This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3002544, IEEE Access



**Deep Edu: A Deep Neural Collaborative Filtering for Educational Services Recommendation**

**FARHAN ULLAH1, BOFENG ZHANG1 (Member, IEEE), REHAN ULLAH KHAN2, TAE-SUN CHUNG3, MUHAMMAD ATTIQUE4, KHALIL KHAN5, SALIM EL KHEDIRI2, SADEEQ JAN6**

1School of Computer Engineering and Science, Shanghai University, 200444

2Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia

3Department of Computer Engineering, Ajou University, Ajou 16499, Korea

4Department of Software, Sejong University, Seoul 05006, Korea

5Department of Electrical Engineering, University of Azad Jammu and Kashmir, Muzaffarabad 13100, [Pakistan](mailto:Pakistan%3Bkhalil.khan@ajku.edu.pk)

6Department of Computer Science & IT, University of Engineering & Technology, Peshawar, Pakistan

Corresponding author: Farhan Ullah (farhan.marwat@ gmail.com), Muhammad Attique (attique@sejong.ac.kr).

This research was supported by the Basic Science Research program through the National Research Foundation of Korea funded by the Ministry of Education (2019R1F1A1058548) and (2020R1G1A1013221)

**ABSTRACT** In the modern world, people face an explosion of information and difficulty to find the right choice of theirinterest. Nowadays, people show interest in online shopping to meet their demands increasingly. For researchers and students, finding and buying the desired books from online shops is very tedious work. Recently Recommender System is an excellent tool to deal with such problems, but the Recommender System is suffering from multiple problems such as data sparsity, cold-start, and inaccuracy. To address these problems, we propose Deep Edu a novel Deep Neural Collaborative Filtering for educational services recommendation. A Deep Edu architecture consists of three parts of a Deep Neural Network model (such as input layer, a multilayered perceptron, and an output layer). The Deep Edu provides the following contributions: first, the users' identifier and books identifier features are mapped into N-dimensional dense embedding vectors, second, the Multi-Layer-Perceptron (MLP) takes the N-dimensional and non-linear features. To increase the performance of Deep Edu in all metrics, we proposed the advance Loss function. Equipped with the following, Deep Edu not only capable of learning the N-dimensional and non-linear interactions between users’ identifier and books identifier, but moreover, it also considerably mitigates the cold-start, data sparsity, and inaccuracy problem. Over significant experiments performed on real-world good books dataset, the results show that Deep Edu’s recommendation performance obviously outperforms existing Educational services recommendation methods.

**INDEX TERMS** Deep Learning, Services, Deep Neural Collaborative Filtering, Educational,Services Recommendation.

**I. INTRODUCTION**

Since we develop and move with the aim of technological improvement. Researchers are developing advanced tools and methods to meet our regular demands. There is a massive number of users on the internet to take benefit of online purchasing. In Germany and the UK, approximately 83% population is using the internet, whereas china contributes 22.3% of its population on the internet around the globe. The USA exists 78.1% of its total population, which contributes 10.2% of overall internet users in the world [1]. The growing number of online internet users also changed the lifestyle of people. With growing modern technologies, internet users have also changed at a swift pace. People are more interested in online purchasing for their daily needs. Consequently, students, researchers, and academicians prefer online purchasing of educational or academic resources

such as books because it is a tedious job and time consuming to explore libraries or book shops to buy their desired books. Most of the time, the educational community cannot find their required or desired books after spending several hours of their tight time scheduling. Collaborative Filtering (CF) technique [2], is extensively implemented in the Recommender System to solve different online recommendation problems. In the past, several CF methods have been proposed, which includes user-based [3], item-based [4], and Matrix Factorization.

The Educational Recommender System (ERS) can assist specific students (users) and books (items) based on their purchasing records and has illustrated enormous capability for improvement by generating intelligent recommendations, such as research articles, books, and other related useful information. Collaborative Filtering (CF) is among the primary methods used to build student or

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3002544, IEEE Access



academics ERS, because of its effectiveness and scalability

1. The principle of the CF method is to infer user interests from patterns of behavioral information of similar potential users. The CF method relies on user-item implicit and explicit ratings that signify how many users liked items. In this spirit, we contribute to the related domain in the form of a novel Deep Neural Collaborative Filtering for educational services recommendation. Through getting the benefits of Deep Neural Network and CF techniques, we only take into account the users’ explicit ratings of books (items) instead of any extra side details.

The majority of the typical CF techniques are centered on User-based, Item-based. However, the most successful among them is Matrix Factorization (MF) [6]. MF transform users and items to a mutual latent space and uses a latent vector features to define a user or an item [7]. Because of their efficacy, different variations of the MF technique have been introduced [8], [9]. Though, MF-based methods yet experience the data sparsity and cold-start problems, such that the efficiency of existing methods is limited. From the popularity of Deep Learning (DL) recently, integrating DL approaches is a novel step in Recommender Systems.

In such a scope, the CF-based Recommender System coupled with Deep Neural Network techniques have drawn immense attention from academia and industry. DNN techniques have already been effectively implemented and have accomplished favorable results in several research areas, such as Computer Vision (CV), speech recognition, and Natural Language Processing (NLP) [10], [11]. Research findings have shown that the artificial neural network has a strong ability to learn latent and N-dimensional characteristics from homogeneous and heterogeneous information and to achieve accurate results [12], [13]. Among them, the Embedding based (EB) method and Multi-Layer Perceptron (MLP) have been extensively applied in Recommender Systems lately. Although, they do have their own pros and cons while developing a recommendation model. In the embedding base method, a traditional and efficient method is used to recreate the input information in the predictive layer. The main concept of embedding for the recommendation system is to transform users’ preferences via transforming an input vector, and after that produce recommendations. Although, the majority of the current studies utilizing embedding-based methods primarily concentrate on the sentiment analysis of user and item text data. Regarding Multi-Layer Perceptron (MLP), it is a feed-forward neural network feed by many hidden layers that is perfect for efficient learning the N-dimensional features.

To solve the above problems. In this paper, we proposed the Deep Edu, a novel Deep Neural Collaborative Filtering for educational services recommendation. Through getting the benefit of Deep Neural Network and CF techniques, we only take into account the users’ explicit ratings of books (items) instead of any extra side details. Since we only rely on the prediction of the rating values and present

it as a regression problem. The Deep Edu architecture consists of three parts 1) the input layer 2) Multiplayer Perceptron Layer (MLP) and 3) the Output Layer. The Input Layer part transforms the low-dimensional space into N-dimensional space and fuse the N-dimensional features to the second part of the Deep Edu architecture MLP, the MLP part learns the N-dimensional features from the input layer, the MLP part is feed with Relu activation of each layer, and finally, the N-dimensional features are feed into the output layer. To evaluate on all metrics, the output layer is embedded with an advanced loss function named Huber Loss, and finally, the output layer generates the probabilistic values of the ratings. We evaluate the results with the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE). Some experiments with different hyper-parameter settings are carried out to achieve optimum results. The experimental results have been shown to explain the effects of several different factors. Moreover, we also make a comparison of the performance of the proposed model with the existing techniques. As such, we summarize the contributions as follows:

* The Deep Edu model integrates the embedding layers to Multi-Layer Perceptron with a built in order to learn the N-dimensional and non-linear interactions of users and books information.
* We implement the Huber Loss function in the Deep Edu, which has demonstrated stability, reliability, and exceptional effectiveness in all performance metrics. Hence, Deep Edu has better flexibility and scalability to exploit users and books data.
* A number of experiments have been performed on real-world dataset to analyze the efficiency of proposed model and compare it with existing methods. Experimental results show that our proposed Deep Edu not only acquires increased recommendation efficiency, but also mitigates the cold-start and data sparsity problems.

The remainder of the paper is organized as follows: Section 2 analyzes the literature review. Section 3 discusses the problem statement. Section 4 discusses the Deep Edu architecture and the details of the model techniques that we employed in our proposed model. Section 4 evaluates the proposed model and presents the comparative analysis. Section 5 concludes the paper.

1. **LITERATURE REVIEW**

In this section, we discuss state-of-the-art in recommendation approaches. As such, we divide the techniques in two categories. Firstly, the recommendation approaches based on the Collaborative Filtering and secondly, the Deep Learning paradigm.

***A. COLLABORATIVE FILTERING BASED TECHNIQUES***

The CF-based methods utilize historical data for recommending relevant items to target users. The CF-based technique has been employed extensively to recommend

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3002544, IEEE Access



items. The first use of the CF-based technique was employed in [14] for predicting the items. The CF-based technique can be classified into the following approaches the Model-based (M-B) and Memory-based (MM-B).

Consequently, the MM-B approach classified into a user-based [15], item-based method [16], the hybrid approach of the previously two approaches have been discuses in [17]. The key challenge of the CF-based method is predicting the missing values of items for the target users. The main objective is to execute similarity measurements on items or users. To calculate user-item similarity more accurately, several enhanced memory-based approaches have been proposed. In [18], the authors proposed a novel CF technique named ‘ADF.’ In ADF method, the authors combined the data smoothing and fusion of similarity techniques. The authors in [19], integrated the Covering-based clustering (CC) technique with Matrix Factorization (MF), hence introduced the Covering-based with Neighborhood-aware MF (CNMF) technique to take benefit of side data of the neighborhood for the quality of web the services prediction and recommendations. In order to maintain the resultant hybrid service is implemented correctly, the authors in [20], proposed method that integrated a Graph Search (GS) technique, and moreover, they also introduce two pre-processing techniques. In [21], the authors proposed the idea of Generalized Component Services (GCSs). The authors in [22], studied the problem of selection and composition web services by using the Genetic Algorithm (GA) and proposed a Qos-Aware selection method. In [23], the authors proposed a novel Ratio-based (RB) Collaborative Filtering service recommendations method to calculate the similarity.

To enhance the performance of similarity computation in memory-based CF techniques, most studies have tended to concentrate on textual and non-textual data, including precision, time information, and location information, respectively. In [24], prior to the similarity calculation, the authors have taken into account the user’s trust score and location information for quality of service recommendation prediction. The authors in [25], considered the predictions of personalization that utilize the previous corrupt data to estimate the likelihood of web service failure. In [26], the authors employed time data to enhance the calculation of similarity to predict the quality of service. In [27], the authors enhanced the accuracy of the quality of service performance by merging the locations information of users and services into typical similarity computations. In [28], the authors integrate the location information into Collaborative Filtering method and hence proposed a Location-Aware Collaborative Filtering (LCF), towards further enhancement of Recommender system performance. The authors in [29], integrated the Matrix Factorization (MF) technique with network map features and hence proposed network-aware Matrix factorization (NAMF). Although, while dealing with a massive scale of data integration, memory-based CF technique is unable to execute

recommendations in real-time because of the complexity of the computations associated.

Luckily, model-based Collaborative Filtering approaches resolve such issues efficiently. In [30], the authors discussed the WSPred model with time information integration to predict the quality of service. The authors in [31], implemented Factorization Machine (FM) with the integration of location data for the quality of service prediction and recommendation. Whereas contextual data leads to the similarity of CF computations, such type of computation learns the linear characteristics of the users and also only accept the low-dimensional features. While dealing with the real-world issue of data sparsity, feature learning is inadequate, therefore restricting the efficiency of the recommendation. Our proposed model employs Multi-Layer Perceptron (MLP) to address this problem in order to exploit the diverse N-dimensional and non-linear interactions between users and books.

***B. DEEP LEARNING BASED TECHNIQUES***

Deep learning models have been used many ages ago in the research field of Computer Vision, and NLP problems, etc. the authors in [32], to the best of our knowledge are the first to apply deep learning method in recommendation systems, and hence proposed the Neural Collaborative Filtering (NCF). The NCF method target the issue of poor representation of MF in low-dimensions. Consequently, in [33], the authors proposed Deep Matrix Factorization (DMF) method in recommendation system; in this deep learning approach, the authors extracted the features from the user-item matrix and integrated it into the Neural network. Moreover, they exploit the explicit and implicit feedback data into DMF. In [34], the authors proposed the convolution neural Collaborative Filtering method (ConvNCF) by integrating the Convolution Neural Network (CNN) by exploiting the local embedding dimensions and also considered the global embedding dimensions in a hierarchical approach for Recommender System. The authors in [35], integrated the CNN model with the Probabilistic Matrix Factorization method and proposed the ConvMF for quality of service recommendation. In [36], they proposed CNSR neural architecture, which jointly integrates the social media information and user-item relationship in a joint method for online social networking Recommender System. In [37], the authors discussed a novel technique based on personalized Long and Short-Term Memory (LSTM) and Matrix Factorization technique. This can take implicit feature representation of several various users and services; it redesigns the predictive method in period to handle novel information. The majority of findings primarily focus on the user ID and the item ID to accomplish fine results in the domain of Recommender System.

Lately, in [38], the authors introduced a new Deep Hybrid method for web Service Recommendation (DHSR), which employed the Multi-Layer Perceptron with the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

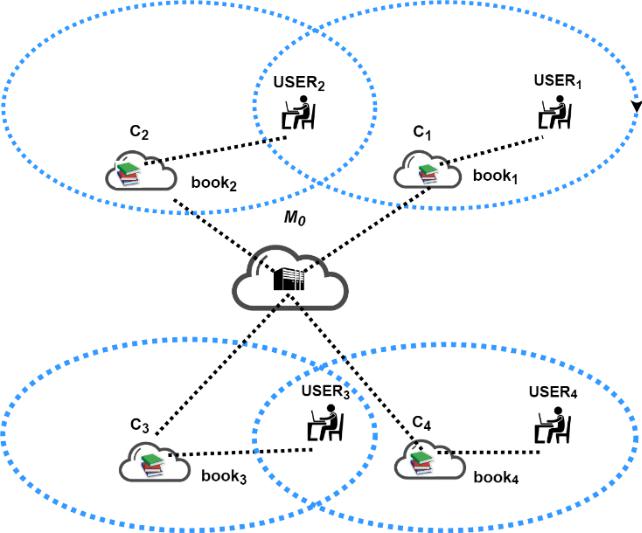
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3002544, IEEE Access



integration of textual information similarity for learning the non-linear relationship between the services and mashups creation. The authors in [39], discussed the stacked denoising autoencoders (SDAE) method to build a novel Deep Neural Network framework (DLTSR) to resolve the issue of long-tail in web services recommendation system. In [40], the authors proposed a Deep Neural Network based healthcare service recommendation method, where they integrated the trust and distrust interrelationship of the targeted users. The authors in [41], discussed the web services recommendation problem, they used the deep learning approach with the CF technique with the integration of location information. In [42], the authors used the image-based technique for services recommendation. They extract image features by applying the JPEG coefficient algorithm and also extracting the image features by applying the convolution neural network (CNN), and used the Random Forests ensemble technique as learning model.

**III. PROBLEM STATEMENT**

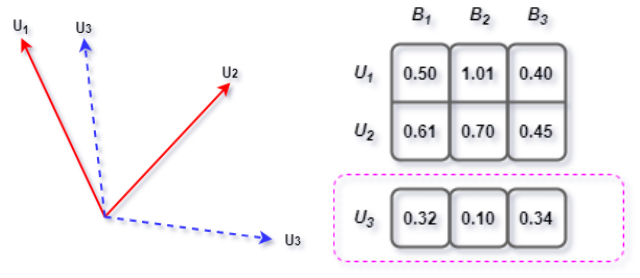
Now a day, students, researchers, and academics prefer to explore and purchase books from the ease of their places. No extra roaming to numerous shops to search the right books; no extra dealing with under-enthusiast salespeople; no additional stand at the checkout counter in long lines. The electronic commerce success has positively transformed the purchasing fashion for the academic community. However, similar to other domain problems, the world of online shopping is not still matured enough. Despite all the progressive steps taken by e-commerce companies and researchers to mitigate them, there are still problems of inaccurate items recommendations that customers have to face while shopping online.



**Figure 1. The Educational Services Recommendation scenario.**

Figure. 1 exhibits an educational services recommendation scenario. The figure comprises four users: user 1, user 2, user

3, user 4 one main server M0, and four online clouded servers: C1, C2, C3, C4. Black dotted lines denote sachet wireless transmission routes. Our objective is to recommend the related and most relevant books to the target users.



**(a)** **(b)**

**Figure 2. The example of Collaborative Filtering similarity computation.**

Figure. 2 shows how the calculation of the similarity constrains the efficiency of CF technique. The CF-based techniques use similarity computations for book recommendation such as Pearson correlation coefficient, cosine similarity, Euclidean similarity, etc. its constraint the capability of CF-based techniques inefficient features mining. Figure. 2 illustrates its constraint with the cosine similarity.

By the mentioned user-book matrix described in Figure. 2, we can get user *u*1 and *u*2’s feature vectors: *u*1= [0.50, 1.01, 0.40], *u*2= [0.61, 0.70, 0.45]. The cosine similarity between *u*1 and *u*2 is: Sim (*u*1, *u*2) = 0.54. Figure. 2(a) illustrates their geometrically interaction in a two-dimensional (2D) space. Suppose that there is a new user *u*3 = [0.32, 0.10, 0.34]. There is Sim (*u*1, *u*3) = 0.29 < Sim (*u*1, *u*2) = 0.49 < Sim (*u*2, *u*3) = 0.68. This reveals that *u*3 is more similar to *u*2 than *u*1. Furthermore, if the CF-based technique places *u*3 as the nearest user to *u*1 as illustrated in Figure. 2(a), *u*3 will be nearest to *u*1 as compare to *u*2, i.e., Sim (*u*1, *u*’3) > Sim (u2, *u*’3). It will contribute to inaccuracy computation andmiscalculation in the evaluation process of user similarity. A similar problem has also been highlighted and solved in paper

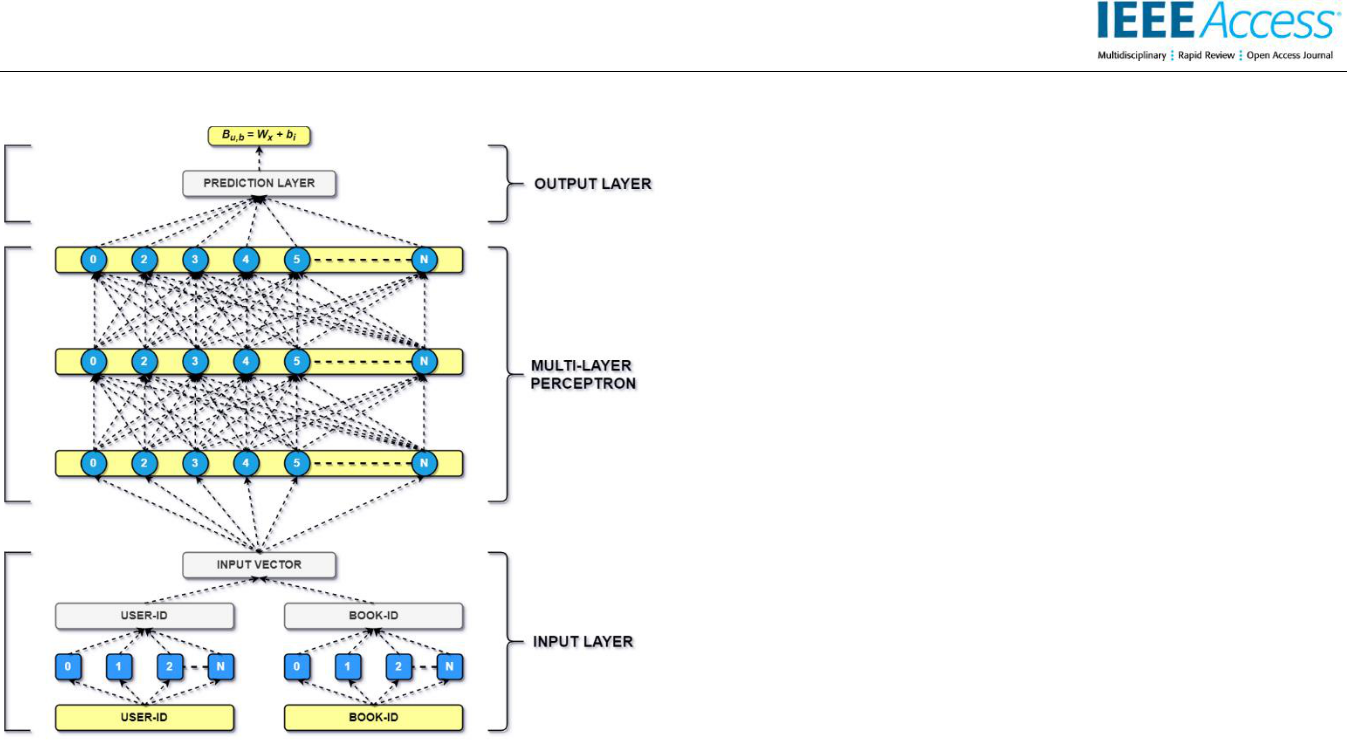
1. To address these problems, in this paper, we utilize the potential of Deep Neural Network to efficiently extract features [43].

**IV. PROPOSED METHODOLOGY**

In this part, we initially concisely present the architecture of the Deep Edu recommendation model, then characterize the parts of it. Furthermore, we include an illustration of the loss function and optimization hyper-parameters used in the proposed model.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3002544, IEEE Access



**Figure 3. The Deep Edu architecture. This architecture composes of three different parts of a Deep Neural Network. The first part is the Input Layer, the second part is a Multi-Layer Perceptron, and finally the third part is the Output Layer.**

As illustrated in Figure 3. The Deep Edu architecture is a multi-layer feedforward neural network that contains three explicit functional layers, such as the Input Layer, the Multi-Layer Perceptron or MLP, and finally the Output Layer. From the forward propagation process, the output of every single layer is employed as the input of the subsequent layer. For instance, we utilize the Input Layer to generate the input vectors needed by the Multi-Layer Perceptron and final Output Layer. The Multi-Layer perceptron is employed for the fusion part to train the acquired N-dimensional and non-linear features.

The primitive denotation of each part in Figure 3 are defined as following: The blue square and circle boxes indicate the computation node, which contains all neurons of the Deep Edu Deep Neural Network. The arrows lines indicate information run through; the light-gray colored rounded rectangle denotes the merging process. The following operational layer will be explained in a comprehensive manner below.

***A. INPUT LAYER***

The input layer is the key part of every Deep Neural Network architecture. The main applicability of the input layer is to process the given data. In order for Deep Neural Networks to process and learn information or data attributes. We feed the users’ identifier and books’ identifier into the Embedding Layer of Keras1, which can be considered as a unique, fully connected layer with bias term. In particular, embedding executes one-hot encoding on the given data to produce a zero vector with a given dimension, and the ith of

1. <https://keras.io/>

the vector is initiate to 1 [44]. Our embedding method, adopt the same process as word2vec, doc2vec, and Glove, utilizes dense vectors to indicate words or documents same as Natural Language Processing (NLP) [45],[46],[47]. Over this process, the categorical features are mapped into the N-dimensional dense embedding vectors. The method of the mapping function is illustrated in equations (1) - (2):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | = ( | |  | + | ) | (1) |  |
|  | 1 | 1 | 1 |  |  |  |
|  | = ( | | | + | ) | (2) |  |
|  | 1 | 1 |  | 1 |  |  |  |

Where and , define the user's and the books’ identifier

* respectively; the user's embedding weight matrix; the bias term initialized to zero; the activation function of the layer, and the standard identity function is designated in the paper. is the N-dimensional user’s identifier embedding vector. Similarly, is the N-dimensional books identifier embedding vector. Finally, we used the concatenating process on the following two feature vectors to get the input vector needed for the Multi-Layer Perceptron. Hence, the given process is expressed in the formula as follows:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| = ∅( | , |  |  | |  | (3) |  |
| ) = [ | | 1 | ] | |  |
| 1 | 2 | |  | |  |  |  |
|  |  |  |  | 1 |  |  |  |
| = ∅( | , | ) = |  |  |  | (4) |  |
| [ 1 | | | ] |  |
| 1 | 2 |  |  |  |  |  |  |
|  |  |  |  | 2 | |  |  |
| = ∅( , ) = | | | [ | ] | | (5) |  |
|  |  |  |  |  |  |  |  |

Where ∅ denotes the merging process, , and the embedding vector of a user and a book, and the input vector.

***B. MULTI-LAYER PERCEPTRON***

The Multi-Layer Perceptron (MLP) is applied to execute the input vector from the Input layer for acquiring the non-linear features. In this paper, a fully connected Multi-Layer Perceptron (MLP) architecture is applied to learn the N-dimensional non-linear relationship between users’ identifier and books’ identifier. For activation functions of MLP layers, we can use sigmoid, hyperbolic tangent (tanh), and Rectified Linear Unit (RELU), etc. We want to examine the following activation function*.* First, the sigmoid function constrains the ability of each neuron to be in (0,1), which may reduce the efficiency of the models; and it is obvious that it tends to experience saturation, where neurons discontinue the learning process where the output is close to either 1 or 0. Second, tanh is a more enhanced candidate and has been extensively used [12,48,49]. It can only reduce the Sigmoid problem to some level; hence it is a rescaled variant of Sigmoid (tanh(x=2) = 25-007(x) – 1).

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3002544, IEEE Access



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lastly, we choose ReLU as such, which is more biologically | | | | | | 1) LOSS FUNCTION | | | | |  |  |  |  |  |  |
| reasonable and proved unsaturated [9],[50]. Moreover, it | | | | | |  | One of the essential steps of every deep learning | | | | | | | | |  |
| enables sparse activation, is well suited to sparse data, and | | | | | | model is to select the appropriate loss function. The loss | | | | | | | | | |  |
| makes the Deep Neural Network less tenable to be over- | | | | | | functions commonly used in the conventional Recommender | | | | | | | | | |  |
| fitting. In addition, to learn more features by the multi-Layer | | | | | | Systems can be classified into two categories: pointwise and | | | | | | | | | |  |
| perceptron, the neural network architecture requires to | | | | | | pairwise. A point-wise loss function transforms the | | | | | | | | | |  |
| pursue the standard tower structure; for instance, the more | | | | | | Recommender Systems problem into a multi-classification | | | | | | | | | |  |
| neurons at the bottom levels, the lower number of neurons at | | | | | | or regression problem, whereas a pairwise function | | | | | | | | | |  |
| the top levels [32]. Lastly, we applied L2 regularization term | | | | | | transforms the Recommender Systems problem into a | | | | | | | | | |  |
| on weights to prevent the model from the overfitting | | | | | | problem of binary classification. The pointwise loss | | | | | | | | | |  |
| problem. The input vectors forward propagation process in | | | | | | functions based on the most recent applications (such as root- | | | | | | | | | |  |
| the MLP is describe as follows. | | |  |  |  | mean-square loss, log loss, etc.) can also be classified into | | | | | | | | | |  |
|  |  |  |  |  |  | regression, classification, ordinal regression, and pairwise | | | | | | | | | |  |
|  | θ1MLP = f1(W1Tx + b1) | |  |  | (6) | loss functions include BPR [51] AUC and so forth. Since the | | | | | | | | | |  |
|  |  |  |  |  |  | Deep Edu model output the value of users, books and is a | | | | | | | | | |  |
| θlMLP = fl(WlTθn−1 + bl),l = 3,4, … , n − 1 | | | | | (7) | part of regression problem class. Therefore, the multi and | | | | | | | | | |  |
|  |  |  |  |  |  | binary cross information [7],[38], is no more appropriate for | | | | | | | | | |  |
|  |  | θMLP = θn−1MLP |  |  | (8) | our model. The widely employed pointwise regression loss | | | | | | | | | |  |
|  |  |  |  |  |  | functions are square loss, absolute loss, etc. in statistics, an | | | | | | | | | |  |
|  |  |  |  |  |  | absolute loss function at a particular point(origin) is not | | | | | | | | | |  |
| Where θMLPdenotes the i-th layer of the Middle | | | | | Layer’ | arbitrary and can cause to a non-biased estimation of the | | | | | | | | | |  |
| l |  |  |  |  |  | arithmetic average. | | | | The | square | | | | loss function is highly |  |
| output, WT | the consequent weight matrix, b | |  | the bias term | |  |
| l | sensitive to outliers and simply leads to a median non-biased | | | | | | | | | |  |
| l |  |  |  |  |  |
| subsequent to the Middle layer, and θMLP the output of the | | | | | | estimate. To ensure that the Deep Edu executes efficiently | | | | | | | | | |  |
| Middle Layer. | |  |  |  |  |  |
|  |  |  |  | throughout various evaluation metrics, we select the Huber | | | | | | | | | |  |
|  |  |  |  |  |  |  |
| ***C. OUTPUT LAYER*** | |  |  |  |  | Loss function [52], | | | | which integrates the strengths of the | | | | | |  |
|  |  |  |  | previously mentioned two losses. In statistics, Huber Loss is | | | | | | | | | |  |
| The final layer or Output Layer is mainly applied to | | | | | |  |
| a loss function used in robust regression, and is less sensitive | | | | | | | | | |  |
| produce the final prediction result. Deep Edu models users’ | | | | | |  |
| to outliers in data analysis than the squared error loss2. The | | | | | | | | | |  |
| identifier and books’ identifier in one-way, and we left the | | | | | | Huber Loss function is described as following: | | | | | | | | | |  |
| two-way approach for future work. We directly concatenate | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |
| on the inputs of this one-way approach. Finally, Deep Edu | | | | | |  |  |  | ̂ |  |  |  |  |  | (10) |  |
| produces final predictions with a singular output-layer neural | | | | | | (  , − , ) | | | |  |  |  |  |  |  |  |
| 1 | | |  |  |  |  |  |  |  |  |
| network. Subsequently, the prediction result is a unique | | | | | | ̂ | 2 |  |  |  |  | ̂ |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| value, it can be accounted as a problem of regression, and the | | | | | | = {2 (  , − , ) | | | | |  | |  ,  −  , | ≤ , | | | |  |
| identity function is also choosing as the activation function. | | | | | |  |  |  |  | ̂ | − | 1 | | ), | ℎ . |  |
|  |  |  |  |  |  |  |
| The parameter initialization of this layer uses Gaussian | | | | | |  | (|  ,  −  , | | | | | 2 | |  |
|  |  |  |  |  |  |  |  |  |
| value, as exhibited in Equations (9): | | |  |  |  | Where  , , is the real value of users invoking books , | | | | | | | | | |  |
|  |  |  |  |  |  |  |
|  | ̂ |  |  |  | (9) | ̂ | | |  |  |  |  |  |  |  |  |
|  |  |  |  | ,  is the output value of user invoking books , and is | | | | | | | | | |  |
|  | ,  = ( + ) | |  |  |  | a threshold for switching and value is initialized to 1.0 in | | | | | | | | | |  |
|  |  |  |  |  |  |  |
| ̂ | denotes the output value of userinvoking | | | | | the paper. Figure 4. | | | | plots the function of the Huber Loss | | | | | |  |
| Where,  , | function. | | |  |  |  |  |  |  |  |  |
| books , and is a standard identity function that denotes | | | | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

the activation function of the final layer.

***D. DEEP EDU LEARNING PROCESS***

Through supervised learning, the learning process

of the deep learning architecture can be viewed as a process

of measuring predicted values with actual values and then

constantly optimizing the target loss function to obtain the

final optimal fit. The choice of the loss function selection and

optimizer selection has a significant overall impact on the

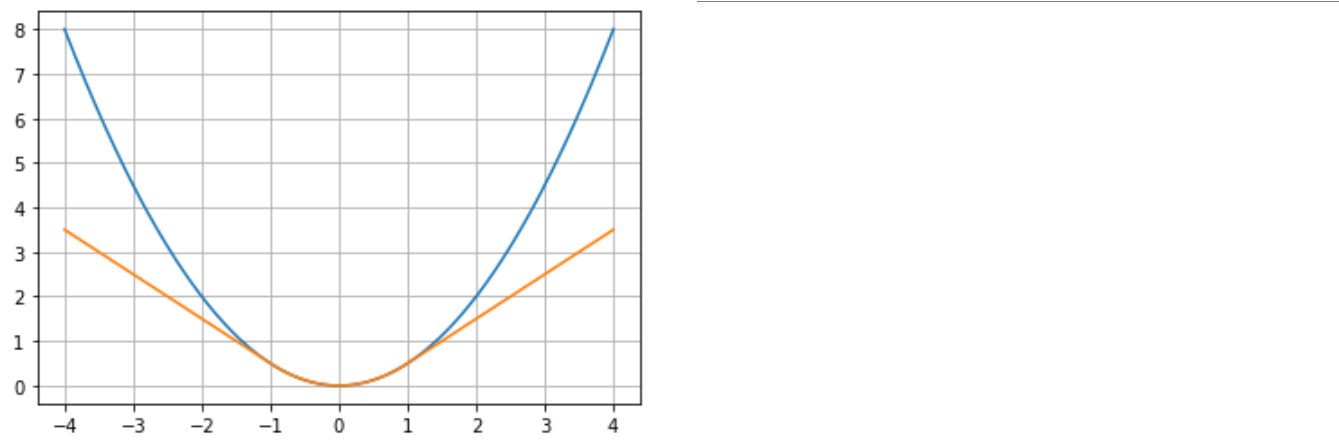
algorithm’s execution. In this part, we define the Deep Edu

model’s loss functions and optimizer.

1. <https://en.wikipedia.org/wiki/Huber_loss>

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3002544, IEEE Access



**Figure 4. Shows the significance of Huber Loss, the orange line shows Huber Loss, we set the parameter of Huber Loss** = **, and the blue line shows the squared error loss as a function of** − ( )

2) OPTIMIZER

the optimizer selection is an important part of deep learning model building. We select the mini-batch adaptive moment estimation (Adam) [53] optimization algorithm. The Adam has the superiority of high computing effectiveness, less memory specifications, and powerful interpretability, etc. moreover, Adam effectively accepts the estimation of the first moment and the second moment of the gradient to calculate the step size and efficiently integrates the strengths of the former given two optimization algorithms RMSProp

1. and AdaGrad [55]. The pseudo-code of the Deep Edu algorithm is presented below.

**Algorithm 1: Deep Edu**

**Input:** *U*ser-Book Matrix R,

Deep Neural Network topology architecture A, Learning rate l, decay ratio r, number of iteration i.

**Output:** Weight matrices and bias terms *,* *,* *,*

*, …,* ***, ,*** ***, …,*** *to 0.*

1. generate training entriesand test entries;
2. generate input features , ;
3. build neural networks according to A and Equation. (6)

(8);

1. initialize 1*,* 1*,* 1*, …,*;
2. initialize 1*,* 2*, …,*to 0;
3. **for** epoch = 1, 2, …,*i*do
4. **for** user and book indo
5. generate embedding vectors via Equation. (1)-(2)
6. generate input vector via Equation. (3)-(5);
7. generate prediction via Equation. (9);
8. **end for**
9. pass *l* and *r* to Adam;
10. update model parameters by Adam minimizing Equation. (10);

14 **for** user and book in do

1. Model evaluation performance via Equation. (11)

-(12);

* 1. **end for**
  2. **end for**

1. **EXPERIMENTS**

In this section, we performed number of experiments with the objective to validate the proposed model performance:

1. ***DATASET***

We performed experiments on the benchmark goodbooks10k, a massive-scale dataset obtained and sustained by Zygmunt Zajac [56]. This dataset is publicly

available on the website [(https://www.kaggle.com/zygmunt/goodbooks-10k),](https://www.kaggle.com/zygmunt/goodbooks-10k) which consists of 53424 unique users’ information and 10000 unique books’ information data with the inclusion of ratings of 1000000. the maximum rating is 5.0, and the minimum rating is 1.0. We do not need to preprocess the book’s dataset because it is already well filtered. The statistical analysis of the benchmark good books dataset is given below in Table 1.

**Table 1. The statistics of the Good Books’ dataset**

|  |  |  |
| --- | --- | --- |
|  | **Statistics** | **Good books 10k** |
|  |  |  |
|  | **Number of users** | **53424** |
|  | **Number of books** | **10000** |
|  | **Number of ratings** | **1000000** |
|  | **Maximum rating** | **5.0** |
|  | **Minimum rating** | **1.0** |
|  |  |  |

***B. EVALUATION MATRICES***

We use the extensively utilized RMSE and MAE metrics for evaluation of the prediction accuracy of the proposed model. The RMSE and MAE are discussed below.

1. ***RMSE***

RMSE computes the average of all squared differences between the actual and predicted rating values and then continues to calculate the square root of the results. Therefore, larger errors can significantly affect the RMSE rating value, making RMSE more valuable when they do not need more significant errors. The following is the RMSE between the actual rating values and the predicted rating values:

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3002544, IEEE Access



**TABLE 2. Performance evaluation of Deep Edu based on RMSE. The methods with the lowest RMSE score are better. “Deep Edu” represents the approach where the N-dimensional features are learned and then generates the probabilistic values.**



**TABLE 3. Performance evaluation of Deep Edu based on MAE. The methods with the lowest MAE score are better. “Deep Edu” represents the approach where the N-dimensional features are learned and then generates the probabilistic values.**



**Methods** **RMSE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **FWMF** | |  | 1.004 |  |
|  | **AVG** | |  | 0.984 |  |
|  | **ALS** | |  | 0.936 |  |
|  | **Slope one** | |  | 0.862 |  |
|  | **U-I** | |  | 0.844 |  |
|  | **NCF** | |  | 0.83101 |  |
|  | **Deep Edu** | |  | **0.82265** |  |
|  |  |  |  | (11) |  |
|  |  | ̂ | 2 |  |
|  |  |  |  |
|  | √∑ , (  , − , ) | |  |  |  |

=



Where  ,  is the actual ratings value, ̂ ,  is the predicted ratings value, and is the total number of ratings.

1. ***MAE***

MAE computes the average difference based on the number of values that need to be predicted, and calculates absolute difference between the actual values and predicted values.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ̂ | | (12) |  |
| = | ∑ , | , − , | |  |  |  |
|  | |  |  |
|  |  |  |

Where  ,  is the actual ratings value, ̂ ,  is the predictive ratings value, and is the total number of ratings.

***C. PARAMETER SETTING***

The proposed model Deep EDU is implemented in Keras (using TensorFlow and Theano as backend). The learning rate is set to 0.00146, the number of implicit feature factors (dimensions) is set to 10, the maximal number of epochs is designated to 250, the regularization parameters initialize to 0.01 and the random factor for rarefy matrix is set to 8, to initialize the model parameters, we use formula

1. to update parameter of the model. We initialize the batch size to 40000, the number of MLP to 4, and use Adam for optimizing. For a fair comparison, we use an open-source SuperLib library in GitHub for Recommender System to compare the results of the existing methods for Educational Recommender System.

***D. PERFORMANCE COMPARISON METHODS***

In this section, we illustrate the comparative results of our proposed model against the existing techniques of CF machine learning models for educational services recommendation. Our model focuses on the interrelationship between users and items.

**Methods** **MAE**

|  |  |
| --- | --- |
| **AVG** | 0.788 |
| **FWMF** | 0.744 |
| **ALS** | 0.727 |
| **U-I** | 0.668 |
| **slope one** | 0.636 |
| **NCF** | 0.62125 |
| **Deep Edu** | **0.605606** |



***Average (AVG):*** It ranked the books by taking the average

of all ratings of books. Their performance usually used as

a baseline for model evaluation. We used the AVG for the

evaluation.

***Slope one:*** This is a mainstream algorithm for booksrecommendation system. We used the same given parameters of this algorithm.

***User-Item (U-I):*** This algorithm is also the baseline methodfor user-item Collaborative Filtering. It is one of the well-known methods for recommending books for related users [57].

***Factor-wise Matrix Factorization (FWMF):*** is also a well-known matrix factorization technique for a Recommender System by adding the users and items factors and combines it with matrix factorization technique for the efficient result. We used this technique for recommending books [58].

***ALS:*** is a well-known Matrix Factorization technique forrecommendation system which uses a square loss function. We used the same parameters as given in the paper [59].

***NCF:*** is the state-of-the-art deep learning model for theRecommender Systems. We used the same parameters as given in the paper [32]

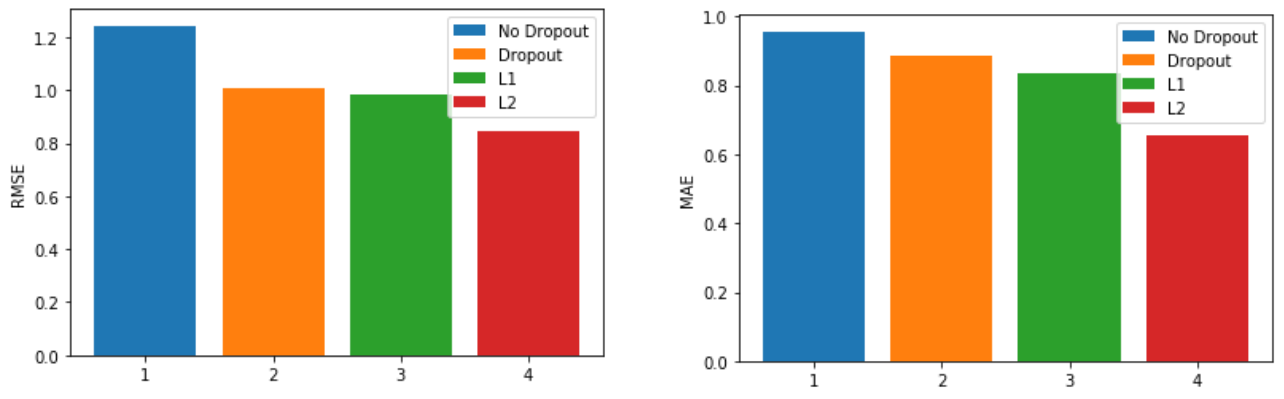
***Deep Edu:*** This is our proposed Deep Neural CollaborativeFiltering model for educational services recommendation.

***E. EXPERIMENTAL RESULTS AND ANALYSIS***

We measured our proposed model by using the two evaluation metrics: RMSE and MAE. We run the following experiments 10 times for analysis and to acquire the best final results. The experimental results are shown in Table 2 and Table 3. From Table 2 and Table 3, we can see that our proposed Deep Edu model has the lowest RMSE and MAE at both metrics. From the results point of view, the Deep Edu is considerably having an edge against the other existing techniques. Particularly, when the sparsity is significant. Our proposed Deep Edu model has apparently led in recommending similar or related items. Moreover, the

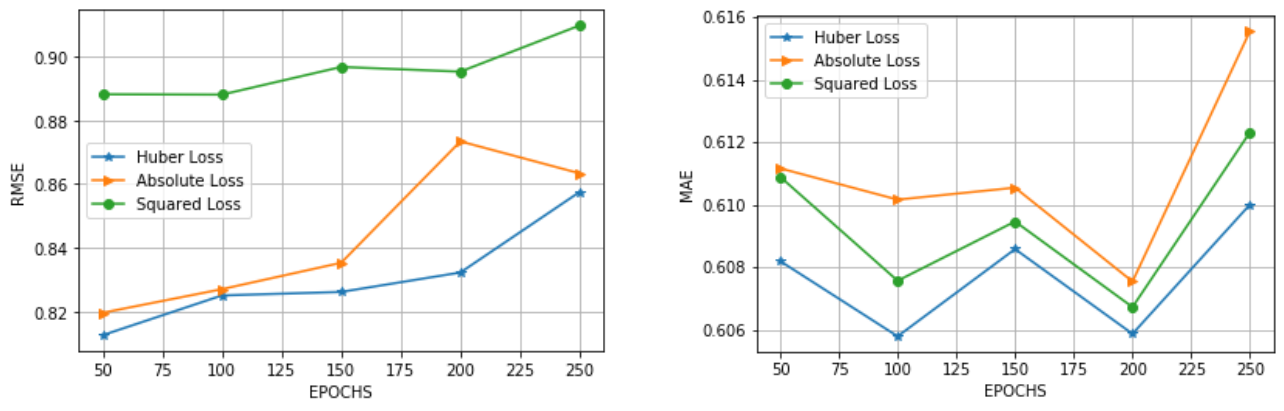
This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3002544, IEEE Access



**(a)** **(b)**

**Figure 5. Effects of different regularization terms in the Y-axis shows the RMSE and MAE of different regularization terms used in the proposed model the lowest score is better.**



**(a)** **(b)**

**Figure 6. Impact of Huber Loss the Y-axis shows the RMSE and MAE of against the comparative losses; the lowest is the better, the X-axis of (a), (b) shows the number of iterations.**

performance of Deep Edu is significantly better than that of Matrix factorization and other traditional CF techniques when the sparsity is huge. This indicates that the deep learning based techniques are practical for N-dimensional and non-linear relationship of feature learning.

***F. PERFORMANCE COMPARISON***

Table 2. illustrates the performance evaluation of Deep Edu based on RMSE. In Table 2, the FWMF acquires a very low RMSE of 1.004. the AVG has slightly improved RMSE of 0.984, the ALS with the RMSE of 0.936 is better than AVG approach. However, the performance of U-I is better than slope and other techniques. The NCF acquired a score of 0.83101. The Deep Edu shows increased performance with RMSE of 0.82 against the existing methods, as shown in Table.2.

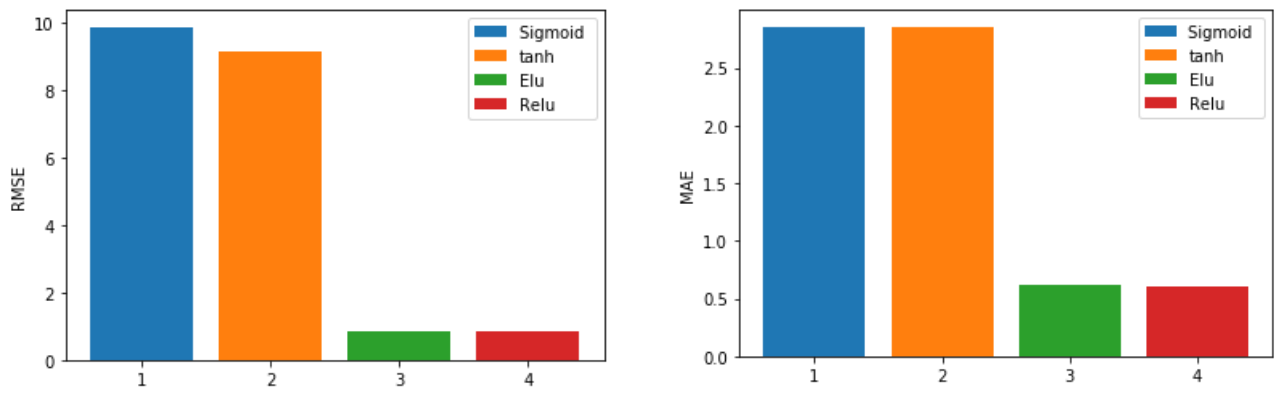
Moreover, we also compared our proposed model with the baseline and existing methods on Mean Absolute Error (MAE) result. We keep the same parameters of all baseline methods and our proposed model, as shown in the

above section. As shown in Table 3, the AVG acquired the 0.788, the FWMF obtains the 0.744, the ALS gets a slightly better score than AVG and FWMF based approach. The U-I and slope one methods with a score of 0.668, and 0.636, respectively. NCF model gains a score of 0.62125. Our proposed model Deep Edu also showed improvement in the MAE performance as compared to the existing methods with a score of 0.605606 in the 10k good books dataset, as shown in Table 3.

This shows that the typical CF technique has lower recommendation accuracy once the sparsity is huge; however, the Deep Neural Network-based method still executes efficiently. Via extensive analysis, we realize that a typical CF technique can only learn low-dimensional and linear features, therefore the learning potential is severely limited. In contrast, Deep Neural Networks can take N-dimensional non-linear features, efficiently accommodate for data sparseness, and efficiently compensate for the restriction of CF data sparsity features learning.

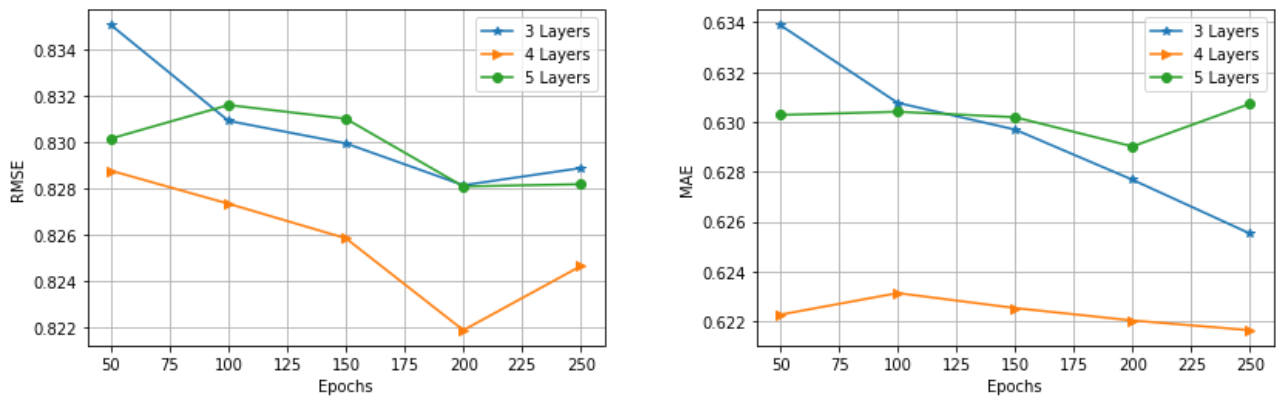
This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3002544, IEEE Access



**(a)** **(b)**

**Figure 7. Various activation functions in the Y-axis shows the RMSE and MAE of different activation functions used in the proposed model the lowest score is better.**



**(a)** **(b)**

**Figure 8. Impact of different number of layers used in the proposed model, the Y-axis shows the RMSE and MAE of different number of layers the lowest is the better, the X-axis of (a), (b) shows the number of iterations.**

We arranged the models based on relative strength, where low accuracy methods have been arranged on top and high accuracy methods at the bottom.

***G. IMPACT OF DIFFERENT REGULARIZATION TERMS***

The Regularization term has been significantly employed in the research area of the Computer Vision (CV) problems and Natural Language Processing (NLP) problems etc. Different Regularization terms have different impacts. The regularization term can prevent model from the problem of overfitting and significantly impact the predictions of the output results. The mainstream regularization terms techniques are L1 and L2. Dropout is also a different inclusive, deep learning technique [60], [61]. Dropout arbitrarily restricts certain units in the Multi-Layer Perceptron hidden layers from operating throughout the training phase, where it has proven to be effective in various deep learning models. Generally, the loss function uses one type of regularization term in the same period of execution.

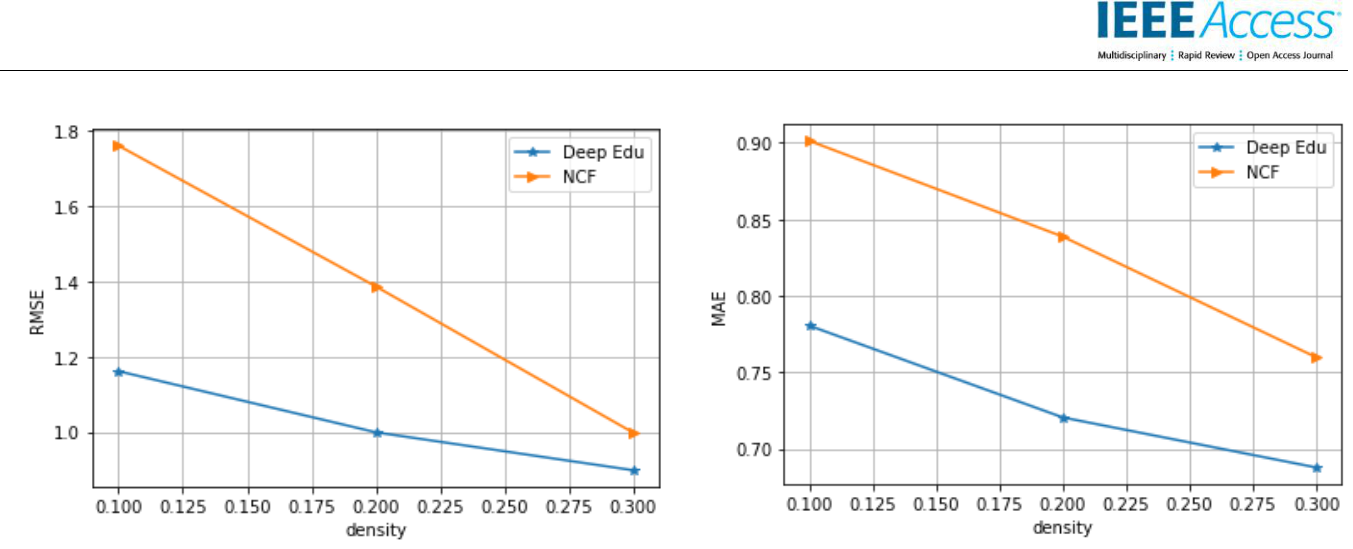
In order to identify the most effective approach of regularization, we make a comparison of the performance of the Deep Edu with various types of regularization and non-regularization techniques in this set of experiments. The outcomes are demonstrated in Figure 5. in the experimental results shown in Figure 5, we find that L2 regularization term surpasses the L1. By using the MLP without employing regularization term, the models’ face a significant over-fitting problem, which leads the model to have the worst generalization capability. We employed the Dropout in MLP for the performance analysis. Moreover, the dropout regularization is not suitable for the model training phase in simple neural networks due to the limited number of hidden units. As shown in Figure 6, the dropout results are not better than regularization term of L2, however better than the non-regularization technique.

***H. IMPACT OF HUBER LOSS***

We first introduced the Huber Loss (HL) in our proposed model by given the equation as (10). We have conducted

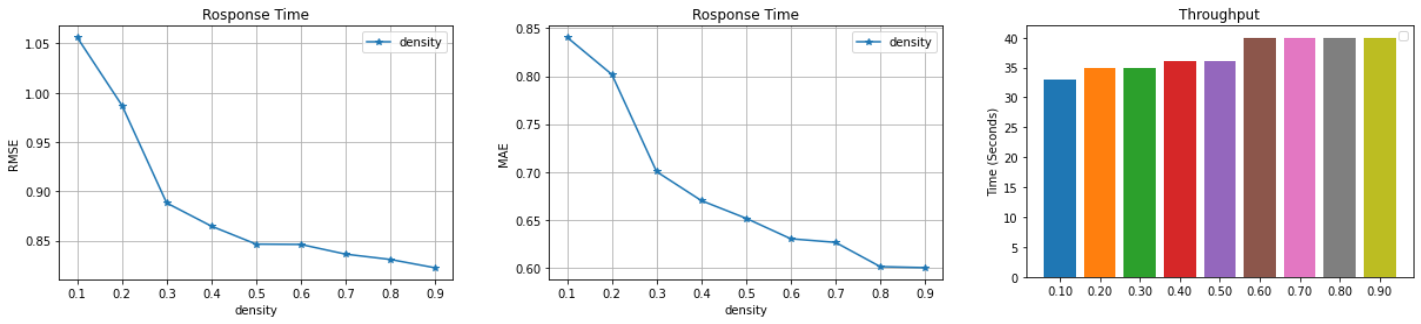
This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3002544, IEEE Access



**(a)** **(b)**

**Figure. 9. Performance comparison of different number of densities**



**(a)** **(b)** **(c)**

**Figure. 10. Scalability of proposed algorithm, Response Time and Throughput has been measured on nine different densities. Figure (a) shows the Response Time with RMSE, Figure (b) shows the Response Time with MAE, Figure (c) shows the Throughput.**

a number of experiments to validate the performance of the Huber Loss function against the following losses, such as (absolute error loss, squared error loss). The experimental results are given in Figure 6. From the comparison analysis between the squared error loss and absolute loss functions, we extract that the absolute loss function is shown good performance against the squared loss in RMSE; however, the performance of the squared error loss is better than absolute loss in MAE. From the experimental point of view, we can see the HL shown well balance between RMSE and MAE and shown good performance against the absolute loss and squared error loss functions significantly in terms of RMSE and MAE.

***I. EFFECTS OF ACTIVATION FUNCTIONS AND HYPER-PARAMETERS***

Activation function selection is an important part of the Deep Neural Network. Different activation has different effects on final prediction results; hence it is a necessary task to choose the most relevant and best fit activation for our proposed model. In this paper, we employed Relu to extract some inconsequent features from the input data and only concentrate on the related features. We employed the Relu activation in the MLP part in each layer. To show the effect of the Relu function, we compare the results of Relu with

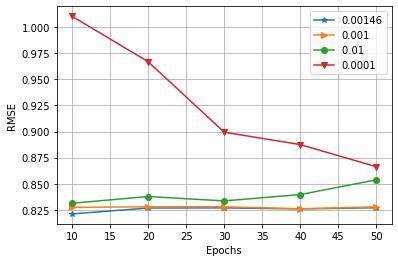
different activation functions, which include elu, tanh, Sigmoid functions, as exhibited in Figure 7. From the results perspective, we can see that Relu shows better performance against the different activation functions.

***J. IMPACT OF THE DEPTH OF NEURAL NETWORK*** Several studies recently demonstrated that Deep Neural Networks can really significantly improve the

performance of numerous artificial intelligence applications, including image processing. ResNet [62] is the latest successful neural network for computer vision problems and shown tremendous performance. A total of 100 layers has been used in ResNet Deep Learning model to get the best performance in ImageNet contest. In our proposed model, we take advantage of a tower structure pattern of DNN to enable the model to learn further non-linear features of the users’ identification and books’ identification information. Within the paradigm of the tower structure, we employed the Deep Edu network with 4 MLP layers, and each layer has 256, 128,64,32 neurons, respectively. If the network topology map formed by 32 neurons with the first layer, it can be expressed as (32). Same as if the depth of the Deep Edu second layer with 64 neurons, it can be expressed as <64,32>. Similarly, the third layer with 128 neurons

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3002544, IEEE Access



**Figure 11. The impact of different learning rates.**

(128,64,128) and fourth layer with 256 (256,128,64,32) respectively. Hence, if we denote the depth of the layers is I, then the Deep Edu Neural Network topology structure can be expressed as [ ∗  − , ∗  − , … , ]**.**

The results of the experiments are exhibited in Figure 8. We can see from Figure 8 that, as when the amount of MLP layers transform from 3 to 5, efficiency decreases exponentially at first and after which slowly increases, stating that Deep Neural Network can significantly increase the efficiency of the model. Though, as when the non-linear features are minimal, the increase in performance of further than 5 MLPs is not substantial. From the experimental analysis, we realize that even the MLP can learn N-dimensional and non-linear features complexity, hence accounting for the constraints of Collaborative Filtering for feature learning. Whereas the combination of additional layers of MLP can only fractionally boost the efficiency of the model, the performance of Deep Neural Networks must not be misjudged.

***K. SPARSITY AND COLD-START***

In the real-world scenario, the matrix of users-books is actually extremely sparse, and potential users rate a significantly limited number of books. This sparsity leads to the cold-start issue in Collaborative Filtering. In terms of making the experimental work in a realistic approach, we arbitrarily remove entries from the user-book matrix and to transform the matrix sparse at three various densities. For instance, a user-book matrix density of 0.20 represents that we arbitrarily choose 20% of the user-book entry as the training set for the model, and the 80% is utilized as the test set for the evaluation of the proposed model prediction accuracy. The sparse user-book matrix density ranges from 0.10 to 0.30.

Hence, to illustrate the effectiveness of our proposed model in mitigating the sparsity and cold start issues. We performed a number of experiments based on RMSE and MAE metrics. From Figure 9, it can be seen that Deep Edu shows better performance than the

NCF model. Specially, when the sparsity is significant, Deep Edu has shown an edge in recommendation of education services. Moreover, Deep Edu efficiency based on Deep Learning is considerably better than the NCF model, when the sparsity is huge. It shows that Deep Edu is Effective for N-dimensional and non-linear relationship features learning.

***L. SCALABILITY***

We measured two evaluation metrics for the scalability of our proposed method: Response Time and Throughput. The experiments are performed using the Core I9, running the Tesla K80 GPU. We performed experiments on nine different densities of the dataset 0.10, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9. From Figure 10 (a), (b), and (c), it can be seen that the performance of proposed model on density 0.10 with scores of 1.056 RMSE, 0.801 MAE, and Time 33 seconds. The performance of the density 0.2 scores is 0.987, 0.681, and 35 seconds. The performance of density 0.3 scores is 0.88, 0.69, and 35 seconds. The performance of the density 0.4 scores is 0. 844, 0.66, and 36 seconds. The density 0.5 scores is 0.836, 0.651, and 36 seconds. The density 0.6 scores is 0.833, 0.64, and 40 seconds. The density 0.7 scores is 0.830, 0.62, and 40 seconds. The density 0.8 scores is 0.825, 0.611, and 40 seconds. Finally, the performance of density of 0.9 is 0.822, 0.605, and 40 seconds.

Generally, the scalability of machine learning algorithms refers to the set of machine learning models which can deal with varying amount of data and parameters. This varying amount of data can either be before building the model, which is the training phase, or it can be after the model is built, which in turn is the testing or deployment phase. Once the model is built, the scalability will thus refer to the success of the model on varying test datasets and parameters. The success of the algorithms is also very subjective and can refer to many parameters. For example, the success can either by optimized memory consumption or increased recognition performance independent of the testing data sizes and types. Moreover, the optimal scalable

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3002544, IEEE Access



machine learning models should also generalize well and allow rapid computations of big datasets.

With reference to the Deep-Edu, and in the light of above definitions of scalability, our model is quite robust. As the model is built with training and testing based validation, therefore, the performance reported is the result of number of permutations that mostly represent a practical real-world scenario. In terms of the new training instances, the Deep-Edu allows a very scalable model of deep inception learning where the new instances take benefit in training of the old trained model. This not only reduces the computation times, but also increases the overall performance. For scalability in large datasets, the deep learning provides many real-time implementations. For testing new and variety of datasets, the deep learning has proved its efficacy in a large number of practical and real-world problems. Therefore, we believe that our approach takes advantage of the recent development in the machine learning models and deep learning paradigm and thus provides viable, useful, and scalable solution. From the experimental evaluation in Figure 10, we believe that the proposed model provides a scalable setup.

***M. IMPACT OF LEARNING RATES***

Choosing the appropriate learning rate is a very

important step in Deep Neural Network architecture. Hence,

the learning rate plays a vital role in building an optimized

Deep Neural Network and prevent the model from the

overfitting problem. Therefore, to prevent Deep Edu model

from overfitting and to select the appropriate learning, we

performed a number of experiments to analyze the impact of

different learning rates. From Figure 11. we can see that the

learning rate of 0.00146 has been shown better result as

compared to the standard learning rate of 0.001. However,

0.001 has shown better performance as compared to 0.01 and

0.0001. From the experimental analysis, we adopt the

0.00146 learning rate in this paper.

**IV. CONCLUSION**

In this paper, we propose Deep Edu a novel Deep Neural Collaborative Filtering method for the educational services recommendation. The proposed model can learn the N-dimension and non-Linear relationships between users and books with the help of Multi-Layer Perceptron and finally generate similarity values with the integration of an output layer between users and books. Extensive experimentation has been performed on real-world good books datasets for the evaluation of the efficiency of Deep Edu model. Compared to the existing Collaborative Filtering techniques for educational service recommendation, our method can significantly improve the accuracy of the educational services recommendation. In future work, we will integrate the side information and other contextual data in the educational service recommendation.

**REFERENCES**

1. https://www.internetworldstats.com/top20.htm.

[Online]. [Accessed:03-december-2019]

1. J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang,

“Collaborative filtering and deep learning based recommendation system for cold start items,” *Expert* *Syst. Appl.*, vol.69, pp.29-39,2017.

1. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proceedings of the 10th* *International Conference on World Wide Web,*
   1. *2001*, pp. 285-295, 2001.
2. Z. D. Zhao and M. S. Shang, “User-based collaborative-filtering recommendation algorithms on hadoop,” in *3rd International Conference on* *Knowledge Discovery and Data Mining, WKDD 2010*, pp. 478-481,2010.
3. X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F.

Zhang, “A hybrid collaborative filtering model with deep structure for recommender systems,” in *31st* *AAAI Conference on Artificial Intelligence, AAAI 2017*, 2017.

1. Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,”

*Computer (Long. Beach. Calif).* vol. 42, no.8, pp.30-27, 2009.

1. M. Vlachos, C. Dünner, R. Heckel, V. G. Vassiliadis, T. Parnell and K. Atasu, "Addressing Interpretability and Cold-Start in Matrix Factorization for Recommender Systems," in *IEEE* *Transactions on Knowledge and Data Engineering*,vol. 31, no. 7, pp. 1253-1266, 1 July 2019.
2. R. Kumar, B. K. Verma, and S. Sunder Rastogi,

“Social Popularity based SVD++ Recommender System,” *Int. J. Comput. Appl.*, vol.87, no.14, pp.33-37, 2014.

1. H. J. Xue, X. Y. Dai, J. Zhang, S. Huang, and J.

Chen, “Deep matrix factorization models for recommender systems,” in *IJCAI International Joint* *Conference on Artificial Intelligence*, p. 3203-32092017.

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton,

“ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol.60, no.6, 2017.

1. G. A. Hembury, V. V. Borovkov, J. M. Lintuluoto, and Y. Inoue, “Deep Residual Learning for Image Recognition Kaiming,” *Chem. Lett.*, vol. 32, no. 5, pp. 428–429, 2003.
2. Y. Jhamb, T. Ebesu, and Y. Fang, “Attentive

contextual denoising autoencoder for recommendation,” in *ICTIR 2018 - Proceedings of* *the 2018 ACM SIGIR International Conference on the Theory of Information Retrieval*, pp.27-34, 2018.

1. K. Christakopoulou, A. Beutel, R. Li, S. Jain, and E.
   1. Chi, “Q&R: A two-stage approach toward interactive recommendation,” in *Proceedings of the* *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.139-

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3002544, IEEE Access



148, 2018.

1. L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H.

Mei, “Personalized QoS prediction for web services via collaborative filtering,” in *Proceedings - 2007* *IEEE International Conference on Web Services, ICWS 2007*, pp.439-446, 2007.

1. Z. Tan and L. He, “An Efficient Similarity Measure

for User-Based Collaborative Filtering Recommender Systems Inspired by the Physical

Resonance Principle,” *IEEE Access*, vol. 5, pp. 27211-27228, 2017.

1. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proceedings of the 10th* *International Conference on World Wide Web,*
   1. *2001*, pp.285-295, 2001.
2. H. Ma, I. King, and M. R. Lyu, “Effective missing data prediction for collaborative filtering,” *Proc.* *30th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf.*

*Retrieval, SIGIR’07*, pp. 39-46, 2007.

1. J. Wu, L. Chen, Y. Feng, Z. Zheng, M. C. Zhou, and
   1. Wu, “Predicting quality of service for selection by neighborhood-based collaborative filtering,” *IEEE* *Trans. Syst. Man, Cybern. Part ASystems Humans*,vol. 43, no. 2, pp. 428-439, 2013.
2. Y. Zhang *et al.*, “Covering-based Web Service Quality Prediction via Neighborhood-aware Matrix

Factorization,” *IEEE Trans. Serv. Comput.*, to be published, 2019.

1. P. Wang, Z. Ding, C. Jiang, and M. Zhou, “Constraint-aware approach to web service composition,” *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 44, no. 6, pp. 770-784, 2014.
2. Q. Wu, F. Ishikawa, Q. Zhu and D. H. Shin, “QoS-aware multi-granularity service composition: modeling and optimization,” *IEEE Trans. Syst.,* *Man, Cybern.: Syst.*, vol. 46, no. 11, pp. 1565-1577,2016.
3. Z. Ding, J. Liu, Y. Sun, C. Jiang, and M. Zhou, “A transaction and QoS-aware service selection approach based on genetic algorithm,” *IEEE Trans.* *Syst. Man, Cybern. Syst.*, vol. 45, no. 7, pp. 1035-1046, 2015.
4. X. Wu, B. Cheng, and J. Chen, “Collaborative Filtering Service Recommendation Based on a

Novel Similarity Computation Method,” *IEEE* *Trans. Serv. Comput.*, vol. 10, no. 3, pp. 352-365,2017.

1. K. Chen, H. Mao, X. Shi, Y. Xu, and A. Liu, “Trust-Aware and Location-Based Collaborative Filtering for Web Service QoS Prediction,” in *Proceedings -* *International Computer Software and Applications Conference*, pp. 143-148, 2017.
2. Z. Zheng and M. R. Lyu, “Personalized reliability prediction of web services,” *ACM Trans. Softw. Eng.* *Methodol.*, vol. 22, no. 2, pp. 1-25, 2013.
3. Y. Hu, Q. Peng, X. Hu, and R. Yang, “Time Aware

and Data Sparsity Tolerant Web Service Recommendation Based on Improved Collaborative Filtering,” *IEEE Trans. Serv. Comput.*, vol. 8, no. 5,

pp. 782-794, 2015, 2015.

1. M. Tang, Y. Jiang, J. Liu, and X. Liu, “Location-aware collaborative filtering for QoS-based service recommendation,” in *Proceedings - 2012 IEEE 19th* *International Conference on Web Services, ICWS 2012*, pp. 202-209, 2012.
2. J. Liu, M. Tang, Z. Zheng, X. F. Liu, and S. Lyu, “Location-Aware and Personalized Collaborative

Filtering for Web Service Recommendation,” *IEEE* *Trans. Serv. Comput.*, vol. 9, no. 5, pp. 686-699,2016.

1. M. Tang, Z. Zheng, G. Kang, J. Liu, Y. Yang, and T.

Zhang, “Collaborative Web Service Quality

Prediction via Exploiting Matrix Factorization and

Network Map,” *IEEE Trans. Netw. Serv. Manag.*, *vol.* 13*, no.* 1*,* pp.126-137 2016.

1. Y. Zhang, Z. Zheng, and M. R. Lyu, “WSPred: A time-aware personalized QoS prediction framework for Web services,” in *Proceedings - International* *Symposium on Software Reliability Engineering, ISSRE*, pp. 210-219, 2011.
2. Y. Yang, Z. Zheng, X. Niu, M. Tang, Y. Lu, and X. Liao, “A Location-Based Factorization Machine

Model for Web Service QoS Prediction,” *IEEE* *Transactions on Services Computing*, 2018, to bepublished.

1. X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S.

Chua, “Neural collaborative filtering,” in *26th* *International World Wide Web Conference, WWW 2017*, pp. 173-182, 2017.

1. H. Xue, X. Dai, J. Zhang and S. Huang, “Deep matrix factorization models for recommender systems,” in *Proc.* *6th Int. Joi. Conf. Artifi.Intell.*, Melbourne, AU, 2017, pp.3203-3209
2. X. He, X. Du, X. Wang, F. Tian, J. Tang, and T. S. Chua, “Outer product-based neural collaborative filtering,” in *IJCAI International Joint Conference* *on Artificial Intelligence*, pp. 2227-2233, 2018.
3. D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, “Convolutional matrix factorization for document context-aware recommendation,” in *RecSys 2016 -* *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 233-240, 2016.
4. S. Zhang, L. Yao, A. Sun and Y. Tay, “Deep

Learning based Recommender System: A Survey and New Perspectives”, *ACM Comput. Surveys*, vol. 52, no. 1, 2019.

1. R. Xiong, J. Wang, Z. Li, B. Li, and P. C. K. Hung,

“Personalized LSTM Based Matrix Factorization for Online QoS Prediction,” in *Proceedings - 2018 IEEE* *International Conference on Web Services, ICWS 2018 - Part of the 2018 IEEE World Congress on Services*, pp. 34-41, 2018.

1. R. Xiong, J. Wang, N. Zhang, and Y. Ma, “Deep hybrid collaborative filtering for Web service

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3002544, IEEE Access



recommendation,” *Expert Syst. Appl.*, vol. 110, pp.

191-205, 2018.

1. B. Bai, Y. Fan, W. Tan, and J. Zhang, “DLTSR: A

Deep Learning Framework for Recommendation of Long-tail Web Services,” *IEEE Trans. Serv.* *Comput.*, vol. 13, no. 1, pp.1-1, 2017.

1. W. Yuan, C. Li, D. Guan, G. Han, and A. M.

Khattak, “Socialized healthcare service recommendation using deep learning,” *Neural* *Comput. Appl.*, vol. 30, no. 7, pp. 2071-2082, 2018.

1. Y. Zhang, C. Yin, Q. Wu, Q. He and H. Zhu, "Location-Aware Deep Collaborative Filtering for Service Recommendation," in *IEEE Transactions on* *Systems, Man, and Cybernetics: Systems*. pp. 1–12,2019
2. F. Ullah, B. Zhang and R. U. Khan, "Image-Based Service Recommendation System: A JPEG-Coefficient RFs Approach," in IEEE Access, vol. 8, pp. 3308-3318, 2020.
3. S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep learning based recommender system: A survey and new perspectives,” *ACM Computing Surveys*. vol. 52, no. 1, 2019.
4. L. Zhang, T. Luo, F. Zhang, and Y. Wu, “A

Recommendation Model Based on Deep Neural Network,” *IEEE Access*, vol. 6, pp. 9454-9463, 2018.

1. M. Ailem, B. Zhang, A. Bellet, P. Denis, and F. Sha,

“A Probabilistic Model for Joint Learning of Word Embeddings from Texts and Images,”, pp. 1478– 1487, 2019.

1. W. Sun, J. Cao, and X. Wan, “Semantic dependency parsing via book embedding,” in *ACL 2017 - 55th* *Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*. pp. 8285-838 ,2017.
2. R. Wang, A. Finch, M. Utiyama, and E. Sumita,

“Sentence embedding for neural machine translation domain adaptation,” in *ACL 2017 - 55th Annual* *Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, pp. 560-566, 2017.

1. Y. Koren, “Factor in the neighbors: Factor in the

Neighbors: Scalable and Accurate Collaborative

Filtering,” *ACM Trans. Knowl. Discov. Data*, vol. 4, no. 1, pp. 1–24, 2010.

1. R. Bell, Y. Koren, and C. Volinsky, “Modeling relationships at multiple scales to improve accuracy of large recommender systems,” 2007, pp. 95–104.
2. Y. Wu, C. DuBois, A. X. Zheng, and M. Ester,

“Collaborative denoising auto-encoders for top-N recommender systems,” in *WSDM 2016 -* *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, pp.153-162, 2016.

1. S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “BPR: Bayesian personalized

ranking from implicit feedback,” in *Proceedings of* *the 25th Conference on Uncertainty in Artificial Intelligence, UAI 2009*, pp. 452-461, 2009.

1. P. J. Huber, “Robust Estimation of a Location

Parameter,” *Ann. Math. Stat.*, vol. 35, no. 1, pp.73-101, 1964.

1. D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *3rd International*

*Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2014, pp.1-15.

1. G. Hinton and T. Tieleman, “Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning,” *Tech. Rep., Technical report*. 2012.
2. J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *J. Mach. Learn. Res.*, pp.2121-2159, 2011.
3. [https://www.kaggle.com/zygmunt/goodbooks-10k.](https://www.kaggle.com/zygmunt/goodbooks-10k) [Online]. [Accessed:03-december-2019]
4. Y. Koren, “Factor in the neighbors: Factor in the

Neighbors: Scalable and Accurate Collaborative

Filtering,” *ACM Trans. Knowl. Discov. Data*, vol. 4, no. 1, pp. 1–24, 2010.

1. Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, “Large-scale parallel collaborative filtering for the netflix prize,” in *international conference on* *Algorithmic Aspects in Information and Management*, 2008, pp. 337–348.
2. Y. Shi, M. Larson, and A. Hanjalic, “List-wise learning to rank with matrix factorization for collaborative filtering,” in *RecSys’10* *- Proceedings* *of the 4th ACM Conference on Recommender Systems*, pp.269-272,2010.
3. P. Baldi and P. Sadowski, “Understanding dropout,” in *Advances in Neural Information Processing* *Systems*, 2013.
4. Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” pp.248-255, 2009.
5. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of* *the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.770-778, 2016.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.