IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 2, 2020, accepted June 1, 2020, date of publication June 15, 2020, date of current version June 25, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3002544

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

FARHAN ULLAH[1], BOFENG ZHANG[1], (Member, IEEE), REHAN ULLAH KHAN[2], TAE-SUN CHUNG[3], MUHAMMAD ATTIQUE[4], KHALIL KHAN[5], SALIM EL KHEDIRI[2], AND SADEEQ JAN[6]

[1]School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
[2]Department of Information Technology, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia
[3]Department of Computer Engineering, Ajou University, Suwon 16499, South Korea
[4]Department of Software, Sejong University, Seoul 05006, South Korea
[5]Department of Electrical Engineering, University of Azad Jammu and Kashmir, Muzaffarabad 13100, Pakistan
[6]Department of Computer Science and IT, University of Engineering and Technology at Peshawar, Peshawar 25120, Pakistan

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

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education under Grant 2019R1F1A1058548 and Grant 2020R1G1A1013221.

ABSTRACT In the modern world, people face an explosion of information and difficulty to find the right choice of their interest. 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],

The associate editor coordinating the review of this manuscript and approving it for publication was Yunjie Yang.

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 academics ERS, because of its effectiveness and scalability [5]. 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.

II. 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 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 [3], the hybrid approach of the previously two approaches have been discuses in [16]. 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 [17], 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 [18], 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 [19], proposed method that integrated a Graph Search (GS) technique, and moreover, they also introduce two pre-processing techniques. In [20], the authors proposed the idea of Generalized Component Services (GCSs). The authors in [21], studied the problem of selection and composition web services by using the Genetic Algorithm (GA) and proposed a Qos-Aware selection method. In [22], 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 [23], 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 [24], considered the predictions of personalization that utilize the previous corrupt data to estimate the likelihood of web service failure. In [25], the authors employed time data to enhance the calculation of similarity to predict the quality of service. In [26], 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 [27], 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 [28], 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 [29], the authors discussed the WSPred model with time information integration to predict the quality of service. The authors in [30], 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 [31], 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 [9], 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 [32], 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 [33], integrated the CNN model with the Probabilistic Matrix Factorization method and proposed the ConvMF for quality of service recommendation. In [34], 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 [35], 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 [36], the authors introduced a new Deep Hybrid method for web Service Recommendation (DHSR), which employed the Multi-Layer Perceptron with the integration of textual information similarity for learning the non-linear relationship between the services and mashups creation. The authors in [37], 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 [38], 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 [39], discussed the web services recommendation problem, they used the deep learning approach with the CF technique with the integration of location information. In [40], 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 exhibits an educational services recommendation scenario. The figure comprises four users: user 1, user 2, user 3, user 4 one main server M_0, 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.

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 $u1$ and $u2$'s feature vectors: $u1= [0.50, 1.01, 0.40]$, $u2 = [0.61, 0.70, 0.45]$. The cosine similarity between $u1$ and $u2$ is: Sim $(u1, u2) = 0.54$. Figure. 2(a) illustrates their geometrically interaction in a two-dimensional (2D) space. Suppose that there is a new user $u3 = [0.32, 0.10, 0.34]$. There is Sim $(u1, u3) = 0.29 <$ Sim $(u1, u2) = 0.49 <$ Sim $(u2, u3) = 0.68$. This reveals that $u3$ is more similar to $u2$ than $u1$. Furthermore, if the CF-based technique places $u3$ as the nearest user to $u1$ as illustrated in Figure. 2(a), $u3$ will be nearest to $u1$ as compare to $u2$, i.e., Sim $(u1, u'3) >$ Sim $(u2, u'3)$. It will contribute to inaccuracy computation and miscalculation in the evaluation process of user similarity. A similar problem has also been highlighted and solved in paper [31]. To address these problems, in this paper, we utilize the potential of Deep Neural Network to efficiently extract features [34].

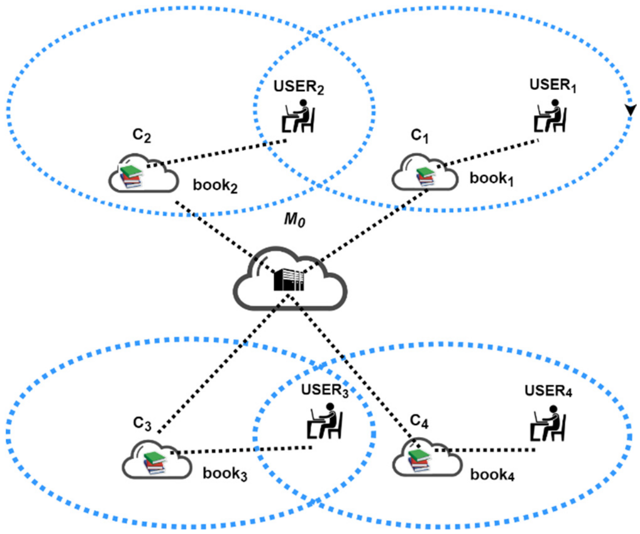


FIGURE 1. The educational services recommendation scenario.

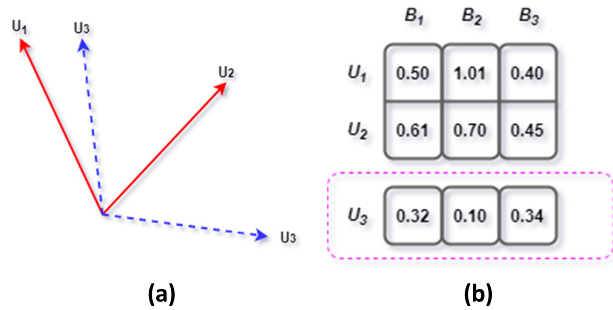


FIGURE 2. The example of collaborative filtering similarity computation.

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.

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

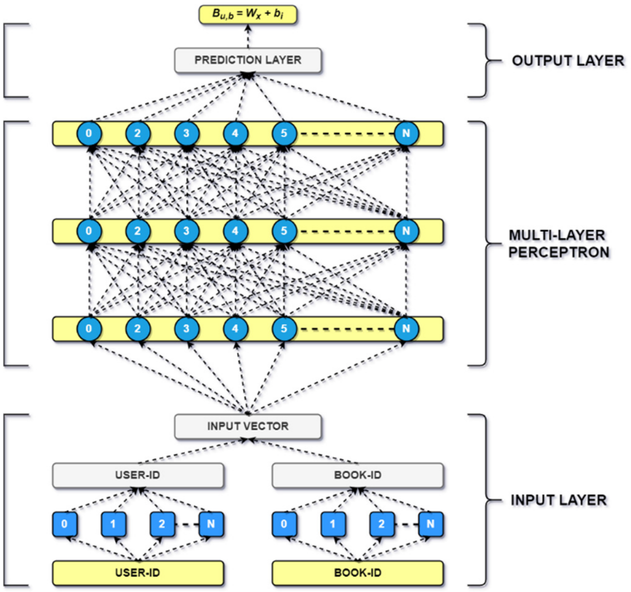


FIGURE 3. The Deep Edu architecture. This architecture composes of three different parts of a deep neural network. The first part is the input layer, the second part is a multi-layer perceptron, and finally the third part is the output layer.

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 Keras,[1] 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 the vector is initiate to 1 [41]. 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) [42]–[44]. 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):

$$I_u^n = f_1(P_1^T i_u + b_1) \tag{1}$$
$$I_b^n = f_1(R_1^T i_b + b_1) \tag{2}$$

where i_u and i_b, define the user's and the books' identifier, respectively; w_l the user's embedding weight matrix; b_l the bias term initialized to zero; f_l the activation function of the layer, and the standard identity function is designated in the paper. I_u^n is the N-dimensional user's identifier embedding vector. Similarly, I_b^n 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:

$$N = \emptyset\left(I_{u1}^k, I_{u2}^k\right) = \begin{bmatrix} I_{u1}^k \\ I_{u1}^k \end{bmatrix} \tag{3}$$
$$M = \emptyset\left(I_{b1}^k, I_{b2}^k\right) = \begin{bmatrix} I_{b1}^k \\ I_{b2}^k \end{bmatrix} \tag{4}$$
$$V = \emptyset(N, M) = \begin{bmatrix} N \\ M \end{bmatrix} \tag{5}$$

where Ø denotes the merging process, N, and M the embedding vector of a user and a book, and V 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], [45], [46]. 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). Lastly, we choose ReLU as such, which is more biologically reasonable and proved unsaturated [9], [47]. Moreover, it enables sparse activation, is well suited to sparse data, and makes the Deep Neural Network less tenable to be overfitting. In addition, to learn more features by the multi-Layer perceptron, the neural network architecture requires to pursue the standard tower structure; for instance, the more neurons at the bottom levels, the lower number of neurons at the top levels [31]. Lastly, we applied L2 regularization term on weights to prevent the model from the overfitting problem. The input vectors forward propagation process in the MLP is describe as follows:

$$\theta_1^{\mathrm{MLP}} = \mathrm{f}_1(\mathrm{W}_1^{\mathrm{T}}\mathrm{x}+\mathrm{b}_1) \tag{6}$$

[1]https://keras.io/

$$\theta_l^{MLP} = f_l\left(W_l^T \theta_{n-1} + b_l\right), \quad l = 3, 4, \ldots, n-1 \quad (7)$$
$$\theta^{MLP} = \theta_{n-1}^{MLP} \quad (8)$$

where θ_l^{MLP} denotes the i-th layer of the Middle Layer' output, W_l^T the consequent weight matrix, b_l the bias term subsequent to the Middle layer, and θ^{MLP} the output of the Middle Layer.

C. OUTPUT LAYER

The final layer or Output Layer is mainly applied to produce the final prediction result. Deep Edu models users' identifier and books' identifier in one-way, and we left the two-way approach for future work. We directly concatenate on the inputs of this one-way approach. Finally, Deep Edu produces final predictions with a singular output-layer neural network. Subsequently, the prediction result is a unique value, it can be accounted as a problem of regression, and the identity function is also choosing as the activation function. The parameter initialization of this layer uses Gaussian value, as exhibited in Equations (9):

$$\hat{B}_{u,b} = f_l(W_l^T x + b_l) \quad (9)$$

where, $\hat{B}_{u,b}$ denotes the output value of user u invoking books b, and f_l 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) LOSS FUNCTION

One of the essential steps of every deep learning model is to select the appropriate loss function. The loss functions commonly used in the conventional Recommender Systems can be classified into two categories: pointwise and pairwise. A point-wise loss function transforms the Recommender Systems problem into a multi-classification or regression problem, whereas a pairwise function transforms the Recommender Systems problem into a problem of binary classification. The pointwise loss functions based on the most recent applications (such as root-mean-square loss, log loss, etc.) can also be classified into regression, classification, ordinal regression, and pairwise loss functions include BPR [48] AUC and so forth. Since the Deep Edu model output the value of users, books and is a part of regression problem class. Therefore, the multi and binary cross information [7], [36], is no more appropriate for 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 arbitrary and can cause to a non-biased estimation of the arithmetic average. The square loss function is highly sensitive to outliers and simply leads to a median non-biased estimate. To ensure that the Deep Edu executes efficiently throughout various evaluation metrics, we select the Huber Loss function [49], which integrates the strengths of the previously mentioned two losses. In statistics, Huber Loss is a loss function used in robust regression, and is less sensitive to outliers in data analysis than the squared error loss.[2] The Huber Loss function is described as following:

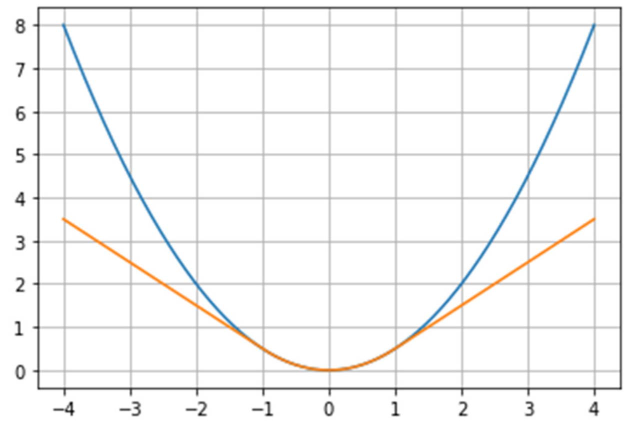


FIGURE 4. Shows the significance of Huber Loss, the orange line shows Huber Loss, we set the parameter of Huber Loss $\delta = 1$, and the blue line shows the squared error loss as a function of $y - f(x)$.

$$L_\delta\left(B_{u,b} - \hat{B}_{u,b}\right) = \begin{cases} \frac{1}{2}(B_{u,b} - \hat{B}_{u,b})^2 & \textit{for } \left|B_{u,b} - \hat{B}_{u,b}\right| \leq \delta, \\ \delta\left(\left|B_{u,b} - \hat{B}_{u,b}\right| - \frac{1}{2}\delta\right), & \textit{otherwise}. \end{cases} \quad (10)$$

where $B_{u,b}$, is the real value of users u invoking books b, $\hat{B}_{u,b}$ is the output value of user u invoking books b, and δ is a threshold for switching and δ value is initialized to 1.0 in the paper. Figure 4. plots the function of the Huber Loss function.

2) OPTIMIZER

the optimizer selection is an important part of deep learning model building. We select the mini-batch adaptive moment estimation (Adam) [50] 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 [51] and AdaGrad [52]. The pseudo-code of the Deep Edu algorithm is presented below.

[2] https://en.wikipedia.org/wiki/Huber_loss

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

V. EXPERIMENTS

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

A. DATASET

We performed experiments on the benchmark goodbooks10k, a massive-scale dataset obtained and sustained by Zygmunt Zajac [53]. This dataset is publicly available on the website (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.

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:

$$RMSE = \frac{\sqrt{\sum_{u,b} (B_{u,b} - \hat{B}_{u,b})^2}}{N} \tag{11}$$

where $B_{u,b}$ is the actual ratings value, $\hat{B}_{u,b}$ is the predicted ratings value, and N is the total number of ratings.

2) 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.

$$MAE = \frac{\sum_{u,b} \left| B_{u,b} - \hat{B}_{u,b} \right|}{N} \tag{12}$$

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 $\mathbf{P_1}$, $\mathbf{R_1}$, $\mathbf{W_1}$, $\mathbf{W_2}, \ldots, \boldsymbol{W_n}, \boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_n}$ *to* 0.

1. generate training entries R_{train} and test entries R_{test};
2. generate input features i_u, i_b;
3. build neural networks according to A and Equation. (6)- (8);
4. initialize $P_1, R_1, W_1, \ldots, W_n$;
5. initialize $b_1, b_2, \ldots, b_n$ to 0;
6. **for** epoch $= 1, 2, \ldots, i$ do
7. **for** user and book in R_{train} do
8. generate embedding vectors via Equation. (1)-(2);
9. generate input vector via Equation. (3)-(5);
10. generate prediction via Equation. (9);
11. **end for**
12. pass l and r to Adam;
13. update model parameters by Adam minimizing Equation. (10);

1) [14] **for** user and book in R_{test} do

15. Model evaluation performance via Equation. (11) -(12);

2) [3] **end for**

17. **end for**

---

where $B_{u,b}$ is the actual ratings value, $\hat{B}_{u,b}$ is the predictive ratings value, and N 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 (10) 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.

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.

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.

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**

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	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**

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

User-Item (U-I): This algorithm is also the baseline method for user-item Collaborative Filtering. It is one of the well-known methods for recommending books for related users [45].

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 [54].

ALS: is a well-known Matrix Factorization technique for recommendation system which uses a square loss function. We used the same parameters as given in the paper [55].

NCF: is the state-of-the-art deep learning model for the Recommender Systems. We used the same parameters as given in the paper [31]

Deep Edu: This is our proposed Deep Neural Collaborative Filtering 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 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.

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

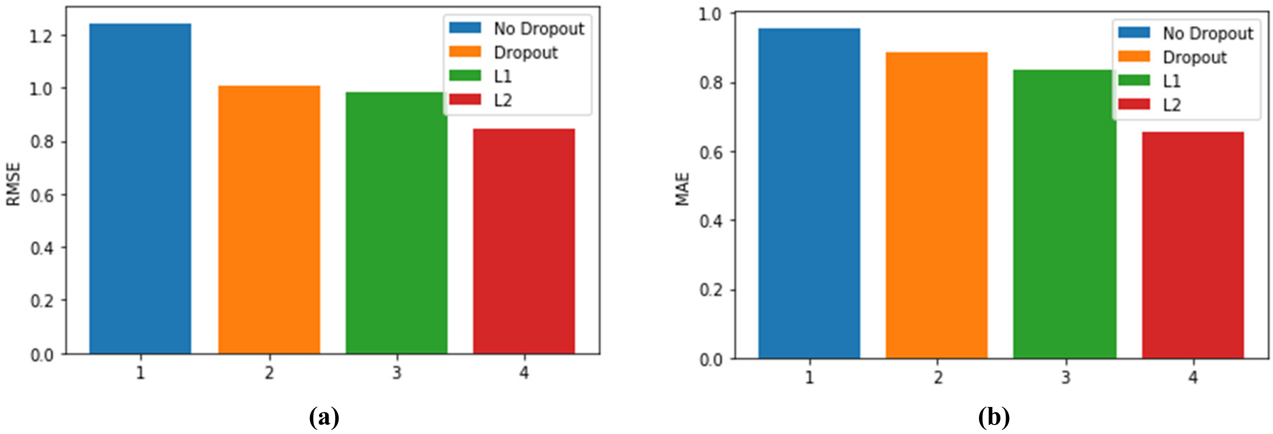


FIGURE 5. Effects of different regularization terms in the Y-axis shows the RMSE and MAE of different regularization terms used in the proposed model the lowest score is better.

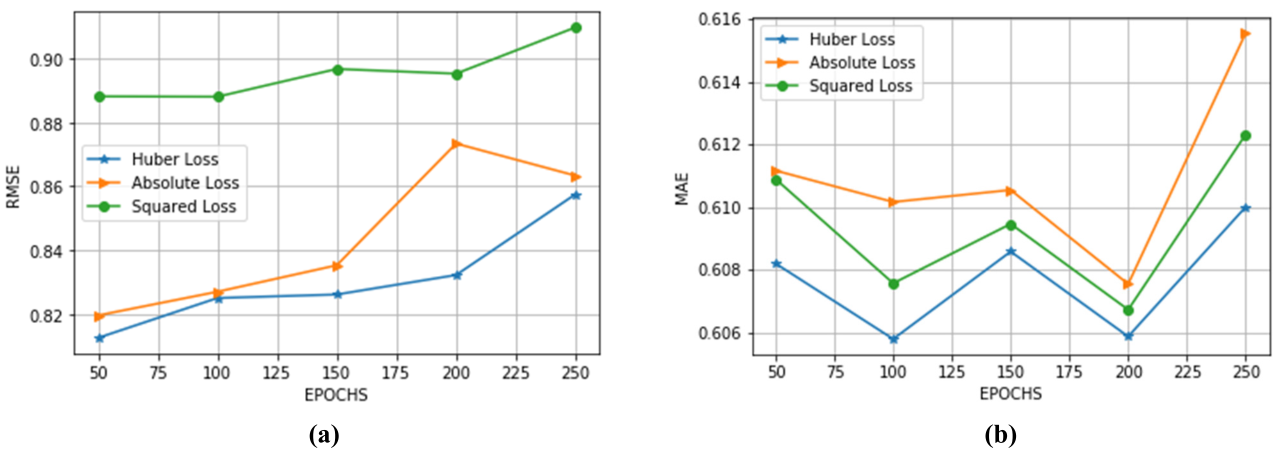
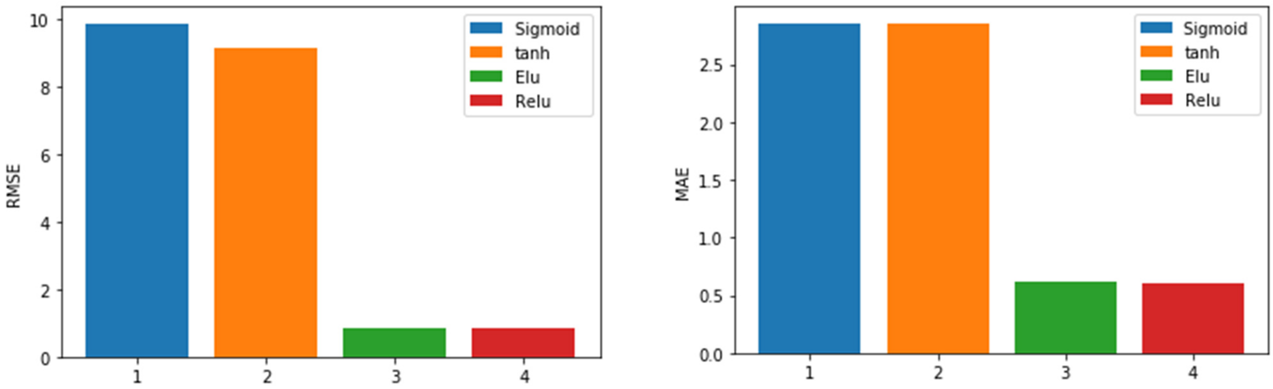


FIGURE 6. Impact of Huber Loss the Y-axis shows the RMSE and MAE of against the comparative losses; the lowest is the better, the X-axis of (a), (b) shows the number of iterations.

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 [56], [57]. 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 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



(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.

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

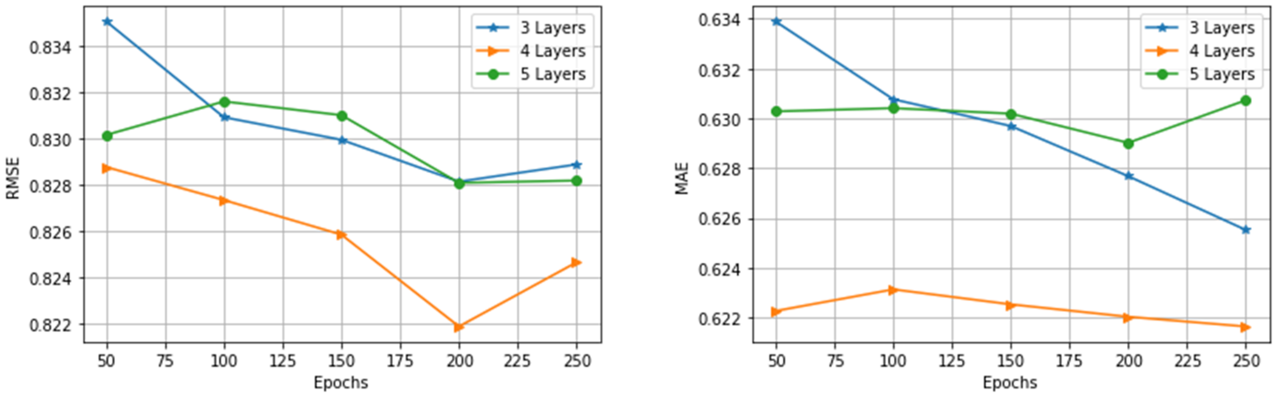
Several studies recently demonstrated that Deep Neural Networks can really significantly improve the performance of numerous artificial intelligence applications, including image processing. ResNet [58] is the latest successful neural network for computer vision problems and shown tremendous performance. 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 (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 $[\mathbf{2^4} * \mathbf{2^{i-1}}, \mathbf{2^3} * \mathbf{2^{i-2}}, \ldots, \mathbf{2^4}]$.

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

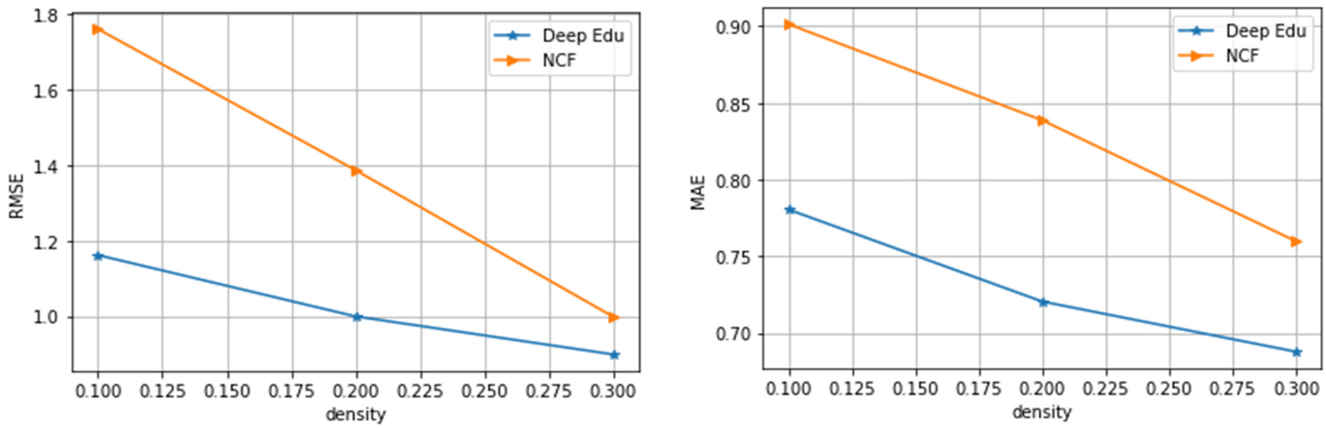
In the real-world scenario, the matrix of users-books is actually extremely sparse, and potential users rate a significantly limited number of books. This sparsity leads to the cold-start issue in Collaborative Filtering. In terms of making the experimental work in a realistic approach, we arbitrarily 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



(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.



(a) (b)

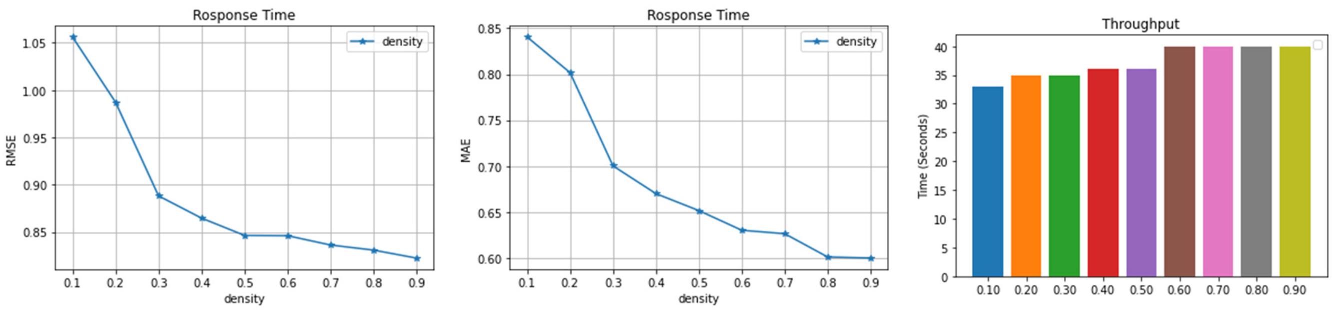
FIGURE 9. Performance comparison of different number of densities.

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



(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.**

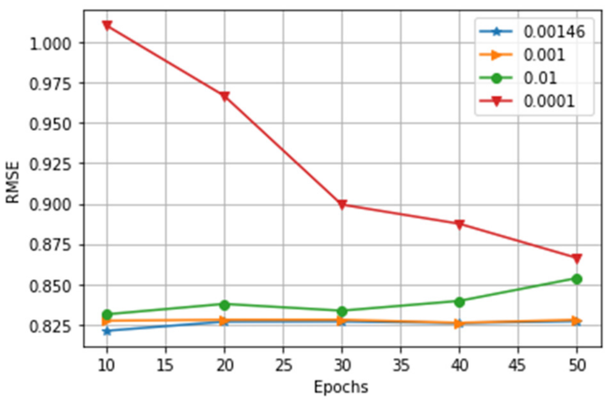


FIGURE 11. **The impact of different learning rates.**

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.

VI. 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] *Internet World Stats—Usage and Population Statistics*. Accessed: Dec. 3, 2019. [Online]. Available: https://www.internetworldstats.com/top20.htm

[2] 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, Mar. 2017.

[3] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web (WWW)*, 2001, pp. 285–295.

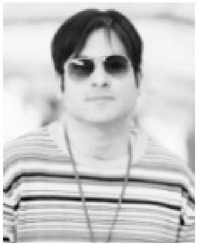
[4] Z.-D. Zhao and M.-S. Shang, "User-based collaborative-filtering recommendation algorithms on Hadoop," in *Proc. 3rd Int. Conf. Knowl. Discovery Data Mining*, Jan. 2010, pp. 478–481.

[5] 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 *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1309–1315.

[6] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[7] M. Vlachos, C. Dunner, R. Heckel, V. G. Vassiliadis, T. Parnell, and K. Atasu, "Addressing interpretability and cold-start in matrix factorization for recommender systems," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 7, pp. 1253–1266, Jul. 2019.
[8] 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, Feb. 2014.
[9] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3203–3209.
[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[11] 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.
[12] Y. Jhamb, T. Ebesu, and Y. Fang, "Attentive contextual denoising autoencoder for recommendation," in *Proc. ACM SIGIR Int. Conf. Theory Inf. Retr.*, Sep. 2018, pp. 27–34.
[13] K. Christakopoulou, A. Beutel, R. Li, S. Jain, and E. H. Chi, "Q&R: A two-stage approach toward interactive recommendation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 139–148.
[14] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS prediction forWeb services via collaborative filtering," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jul. 2007, pp. 439–446.
[15] 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.
[16] H. Ma, I. King, and M. R. Lyu, "Effective missing data prediction for collaborative filtering," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2007, pp. 39–46.
[17] J. Wu, L. Chen, Y. Feng, Z. Zheng, M. C. Zhou, and Z. Wu, "Predicting quality of service for selection by neighborhood-based collaborative filtering," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 43, no. 2, pp. 428–439, Mar. 2013.
[18] Y. Zhang, K. Wang, Q. He, F. Chen, S. Deng, Z. Zheng, and Y. Yang, "Covering-based Web service quality prediction via neighborhood-aware matrix factorization," *IEEE Trans. Services Comput.*, early access, Jan. 9, 2019, doi: 10.1109/TSC.2019.2891517.
[19] 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, Jun. 2014.
[20] Q. Wu, F. Ishikawa, Q. Zhu, and D.-H. Shin, "QoS-aware multigranularity service composition: Modeling and optimization," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 11, pp. 1565–1577, Nov. 2016.
[21] 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, Jul. 2015.
[22] X. Wu, B. Cheng, and J. Chen, "Collaborative filtering service recommendation based on a novel similarity computation method," *IEEE Trans. Services Comput.*, vol. 10, no. 3, pp. 352–365, May 2017.
[23] K. Chen, H. Mao, X. Shi, Y. Xu, and A. Liu, "Trust-aware and location-based collaborative filtering for Web service QoS prediction," in *Proc. IEEE 41st Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2017, pp. 143–148.
[24] Z. Zheng and M. R. Lyu, "Personalized reliability prediction of Web services," *ACM Trans. Softw. Eng. Methodol.*, vol. 22, no. 2, pp. 1–25, Mar. 2013.
[25] 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. Services Comput.*, vol. 8, no. 5, pp. 782–794, Sep. 2015.
[26] M. Tang, Y. Jiang, J. Liu, and X. Liu, "Location-aware collaborative filtering for QoS-based service recommendation," in *Proc. IEEE 19th Int. Conf. Web Services*, Jun. 2012, pp. 202–209.
[27] J. Liu, M. Tang, Z. Zheng, X. Liu, and S. Lyu, "Location-aware and personalized collaborative filtering for Web service recommendation," *IEEE Trans. Services Comput.*, vol. 9, no. 5, pp. 686–699, Sep. 2016.
[28] 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. Service Manage.*, vol. 13, no. 1, pp. 126–137, Mar. 2016.
[29] Y. Zhang, Z. Zheng, and M. R. Lyu, "WSPred: A time-aware personalized QoS prediction framework for Web services," in *Proc. IEEE 22nd Int. Symp. Softw. Rel. Eng.*, Nov. 2011, pp. 210–219.
[30] 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 Trans. Services Comput.*, early access, Oct. 17, 2019, doi: 10.1109/TSC.2018.2876532.
[31] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.
[32] X. He, X. Du, X. Wang, F. Tian, J. Tang, and T.-S. Chua, "Outer product-based neural collaborative filtering," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2227–2233.
[33] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 233–240.
[34] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, p. 5, 2019.
[35] R. Xiong, J. Wang, Z. Li, B. Li, and P. C. K. Hung, "Personalized LSTM based matrix factorization for online QoS prediction," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jul. 2018, pp. 34–41.
[36] R. Xiong, J. Wang, N. Zhang, and Y. Ma, "Deep hybrid collaborative filtering for Web service recommendation," *Expert Syst. Appl.*, vol. 110, pp. 191–205, Nov. 2018.
[37] B. Bai, Y. Fan, W. Tan, and J. Zhang, "DLTSR: A deep learning framework for recommendation of long-tail Web services," *IEEE Trans. Services Comput.*, vol. 13, no. 1, pp. 73–85, Jan./Feb. 2020.
[38] 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, Oct. 2018.
[39] Y. Zhang, C. Yin, Q. Wu, Q. He, and H. Zhu, "Location-aware deep collaborative filtering for service recommendation," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Aug. 19, 2020, doi: 10.1109/TSMC.2019.2931723.
[40] F. Ullah, B. Zhang, and R. U. Khan, "Image-based service recommendation system: A JPEG-coefficient RFs approach," *IEEE Access*, vol. 8, pp. 3308–3318, 2020.
[41] 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.
[42] M. Ailem, B. Zhang, A. Bellet, P. Denis, and F. Sha, "A probabilistic model for joint learning of word embeddings from texts and images," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2019, pp. 1478–1487.
[43] W. Sun, J. Cao, and X. Wan, "Semantic dependency parsing via book embedding," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 838–8285.
[44] R. Wang, A. Finch, M. Utiyama, and E. Sumita, "Sentence embedding for neural machine translation domain adaptation," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2017, pp. 560–566.
[45] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Trans. Knowl. Discovery Data*, vol. 4, no. 1, pp. 1–24, 2010.
[46] R. Bell, Y. Koren, and C. Volinsky, "Modeling relationships at multiple scales to improve accuracy of large recommender systems," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 95–104.
[47] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for Top-N recommender systems," in *Proc. 9th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2016, pp. 153–162.
[48] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell. (UAI)*, 2009, pp. 452–461.
[49] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, Mar. 1964.
[50] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, vol. 2014, pp. 1–15.
[51] T. Tieleman and G. Hinton, "RMSProp: Divide the gradient by a running average of its recent magnitude," Neural Netw. Mach. Learn., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2012.
[52] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011.
[53] *Goodbooks-10k: A New Dataset for Book Recommendations*. Accessed: Dec. 3, 2019. [Online]. Available: https://www.kaggle.com/zygmunt/goodbooks-10k

[54] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, "Large-scale parallel collaborative filtering for the netflix prize," in *Proc. Int. Conf. Algorithmic Appl. Manage.*, 2008, pp. 337–348.
[55] Y. Shi, M. Larson, and A. Hanjalic, "List-wise learning to rank with matrix factorization for collaborative filtering," in *Proc. 4th ACM Conf. Recommender Syst. (RecSys)*, 2010, pp. 269–272.
[56] P. Baldi and P. Sadowski, "Understanding dropout," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2814–2822.
[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.



FARHAN ULLAH received the B.Sc. degree in computer science from the University of Peshawar and the M.C.S. degree from the Sarhad University of Information Science and Technology Peshawar. He is currently pursuing the Ph.D. degree in computer science with Shanghai University, Shanghai, China. His research interests include machine learning, recommender systems, and computer vision.

BOFENG ZHANG (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Northwestern Polytechnical University of Technology, Xi'an, China, in 1991, 1994, and 1997, respectively. From 1997 to 1999, he worked with Zhejiang University as a Postdoctoral Researcher, where he was promoted to an Associate Professor. Since 1999, he has been on the faculty of the School of Computer Engineering and Science, Shanghai University. From August 2006 to August 2007, he worked as a Visiting Professor with the University of Aizu, Japan. From September 2013 to September 2014, he worked as a Visiting Professor with Purdue University Calumet, USA. He is currently the Vice Dean of the School of Computer Engineering and Science, Shanghai University. He has published three books and about 150 research papers in national and international journals and major conference proceedings. His current research interests include intelligent information processing, data science and technology, and intelligent human–computer interaction. He serves as a Steering Committee Member, the Workshop Chair, and a Program Committee Member of several important international conferences.

REHAN ULLAH KHAN received the B.Sc. and M.Sc. degrees in information systems from the University of Engineering and Technology at Peshawar, in 2004 and 2006, respectively, and the Ph.D. degree from the Vienna University of Technology, Austria, in 2011. He is currently an Assistant Professor with the IT Department, CoC, Qassim University, Saudi Arabia. His current research interests include machine learning, segmentation, and image-based object recognition.

TAE-SUN CHUNG received the B.S. degree in computer science from KAIST, in February 1995, and the M.S. and Ph.D. degrees in computer science from Seoul National University, in February 1997 and August 2002, respectively. He is currently a Professor with the Department of Software, Ajou University. His current research interests include flash memory storages, query processing in spatial databases, machine learning, and general database systems.

MUHAMMAD ATTIQUE received the bachelor's degree in information and communication systems engineering from the National University of Science and Technology, Pakistan, in 2008, and the Ph.D. degree in computer engineering from Ajou University, South Korea, in 2017. He is currently working as an Assistant Professor with the Department of Software, Sejong University, South Korea. His research interests include spatial computing, geosocial queries, location-based applications, and big data analysis.

KHALIL KHAN received the B.S. degree in electrical engineering and the M.S. degree in computer engineering from UET, Peshawar, Pakistan, in 2007 and 2012, respectively, and the Ph.D. degree from the Signals and Communication Laboratory, University of Brescia, Italy, in 2016. He is currently working as an Assistant Professor with the Department of Electrical Engineering, The University of Azad Jammu and Kashmir, Muzaffarabad, Pakistan. His research interests include digital image processing and machine learning.

SALIM EL KHEDIRI received the B.S. and M.S. degrees in fundamental computer sciences from the University of Luminy, Marseille, France, in 2009, and the Ph.D. degree in computer systems engineering from the University of Sfax, in 2016. Since 2010, he has been an Assistant Professor with the Department of Computer Sciences, Faculty of Sciences, Gafsa, Tunisia, and an Assistant Professor with the Information Technology Department, College of Computer, Qassim University, Saudi Arabia, since 2018. He is the author of several articles. His research interests include engineering of wireless sensors networks and wireless routing protocol, deep learning, and network security.

SADEEQ JAN received the B.Sc. degree in engineering from the University of Engineering and Technology at Peshawar (UETP), Peshawar, Pakistan, in 2004, the M.Sc. degree from the KTH Royal Institute of Technology, Sweden, in 2007, and the Ph.D. degree from the University of Luxembourg with excellent grade, in 2017, with the thesis entitled Automated and Effective Security Testing for XML-Based Vulnerabilities. He is currently an Assistant Professor and the Scientific Director of the National Centre for Cyber Security (NCCS),UETP, Pakistan. He worked as an Information Security Consultant in Sweden for several years providing services in the areas of penetration testing, PCI testing, ISO27001:2005 compliance process, and developing security policies for organisations. His research interests include information security, software testing and verification, search-based software engineering, and security testing. He was awarded the Presidential Award (IZAZ-E-SABQAT) and the University Gold Medal for academic excellence in Pakistan.