are combined with the ReLU function, as defined by Equation (2):

$$\mathrm{ReLU}(\mathrm{x}) = \max(0, \mathrm{x}) \tag{2}$$

3.4. Residual Blocks

The ResNet Block was used to construct the high-level representation by reading the input's word embedding. ResNet introduces identity shortcut connections that bypass one or more layers. ResNet block has two convolution layers followed by a ReLU function, the kernel size, and the number of filters of the all-convolution layer (3) and (64), respectively. When the input and output dimensions are the same, the identity blocks can be used right away. Identity mapping was discovered to achieve the fastest error reduction and the lowest training loss. Assume that H(x) is an underlying mapping. The residual functions F(x) can be approximated using Equation (3):

$$\mathrm{F}(\mathrm{x}) = \mathrm{H}(\mathrm{x}) - \mathrm{x} \tag{3}$$

3.5. Convolution Layer

The convolution layer is important in the CNN design. Convolution is a mathematical process that applies a convolution filter to input data to create a feature map. It uses a (3 3) filter, a feature map that predicts the class to which each feature belongs, to extract the feature mapping from the text representation. To reduce the dimensionality of the feature map, we perform a pooling layer after a convolution operation. This allows the proposed method to reduce the number of parameters, reducing training time and preventing overfitting.

3.6. RELU Activation

The Classification layer in standard ResNet architectures is made up of three parts. The first is the Flatten layer, which takes the output of the last residual block after applying ReLU and Average-pooling. Convolution and pooling layers both generate (3D) volumes, but a fully connected layer produces a (1D) vector of numbers, so we use the Flatten layer to convert it to a (1D) vector. The output of the Flatten layer becomes the input for the fully connected layer, which has (1000-feature) maps constructed by Average pooling, with each neuron connected to all neurons in the next layer. Following the fully connected layers, the final layer computes the softmax using equation (4), which is used to determine the probability that the input belongs to a specific class (rating from 1 to 5).

$$\sigma\mathrm{j}(\mathrm{z}) = \frac{e^{zj}}{\sum_{k=1}^{k} e^{zk}} \tag{4}$$

Where:

- Z is a vector of the output layer's inputs.
- K is the number of classes (from 1 to 5).

4. RESULTS AND DISCUSSIONS

For experiments, we used three scenarios depending on the word embedding (CAMeLBERT). The first scenario is according to the user type (couple, one person, family), while the second scenario depends on the room type ("standard double room, deluxe double or twin room, and so on"). The last scenario depends on the number of nights booked in the hotel (1, 2, 3, etc.)

The ResNet results for each scenario were evaluated using supervised machine learning. Using the KNIME Analytics (KA) platform, four different classifiers ("Support Vector Machine (SVM), Decision Tree (DT), Probabilistic Neural Network (PNN), and the K Nearest Neighbor") were used. In the first step, we created an Excel file with the preliminary results for the proposed method's various scenarios. The Excel file is then converted from XLSX to CSV format so that it can be used with the KA platform. In the final step, we separate the KA results and classify them based on the scenario name.

The first classifier (SVM) worked based on three scenarios (user type, room type, and the number of nights booked in the hotel). The evaluated results using SVM showed that the F1 score for the first scenario (user type) reached (96.37 %), with (96.48 %) for the second scenario (room type), and (97.43 %) for the third scenario (number of nights booked in the hotel). The overall accuracy for the statistical measurement methods using the SVM classifier reached (96.65 %), as shown in Table (2).

Table 2. The SVM results

Variable	User Type	Room Type	Nights	Overall
TP	478	741	342	-
FP	10	44	0	-
TN	1101	820	1255	-
FN	26	10	18	-
R	0.9484	0.9866	0.95	-
P	0.9795	0.9439	1	-
Sensitivity	0.9484	0.9866	0.95	-
Specificity	0.9909	0.949	1	-
F1	0.9637	0.9648	0.9743	-
Accuracy	-	-	-	**0.9665**

The second classifier (DT) worked based on the same scenarios, where the evaluated results showed that the F1 score for the first scenario (user type) reached (89.65 %), with (100 %) for the second scenario (room type), and (88.14 %) for the third scenario (number of nights booked in the hotel). The overall accuracy for the statistical measurement methods using the DT classifier reached (94.05 %), as shown in Table (3).

Table 3. The Decision Tree results

Variable	User Type	Room Type	Nights	Overall
TP	416	746	357	-
FP	0	0	96	-
TN	1103	869	1162	-
FN	96	0	0	-
R	0.8125	1	1	-
P	1	1	0.788	-
Sensitivity	0.8125	1	1	-
Specificity	1	1	0.9236	-
F1	0.8965	1	0.8814	-

Accuracy	-	-	-	**0.9405**

While the third classifier (PNN) worked also using the same scenarios, where the evaluated results showed that the F1 score for the first scenario (user type) reached (93.00 %), with (94.45 %) for the second scenario (room type), and (93.79 %) for the third scenario (number of nights booked in the hotel). The overall accuracy for the statistical measurement methods using the PNN classifier reached (93.86 %), as shown in Table (4).

Table 4. The PNN results

Variable	User Type	Room Type	Nights	Overall
TP	459	732	325	-
FP	27	72	0	-
TN	1087	797	1247	-
FN	42	14	43	-
R	0.9161	0.9812	0.8831	-
P	0.9444	0.9104	1	-
Sensitivity	0.9161	0.9812	0.8831	-
Specificity	0.9757	0.9171	1	-
F1	0.93	0.9445	0.9379	-
Accuracy	-	-	-	**0.9386**

Finally, the last classifier (KNN) worked also using the same scenarios, where the evaluated results showed that the F1 score for the first scenario (user type) reached (94.75 %), with (96.91 %) for the second scenario (room type), and (96.55 %) for the third scenario (number of nights booked in the hotel). The overall accuracy for the statistical measurement methods using the KNN classifier reached (96.16 %), as shown in Table (5).

Table 5. The KNN results

Variable	User Type	Room Type	Nights	Overall
TP	479	738	336	-
FP	28	34	0	-
TN	1083	830	1255	-
FN	25	13	24	-
R	0.9503	0.9826	0.9333	-
P	0.9447	0.9559	1	-
Sensitivity	0.9503	0.9826	0.9333	-
Specificity	0.9747	0.9606	1	-
F1	0.9475	0.9691	0.9655	-
Accuracy	-	-	-	**0.9616**

On the other hand, we can notice the ability of the proposed method to reduce the CSP based on the Residual Neural Network (ResNet) and the CAMeLBERT embedding. We used three scenarios for testing the results. The first scenario is according to the user type (couple, one person, family), while the second scenario depends on the room type ("standard double room, deluxe double or tin room, and so on"). The last scenario depends on the number of nights booked in the hotel (1, 2, 3, etc.). Table (6) show the results statistics for the user type scenario, while Table (7) shows the results statistics for the room type scenario, and Table (8) shows the results statistics for the number of nights scenario:

Table 6. The final results according to the "user type" scenario

Classifier	R	P	F1
SVM	0.948413	0.979508	0.96371
DT	0.8125	1	0.896552
PNN	0.916168	0.944444	0.930091
KNN	0.950397	0.944773	0.947577

Table 7. The final results according to the “room type” scenario

Classifier	R	P	F1
SVM	0.986684	0.943949	0.964844
DT	1	1	1
PNN	0.981233	0.910448	0.944516
KNN	0.98269	0.955959	0.96914

Table 8. The final results according to the “number of nights: scenario

Classifier	R	P	F1
SVM	0.95	1	0.974359
DT	1	0.788079	0.881481
PNN	0.883152	1	0.937951
KNN	0.933333	1	0.965517

According to the previous tables (6), (7), (8), the first scenario (user type) reached a low average for the (F1) score compared to other scenarios reached (0.9344), while the average of (F1) score for the third scenario (number of nights booked in the hotel) reached (0.9398). The second scenario (room type) reached a high average for the (F1) score compared to other scenarios, where the (F1) score reached (0.96962).

Finally, we compared the ResNet model results to the results of the [1] and [18] studies using the Residual Neural Network (ResNet). The authors [1] used the GNewsRec model for NR, which is a graph neural network-based method for NR that combines long-term and short-term interest modeling, whereas the authors [18] used the DAN model for NR, which is a deep attention neural network for NR that can capture the dynamic diversity of news and user interests, as well as take into account the users' click sequence information. Table (9) compares the ResNet model to the DAN model:

Table 9. Results comparing with other models

Study	Model	F1
[1]	GNewsRec	83.90
[18]	DAN	74.0
Proposed Method	ResNet	94.79

According to Table (9), we can notice the preference for the ResNet model compared to the GNewsRec model, and the DAN model, where the (F1) score for the DAN model in the [18] study reached (74.0), with (83.90) for the GNewsRec model in the [1] study. Finally, the (F1) score for the ResNet model in the proposed method reached (94.79).

5. CONCLUSION AND FUTURE WORK

In certain fields, RSs are critically important because they can generate substantial profits when used effectively or serve as an important method to differentiate from competitors. Several studies attempted to solve the CSP by recommending random objects to new users or new objects to new users (random strategy), recommending popular objects to new users or new objects to most active users (maximum expectation strategy), recommending a set of various objects to new users or a new object to a set of various users (exploratory strategy), and finally, using a non-collaborative method for the users or objects. The proposed method in this study aimed to reduce the impact of the CSP in CFM by constructing a hybrid model using the Residual Neural Networks (ResNets) model that combined each of the CFM (MO-B) and the (ME-B). The experimental results showed State-of-the-Art results with the ability to reduce the CSP

In the future, we are looking forward to using other deep learning methods using different Arabic datasets for solving the CSP. It is additionally expected that the system will be enhanced with the most recent cutting-edge technologies in this field, allowing for the generation of new scenarios by employing different techniques well-known in the broader field of AI, with the overall goal of enhancing new solutions in the NCF for the CSP.

References

[1] L. Hu et al., "Graph Neural News Recommendation with Unsupervised Preference Disentanglement," pp. 4255–4264, 2020, doi: 10.18653/v1/2020.acl-main.392.

[2] J. Wang, P. Huang, H. Zhao, Z. Zhang, B. Zhao, and D. L. Lee, "Billion-scale commodity embedding for E-commerce recommendation in alibaba," Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 839–848, 2018, doi: 10.1145/3219819.3219869.

[3] N. Sachdeva and J. McAuley, "How Useful are Reviews for Recommendation? A Critical Review and Potential Improvements," SIGIR 2020 - Proc. 43rd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., pp. 1845–1848, 2020, doi: 10.1145/3397271.3401281.

[4] Y. Lin et al., "A Survey on Reinforcement Learning for Recommender Systems," pp. 1–25, 2021, [Online]. Available: http://arxiv.org/abs/2109.10665.

[5] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," 4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc., pp. 1–10, 2016.

[6] J. Lian, F. Zhang, X. Xie, and G. Sun, "CCCFNet: A content-boosted collaborative filtering neural network for cross domain recommender systems," 26th Int. World Wide Web Conf. 2017, WWW 2017 Companion, no. 1, pp. 817–818, 2017, doi: 10.1145/3041021.3054207.

[7] S. Rendle, W. Krichene, L. Zhang, and J. Anderson, "Neural Collaborative Filtering vs. Matrix Factorization Revisited," RecSys 2020 - 14th ACM Conf. Recomm. Syst., pp. 240–248, 2020, doi: 10.1145/3383313.3412488.

[8] D. K. Chae, S. W. Kim, J. S. Kang, and J. T. Lee, "CFGAN: A generic collaborative filtering framework based on generative adversarial networks," Int. Conf. Inf. Knowl. Manag. Proc., pp. 137–146, 2018, doi: 10.1145/3269206.3271743.

[9] C. Gao, C. Huang, D. Lin, D. Jin, and Y. Li, "DPLCF: Differentially Private Local Collaborative Filtering," SIGIR 2020 - Proc. 43rd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., pp. 961–970, 2020, doi: 10.1145/3397271.3401053.

[10] G. Hu and Q. Yang, "PrivNet: Safeguarding private attributes in transfer learning for recommendation," Find. Assoc. Comput. Linguist. Find. ACL EMNLP 2020, pp. 4506–4516, 2020, doi: 10.18653/v1/2020.findings-emnlp.404.

[11] Y. Ge et al., "A Survey on Trustworthy Recommender Systems," 2022, [Online]. Available: http://arxiv.org/abs/2207.12515.

[12] Q. Guo et al., "A Survey on Knowledge Graph-Based Recommender Systems," IEEE Trans. Knowl. Data Eng., vol. 34, no. 8, pp. 3549–3568, 2022, doi: 10.1109/TKDE.2020.3028705.

[13] Q. Yao, X. Chen, J. T. Kwok, Y. Li, and C. J. Hsieh, "Efficient Neural Interaction Function Search for Collaborative Filtering," Web Conf. 2020 - Proc. World Wide Web Conf. WWW 2020, pp. 1660–1670, 2020, doi: 10.1145/3366423.3380237.

[14] D. Roy and M. Dutta, "A systematic review and research perspective on recommender systems," J. Big Data, vol. 9, no. 1, 2022, doi: 10.1186/s40537-022-00592-5.

[15] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," ACM Comput. Surv., vol. 47, no. 1, pp. 1–45, 2014, doi: 10.1145/2556270.

[16] E. Shmueli and T. Tassa, "Secure multi-party protocols for item-based collaborative filtering," RecSys 2017 - Proc. 11th ACM Conf. Recomm. Syst., pp. 89–97, 2017, doi: 10.1145/3109859.3109881.

[17] W. Liu et al., "Neural Re-ranking in Multi-stage Recommender Systems: A Review," IJCAI Int. Jt. Conf. Artif. Intell., pp. 5512–5520, 2022, doi: 10.24963/ijcai.2022/771.

[18] F. Zhu, Y. Wang, C. Chen, J. Zhou, L. Li, and G. Liu, “Cross-Domain Recommendation: Challenges, Progress, and Prospects,” IJCAI Int. Jt. Conf. Artif. Intell., pp. 4721–4728, 2021, doi: 10.24963/ijcai.2021/639.

[19] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, “Neural collaborative filtering,” 26th Int. World Wide Web Conf. WWW 2017, pp. 173–182, 2017, doi: 10.1145/3038912.3052569.

[20] A. I. Metsai, K. Karamitsios, K. Kotrotsios, P. Chatzimisios, G. Stalidis, and K. Goulianas, “Evolution of Neural Collaborative Filtering for Recommender Systems,” KST 2022 - 2022 14th Int. Conf. Knowl. Smart Technol., no. April, pp. 86–90, 2022, doi: 10.1109/KST53302.2022.9729082.

[21] J. Bobadilla, A. Gutiérrez, S. Alonso, and Á. González-Prieto, “Neural Collaborative Filtering Classification Model to Obtain Prediction Reliabilities,” Int. J. Interact. Multimed. Artif. Intell., vol. 7, no. 4, pp. 18–26, 2022, doi: 10.9781/ijimai.2021.08.010.

[22] A. Faizin and I. Surjandari, “Product recommender system using neural collaborative filtering for marketplace in indonesia,” IOP Conf. Ser. Mater. Sci. Eng., vol. 909, no. 1, 2020, doi: 10.1088/1757-899X/909/1/012072.

[23] R. A. Pugoy and H.-Y. Kao, “{BERT}-Based Neural Collaborative Filtering and Fixed-Length Contiguous Tokens Explanation,” Proc. 1st Conf. Asia-Pacific Chapter Assoc. Comput. Linguist. 10th Int. Jt. Conf. Nat. Lang. Process., pp. 143–153, 2020, [Online]. Available: https://aclanthology.org/2020.aacl-main.18.

[24] P. H. Tran, H. T. Nguyen, and N. T. Nguyen, “A Hybrid Approach for Neural Collaborative Filtering,” Proc. - 2020 7th NAFOSTED Conf. Inf. Comput. Sci. NICS 2020, pp. 368–373, 2020, doi: 10.1109/NICS51282.2020.9335910.

[25] A. Elnagar, Y.S. Khalifa, and A. Einea, “Hotel Arabic-Reviews Dataset Construction for Sentiment Analysis Applications,” Studies in Computational Intelligence, vol 740, pp. 35-52, doi=10.1007/978-3-319-67056-0_3.

[26] J. Mutinda, W. Mwangi, and G. Okeyo, “Sentiment Analysis of Text Reviews Using Lexicon-Enhanced Bert Embedding (LeBERT) Model with Convolutional Neural Network,” Appl. Sci., vol. 13, no. 3, pp. 1–16, 2023, doi: 10.3390/app13031445.

[27] A. T. Al-Taani and S. T. Ahmad, “Recognition of arabic handwritten characters using residual neural networks,” Jordanian J. Comput. Inf. Technol., vol. 7, no. 2, pp. 192–205, 2021, doi: 10.5455/jjcit.71-1615204606.