

## Article

# A Deep Learning-Based Course Recommender System for Sustainable Development in Education

Qinglong Li <sup>1</sup>  and Jaekyeong Kim <sup>1,2,\*</sup>

<sup>1</sup> Department of Big Data Analytics, Kyung Hee University, 26 Kyungheedae-ro, Dongdaemun-gu, Seoul 02453, Korea; leecy@khu.ac.kr

<sup>2</sup> School of Management, Kyung Hee University, 26 Kyungheedae-ro, Dongdaemun-gu, Seoul 02453, Korea

\* Correspondence: jaek@khu.ac.kr; Tel.: +82-2-961-9355

**Abstract:** Recently, the worldwide COVID-19 pandemic has led to an increasing demand for online education platforms. However, it is challenging to correctly choose course content from among many online education resources due to the differences in users' knowledge structures. Therefore, a course recommender system has the essential role of improving the learning efficiency of users. At present, many online education platforms have built diverse recommender systems that utilize traditional data mining methods, such as Collaborative Filtering (CF). Despite the development and contributions of many recommender systems based on CF, diverse deep learning models for personalized recommendation are being studied because of problems such as sparsity and scalability. Therefore, to solve traditional recommendation problems, this study proposes a novel deep learning-based course recommender system (DECOR), which elaborately captures high-level user behaviors and course attribute features. The DECOR model can reduce information overload, solve high-dimensional data sparsity problems, and achieve high feature information extraction performance. We perform several experiments utilizing real-world datasets to evaluate the DECOR model's performance compared with that of traditional recommendation approaches. The experimental results indicate that the DECOR model offers better and more robust recommendation performance than the traditional methods.



**Citation:** Li, Q.; Kim, J. A Deep Learning-Based Course Recommender System for Sustainable Development in Education. *Appl. Sci.* **2021**, *11*, 8993. <https://doi.org/10.3390/app11198993>

Academic Editors: Elisa Quintarelli and Ángel González-Prieto

Received: 19 April 2021

Accepted: 22 September 2021

Published: 27 September 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The development of internet technology has increased the available amounts of several types of data and has resulted in an information overload problem [1,2]. Recommender systems are widely utilized in internet applications to help users find their favorite items or services when they face too many options or too much information [3]. Recently, in preparation for the post-COVID era, the global education market has accelerated its transition from offline education to online education [4,5]. As the online education market proliferates, online course websites such as edX, Coursera and K-MOOC are becoming widespread and seeing increasing numbers of subscribers. Therefore, the range of course-related information available to users is increasing rapidly [6]. Several studies show that users face difficult situational problems due to the large amount of information available when choosing a course on an online education website [7,8]. This selection process is challenging and time-consuming when choosing their preferred course on such websites. The valuable information provided by a course recommender system can include relevant resource information such as job opportunities and the users' interests. Therefore, course recommender systems on online education websites must utilize diverse resources to match each user's individual goals, interests, and knowledge structure [9].

The process of user choosing personalized course can be highly challenging and complex. Above all, as of recently, the user can easily find information related to courses on an online education website [10]. However, the availability of more course information on an

online education website does not mean that users can choose personalized courses [11]. Instead, users may face the challenge of too many alternatives and information overload problems when using such websites [9]. Deep learning techniques have recently received much attention from several domains due to their higher performance than that of traditional techniques. Many studies have recently conducted deep learning approaches on recommender systems to enhance their performance [12]. However, few studies make personalized recommendations that utilize comprehensive information about the relevant online education domain. One of these studies' motivations is to help users for finding and choosing their preferred courses. Much course information continues to be uploaded on online education websites, and therefore, the recommendation of personalized courses for users is a significant challenge. Many online education websites have built course recommender systems based on traditional data mining approaches, such as Collaborative Filtering (CF), the association rules approach, and the most representative analysis technique [9,13–15]. While the previously developed course recommendation approach is highly effective and useful, it possesses certain significant limitations. First, a previous study has shown that recommender systems based on memory-based models that utilize user-item interactions do not meet individual users' personalized requirements. The most popular memory-based approach (e.g., CF) predicts user preferences from historical actions such as clicking, rating, and viewing. However, this approach only models the action itself without understanding the user behavior motivation. Second, the traditional approach has cold-start problems, where new users lack past data that could be used by the system to make recommendations. Another issue is the first-start problem by which a system cannot make recommendations until a user provides a preference rating. Another limitation of the previously developed recommendation approach is that it does not consider comprehensive information about the user's most relevant courses. For example, users choose courses based on information such as subjects, future careers, and certificates. Categorizing users' individual needs and areas of interest can recommend appropriate courses. It can help users choose personalized courses by integrating different data features and developing establish valuable course-related information. This is a motivation to develop a novel approach to overcome overload information and gain comprehensive knowledge of recommended items. The following research issues need to be addressed: First, integrate all available information about the course (e.g., job opportunities, user interests) and build relationships between the relevant information. Second, it needs to use all integrated information to recommend personalized courses that best suit users' individual needs.

Therefore, this study proposes a novel deep learning-based course recommender system (DECOR) model that can elaborately extract high-level user behaviors and course attribute features. The DECOR model can help users find their preferred courses, solve high-dimensional sparsity problems, and achieve higher feature information extraction performance than that of the traditional method. The DECOR model can also represent the interior features of attribute information structures in a relatively complex form. A user behavior extraction (UBE) module captures user behavior features, and a course attribute extraction (CAE) module captures course attribute feature information. A preference information regression (PIR) module integrates user behavior features and course attribute features into the obtained interaction features. We also conduct several experiments to evaluate the DECOR model's performance compared to that of the traditional recommendation approach using real-world datasets from the Korean education domain. The experimental results indicate that the proposed DECOR model provides better recommendation performance than existing methods. The main contributions of this study are as follows:

- The DECOR model proposed in this study effectively learns the interaction features between user behaviors and course attributes to solve the information overload and data sparsity problems.
- This study proposes a principled approach to integrating additional user and course feature information into the interaction features, thereby fully utilizing the combined information obtained from the user and course interaction features. The UBE module,

CAE module, and PIR module components are very flexible and modular, so they can be effectively adapted to the DECOR model.

- The study conducts several experiments utilizing real-world education datasets to demonstrate the DECOR model's effectiveness compared with that of the traditional recommendation approach. Additionally, we explore the proposed methodology's significance in the online education domain with a real-world dataset.

The remainder of this study is organized as follows: Section 2 discusses related work on traditional recommender systems, deep learning techniques in recommender systems, and course recommender systems. Section 3 discusses the framework of the DECOR model and its components in detail. Section 4 describes the utilized experimental datasets and parameter settings and discusses the results of this study in detail. Section 5 summarizes the study, describes its limitations, and presents ideas for future work.

## 2. Related Work

### 2.1. Traditional Recommender System

The development of internet technology has increased the available amounts of several types of data and has resulted in an information overload problem. Therefore, recommender systems are widely used in various application domains to help users find their favorite items or services [1,13,16]. Recommender system techniques and algorithms have been proposed in various fields, such as information search, marketing, education, and economics [13,17], which means they are exciting and valuable for both the academic and business worlds [13,18,19]. They have become increasingly popular since the mid-1990s, and they provide solutions to information overload problems on online platforms [20].

In a previous study, recommender systems were divided into Content-based Filtering (CBF) methods and CF-based methods [17]. CBF is a traditional approach that is applied when solving information overload problems [21,22]. It has been used for analyzing item contents and the similarities between items and user preferences to recommend suitable items. Consequently, this method has an overspecialization issue, in which items similar to those previously purchased are recommended [23,24]. The CBF approach recommends similar content to that which the user liked in the past, and unlike the CF approach, it does not make recommendations that utilize other users' preferences. The recommendation process of CBF involves finding similarities between user profiles and item features. This method can solve the first-start problem because items with the preferred item attributes are recommended to users. Another approach, CF, is a widely utilized technique in recommender system studies [13]. The idea of this approach is to calculate the similarities between neighborhoods and match users with relevant preferences to provide recommendations [25]. CF approaches are mainly divided into memory-based and model-based approaches [26,27]. Memory-based CF approaches are subdivided into user-based collaborative filtering (UCF) and item-based collaborative filtering (ICF) [17]. The UCF approach compares the ratings of the same items to calculate similarities between users. The item prediction for a given user is calculated as the weighted average of all users' evaluations of the item. The weights here are the similarities between these customers and the target items [13,17]. Conversely, the ICF approach uses the similarities between items to calculate predictions [28]. Model-based CF approaches utilize machine learning or data mining techniques to enhance model performance [13]. The strength of these approaches is that they use pretrained models and can quickly make recommendations. Additionally, they have proven capable of making recommendation results similar to those output by neighborhood-based recommendation techniques [28]. Typical model-based approaches include dimensionality reduction techniques such as singular value decomposition (SVD), regression, and clustering [3,17]. A model-based approach utilizes user-item interactions to capture the relationships between items and solve some of the sparsity issues associated with recommender systems [17].

Despite the success of CF approaches, some problems have been revealed. Such an approach essentially makes suitable recommendations based on users' preferences. How-

ever, recommender systems experience cold-start problems when encountering new users, as there are insufficient data for measuring similarities. Therefore, these systems cannot predict users' preferences [17]. Furthermore, there is a first-start problem in which users' preferred items cannot be recommended because they have not yet been purchased [17]. There is also a scalability problem that slows down a CF algorithm's computation process as the data become increasingly extensive. Most studies have recently started to apply deep learning techniques for recommender systems that can maximize each method's advantages, supplement the CF approaches' disadvantages, and effectively utilize various kinds of information [12].

## 2.2. Deep Learning Technique in Recommender System

Deep learning techniques have recently shown significant improvements over traditional techniques in several domains, such as image processing, computer vision, feature classification, and Natural Language Processing (NLP) [29,30]. Thus, many studies have begun to apply deep learning techniques in recommender systems in recent years. The strength of deep learning techniques is that they can provide effective solutions for overcoming the limitations of traditional approaches, such as accuracy, sparsity, and scalability. Therefore, the application of deep learning techniques in recommender systems has become a popular and important research topic [12,31].

Recently, several recommender system studies have applied deep learning techniques to enhance their recommendation performance [12]. Deep learning techniques can add nonlinear transformations to traditional recommendation approaches, thereby capturing complex user-item relationship representations [12,32]. In most cases, the conventional recommendation approaches consider the interactions between users and items. The Matrix Factorization (MF) approach divides the user-item matrix into user and item latent factors in a lower-dimensional space. It is essential to apply deep learning techniques for capturing the complex interactions between users and items [12]. Lian, et al. [33] proposed a cross-domain recommender system that combines collaborative filtering and content-based filtering through a neural network. Xue, et al. [34] proposed a DeepICF framework that utilizes a Multi-layer Perceptron (MLP) to effectively model the higher-order relationships among items. The results showed that the performances of DeepICF methods are superior to those of traditional item-based CF approaches. Additionally, Fu, et al. [35] proposed a CF framework that consists of learning the low-dimensional feature vectors of users and items and predicting ratings by applying Feed-forward Neural Networks (FFNNs) to model the interactions between users and items. The results showed that the proposed model was more effective than the traditional recommender system. Kiran, et al. [36] proposed a hybrid recommender system that applies embedding techniques to learn nonlinear interaction factors and integrate user and item feature information. The results showed that the proposed method exhibited better prediction performance than that of traditional recommender systems. Lian, et al. [37] proposed an eXtreme deep factorization machine to model explicit and implicit functional interactions to jointly improve DeepFM. The results showed that the machine outperformed state-of-the-art models. He and Chua [38] proposed regularizing a model utilizing dropout and batch normalization approaches and replacing the interaction component by applying an MLP. Additionally, a popular method called wide & deep learning is used in recommendation applications, and it can solve both regression and classification problems [39]. The wide learning component is a single-layer perceptron and can also be considered a generalized linear model. The deep learning component is an MLP. The reason for combining these two components is that the joint system can capture both memorization and generalization features. This approach can enhance recommendation performance, such as in terms of accuracy and diversity. By extending this model, Covington, et al. [40] proposed a YouTube recommender system by applying MLPs. Two deep MLPs can be found in the architecture of the proposed YouTube recommender system. The first deep MLP model is used for candidate generation, and the second model is used for ranking. The results showed that the new collaborative deep

filtering method outperforms the previous YouTube recommender system, which was implemented using the MF approach. Alashkar, et al. [41] has been proposed makeup recommendation algorithms by applying MLP. This approach applies the MLP model to expert rules and labeled examples, proposing makeup recommendation algorithms by applying an MLP. This approach applies an MLP model to expert rules and labeled examples. The parameters minimize the differences between the outputs and perform updates simultaneously. This approach demonstrates the effectiveness of adopting expertise to guide the learning processes of recommendation algorithms under an MLP framework.

When applying deep learning techniques in recommender systems, better performance is achieved than that of traditional recommender systems. However, few studies have explored the application of deep learning techniques in course recommendations for online education websites. This study examines the significance of a deep learning technique in terms of improving the performance of course recommender systems [12].

### 2.3. Course Recommender System

A previous study showed that users are overloaded by the large amount of available information when choosing a course [10,42]. Currently, the range of course-related information available to users is still rapidly increasing. Therefore, recommender systems for online education platforms are essential because they provide personalized course resources through user-course interactions [10]. Thus, the application of recommender systems in online education platforms has become a vital research domain [43,44]. In this regard, deep learning techniques might develop and imitate human decision-making and reasoning processes to minimize the uncertainty of effective learning and ensure lifelong learning mechanisms. Al-Badarenah and Alsakran [45] showed that course choices are influenced by user backgrounds and their personal or career interests. However, offering more course information on university websites does not necessarily suggest that users have the cognitive ability to evaluate them all as alternatives. Instead, this situation confronts users with a problem usually termed “information overload” [46]. It is challenging for students to evaluate all course alternatives themselves, even when search tools do exist. How to automatically find relevant courses to match students’ needs is a pressing problem [42]. Course recommender systems have made progress in helping users explore suitable courses. Outstanding results have been achieved in terms of establishing course recommender systems based on collaborative filtering approaches [47,48]. Additionally, Farzan and Brusilovsky [49] proposed a course recommender system that uses the social navigation method to make recommendations that take advantage of users’ career choices. This approach focuses on collecting explicit feedback from users implicitly as part of their natural communication system. The system’s basic and obvious advantage to users is a course administration system that stores information about courses they have chosen and facilitates communication with their advisors. Chen, et al. [50] proposed a personalized E-Learning recommender system based on item response theory (PEL-IRT). This recommendation approach considers the difficulty of course material and the user’s ability to recommend a suitable course. Users can select course categories and units and use relevant keywords to search for suitable course material. When browsing recommended course material and information, the user answers the system’s questionnaires so that the system can evaluate their ability and the appropriate course material difficulty level. Many studies have recently found that personalization is an essential factor for increasing recommendation performance and information retrieval [9,51]. Punj and Moore [52] realized that a recommender system that can filter and integrate information and offer feedback influences user decisions more than agents that are only aware of alternative options. In their research, relevant course recommendations were obtained by integrating helpful information from multiple valuable sources, such as job sites, social networks, and other related educational data sources [10]. O’Mahony and Smyth [53] proposed a course recommendation approach that applies an item-based CF technique that is widely used in recommender system studies. This method aims to enhance the online course selection

process by recommending personalized courses to users. The result shows that it can increase performance in terms of several metrics. Recent course recommender system studies have focused on usage in university curricula. Xiaoquan and Zhang [54] proposed a course recommender system based on CF techniques for professional users in civil engineering. Additionally, Liu, et al. [55] proposed a personalized recommender system for university physical education courses. A previous study showed that applying recommendation techniques in the education field can facilitate teaching and learning processes. Given the importance of education, the application of recommender systems can increase users' learning efficiency and recommendation validity.

All the above studies highlight the importance of course recommender systems in the education field. However, since most course recommender systems use traditional data mining techniques, there is a limit to input various types of information about courses or students enrolled. Moreover, since the traditional recommender system based on CF and CBF has limitations such as a scalability problem, sparsity problem or the cold start problem, it is necessary to develop a course recommender system based on the deep learning technique that is growing rapidly. Therefore, this study proposes a novel course recommender system that utilizes a deep learning technique to extract high-level user behaviors and course attribute features. It recommends personalized courses to users that consider individual needs, interests, and job experience and therefore support the decision-making process. Simultaneously, the proposed method reduces information overload and heterogeneity in relevant course search results by applying deep learning-based technology. Accordingly, this creates the need for a course recommender system that automatically excludes unrelated courses and allows users to see only the most appropriate options that are suitable for users' needs.

### 3. DECOR: A Deep Learning-Based Course Recommender System

We propose a novel DECOR model to provide personalized course recommendation services on online education websites. The DECOR model aims to reduce information overload and solve high-dimensional sparsity problems in the field of course recommendation. Additionally, it can capture high-level user behaviors and course attributes using multiple real-world data sources. The user behavior aspect includes four features: users, jobs, certificates, and language skills. Additionally, the course attribute aspect includes two features: courses and subjects. The DECOR model can predict each user's preference by learning latent features from user behaviors and course attributes. We first describe the DECOR model's problem formulation and framework in Sections 3.1 and 3.2, respectively. The developed UBE module, CAE module, and PIR module are described in Sections 3.3–3.5, respectively.

#### 3.1. Problem Formulation

This study collects information from user behavior features and course attribute features that express users' interests and preferences in the online education domain. Let define  $U$  and  $V$  be the sets of users and courses and the binary user-course matrix  $Y \in R^{|U| \times |V|}$ . Where  $Y$  indicates whether the user  $u$  takes the course  $v$  and if the user  $u$  has a taking record for course  $v$ , then  $Y = 1$ ; otherwise,  $Y = 0$ . The goal of recommendations for implicit feedback is to generate a course list base on the user's preference.

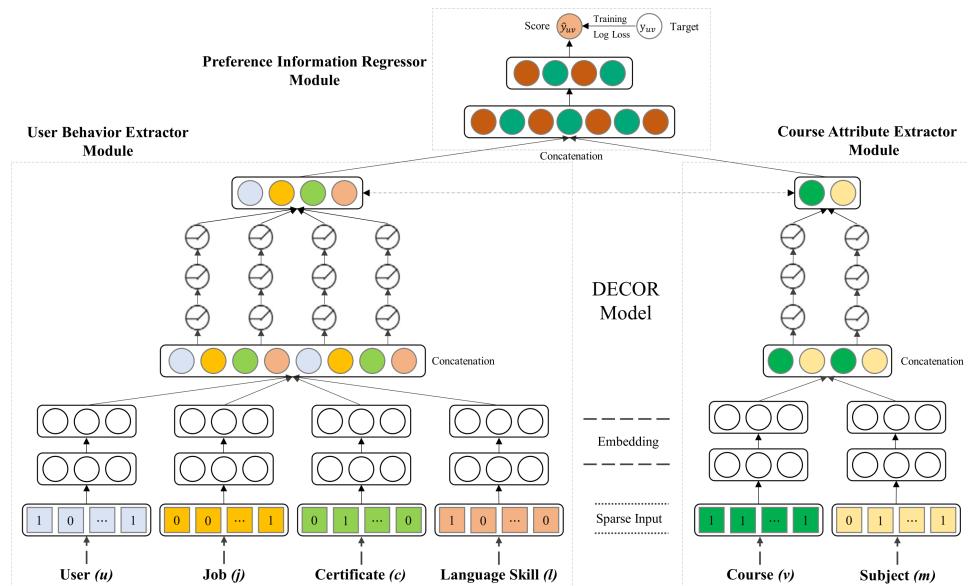
$$Y_{uv} = \begin{cases} 1, & \text{if user } u \text{ takes course } v \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

To recommend a personalized course for users on the online website, it should consider additional information such as subject, job, and language skills. As discussed above, additional information can help improve the recommendation performance. Therefore, we define the collected dataset as follows. Suppose we are given a set of user behavior features  $\mathcal{U} = (U, J, C, L)$ , where  $U = \{u_1, u_2, \dots, u_{|U|}\}$  indicates the set of users,

$J = \{j_1, j_2, \dots, j_{|J|}\}$  indicates the set of jobs,  $C = \{c_1, c_2, \dots, c_{|C|}\}$  indicates the set of certificates and  $L = \{l_1, l_2 \dots, l_{|L|}\}$  indicates the set of language skills. Here,  $|U|, |J|, |C|$  and  $|L|$ , indicate the numbers of users, jobs, certificates, and language skills, respectively.  $\mathcal{U}$  indicates the set of user behavior features, where each feature is indicated as a specific instance  $(u, j, c, l)$ ,  $u \in U$ ,  $j \in J$ ,  $c \in C$ , and  $l \in L$ ;  $u$  indicates a user,  $j$  indicates a job,  $c$  indicates a certificate, and  $l$  indicates language skills. We also define the course attribute feature as  $\mathcal{C} = (V, M)$ , where  $V = \{v_1, v_2, \dots, v_{|V|}\}$  indicates the set of courses and  $M = \{m_1, m_2, \dots, m_{|M|}\}$  indicates the set of subjects. Here,  $|V|$  and  $|M|$  indicate the numbers of courses and subjects, respectively. Additionally,  $\mathcal{C}$  indicates the set of course attribute features and is indicated as  $(v, m)$ ;  $v \in V$  and  $m \in M$ , where  $v$  indicates the course and  $m$  indicates the subject, respectively. The DECOR model utilizes a user behavior feature  $\mathcal{U}$ , and a course attribute feature  $\mathcal{C}$  to predict the probability that user  $u$  will interact with the course  $v$ . Therefore, this model can be defined as a predictive function  $\hat{y}_{uv} = \mathcal{F}(u, v; \Theta, \mathcal{U}, \mathcal{C})$ , where  $\Theta$  indicates the model parameters of function  $\mathcal{F}$ .

### 3.2. Framework

The architecture of the DECOR model is shown in Figure 1. The DECOR model consists of two parallel submodules, the UBE module and the CAE module, and a main module, the PIR module. The DECOR model utilizes user behavior features  $\mathcal{U}$  and course attribute features  $\mathcal{C}$  as inputs. Additionally, the model outputs the predicted probability that user  $u$  will choose course  $v$ . The UBE module is an FFNN used to capture high-level user behavior features and learn complex relationships. Similarly, the CAE module is used to learn high-level course attribute features. The PIR module integrates the high-level representation features obtained from the UBE and CAE modules into an end-to-end process to implement ensemble learning. The following three sections describe each module of the architecture specifically.



**Figure 1.** Overview of the proposed DECOR model framework.

### 3.3. UBE Module

The UBE module is an FFNN utilized to capture high-level user behavior features. The original user behavior feature vectors are highly sparse and high-dimensional, and they contain categorical and continuous values. Therefore, an input vector is converted to a low-dimensional, and dense real-valued vector called an embedding vector through the embedding layer and fed into the neural network's hidden layers in a forward pass.

This study follows the a common strategy from a previous study [39,56] that integrates all user behavior feature embeddings into the UBE module for high-level feature learning to enhance the model's recommendation performance. Accordingly, we first employ one-hot encoding when has given a user  $u$ , their job  $j$ , their certificate  $c$ , and their language skill  $l$ . Let  $v_u^U, v_j^J, v_c^C$ , and  $v_l^L$  be the feature vector for user  $u$ , job  $j$ , certificate  $c$ , and language skill  $l$ , respectively. We can obtain user behavior feature embeddings  $p_u, q_j, s_c$ , and  $z_l$  as follows:

$$p_u = P^T v_u^U, \quad (2)$$

$$q_j = Q^T v_j^J, \quad (3)$$

$$s_c = S^T v_c^C, \quad (4)$$

$$z_l = Z^T v_l^L, \quad (5)$$

where  $P \in R^{|U| \times D}, Q \in R^{|J| \times D}, S \in R^{|C| \times D}$  and  $Z \in R^{|L| \times D}$  indicate the embedding matric for user, job, certificate, and language skill, respectively;  $D, |U|, |J|, |C|$ , and  $|L|$  denote embedding size, the numbers of user, numbers of job, numbers of certificate, and numbers of language skill, respectively. Then, these latent vector representations are concatenated and fed into the hidden layer of the UBE module; this process is defined as:

$$\kappa^{(0)} = \phi_c(p_u, q_j, s_c, z_l) = \begin{bmatrix} p_u \\ q_j \\ s_c \\ z_l \end{bmatrix}, \quad (6)$$

where  $\kappa^{(0)}$  is the concatenated latent vector of user  $u$ , job  $j$ , certificate  $c$ , and language skill  $l$ . We utilize concatenation because it can minimize the loss of information. Then,  $\kappa^{(0)}$  is fed into the hidden layer, and this process is defined as:

$$\begin{aligned} \kappa^{(1)} &= \sigma_1(W_{UBE}^{(1)} \kappa^{(0)} + b_{UBE}^{(1)}) \\ &\vdots \\ \kappa^{(\ell)} &= \sigma_\ell(W_{UBE}^{(\ell)} \kappa^{(\ell-1)} + b_{UBE}^{(\ell)}) \end{aligned} \quad (7)$$

where  $\ell$  is the layer number;  $\sigma$  is an activation function; and  $\kappa^{(\ell)}, W_{UBE}^{(\ell)}$ , and  $b_{UBE}^{(\ell)}$  indicate the output, model weight, and bias of the  $\ell$ -th layer, respectively. The activation function utilizes Rectified Linear Unit (ReLU) activation functions in the first two hidden layers. The final hidden layer utilizes a tanh activation function that is consistent with the course attribute latent feature layer. The learned features in the UBE module represent the latent representations of user behaviors. The ReLU and *tanh* activation functions are defined as:

$$\text{ReLU}(x) = \max(0, x), \quad (8)$$

$$\tanh = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad (9)$$

### 3.4. CAE Module

The CAE module is an FFNN utilized to capture high-level course attribute features. Additionally, the original course attribute vector is converted to an embedding vector through the embedding layer. As with the above UBE module strategy, we integrate all output embeddings of the course attribute features into a CAE module to improve the recommendation of the model performance. We employ one-hot encoding when has given a course  $c$  and subject  $l$ . Let  $v_v^V$  and  $v_m^M$  be the feature vector for course  $v$  and subject  $m$ . We can obtain course attribute feature embeddings  $c_v$  and  $h_m$  as follows:

$$c_v = C^T v_v^V, \quad (10)$$

$$\mathbf{h}_m = \mathbf{H}^T \mathbf{v}_m^M, \quad (11)$$

where  $\mathbf{C} \in R^{|V| \times D}$  and  $\mathbf{M} \in R^{|M| \times D}$  indicate the embedding matrix for the course and subject.  $D$ ,  $|V|$  and  $|M|$  denote embedding size, the numbers of course, and numbers of subject, respectively. Then, these latent vectors are concatenated and fed into the hidden layer of the CAE module, and this process is defined as:

$$\boldsymbol{\omega}^{(o)} = \phi_c(\mathbf{c}_v, \mathbf{h}_m) = \begin{bmatrix} \mathbf{c}_v \\ \mathbf{h}_m \end{bmatrix}, \quad (12)$$

where  $\boldsymbol{\omega}^{(o)}$  is the concatenated latent vector of course  $v$  and subject  $m$ . Then,  $\boldsymbol{\omega}^{(o)}$  is fed into the hidden layer, and this process is defined as:

$$\begin{aligned} \boldsymbol{\omega}^{(1)} &= \sigma_1(\mathbf{W}_{CAE}^{(1)} \boldsymbol{\omega}^{(0)} + \mathbf{b}_{CAE}^{(1)}) \\ &\vdots \\ \boldsymbol{\omega}^{(\ell)} &= \sigma_\ell(\mathbf{W}_{CAE}^{(\ell)} \boldsymbol{\omega}^{(\ell-1)} + \mathbf{b}_{CAE}^{(\ell)}) \end{aligned} \quad (13)$$

where  $\ell$  is the layer number;  $\sigma$  is an activation function; and  $\boldsymbol{\omega}^{(\ell)}$ ,  $\mathbf{W}_{CAE}^{(\ell)}$ , and  $\mathbf{b}_{CAE}^{(\ell)}$  indicate the output, model weight, and bias of the  $\ell$ -th layer, respectively. As with the user behavior extractor, we utilize ReLU activation functions in the first two hidden layers. The final hidden layer utilizes a tanh activation function that is consistent with the user behavior latent feature layer. The learned features in the CAE module represent the latent representations of course attributes.

### 3.5. PIR Module

The UBE module and the CAE module capture the high-level latent features with the same dimensions. The PIR module also utilizes multiple layers to learn the high-level latent features of user behaviors and course attributes and incorporate them into the end-to-end process. All network parameters ( $\mathbf{W}^{(\ell)}$ ,  $\mathbf{b}^{(\ell)}$ ) are trained jointly for the combined model, which is defined as follows:

$$\hat{y}_{uv} = \text{sigmoid}(y_{UBE} + y_{CAE}), \quad (14)$$

The proposed DECOR model mainly focuses on implicit feedback, and  $\hat{y} \in (0, 1)$  is a binary class label. In other words, the output of the DECOR model is the probability value of the course that the user has not yet taken. Based on probability value, the model finally recommends top courses to users in the order of the highest probability values. Additionally,  $y_{UBE}$  is the output of the UBE module, and  $y_{CAE}$  is the output of the CAE module. The sigmoid activation function is defined as:

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}, \quad (15)$$

Finally, we define a loss function for the DECOR model and describe how to optimize the loss function. The DECOR model outputs the predicted rating  $\hat{y}_{uv}$  for each  $(u, v)$  pair. Considering recommendations for implicit feedback, user ratings can be regarded as labels for the user-course interaction. Where 1 denotes user-course interaction is observed, and 0 denotes otherwise. To address the absence of negative data, we sampled negative instances from an unobserved course [32,57]. The predicted  $\hat{y}_{uv}$  can be regarded as the possibility of user interaction with a course. Therefore, the predicted  $\hat{y}_{uv}$  is constrained to a range between 0 and 1 using the sigmoid activation function. The loss function can be defined as follows:

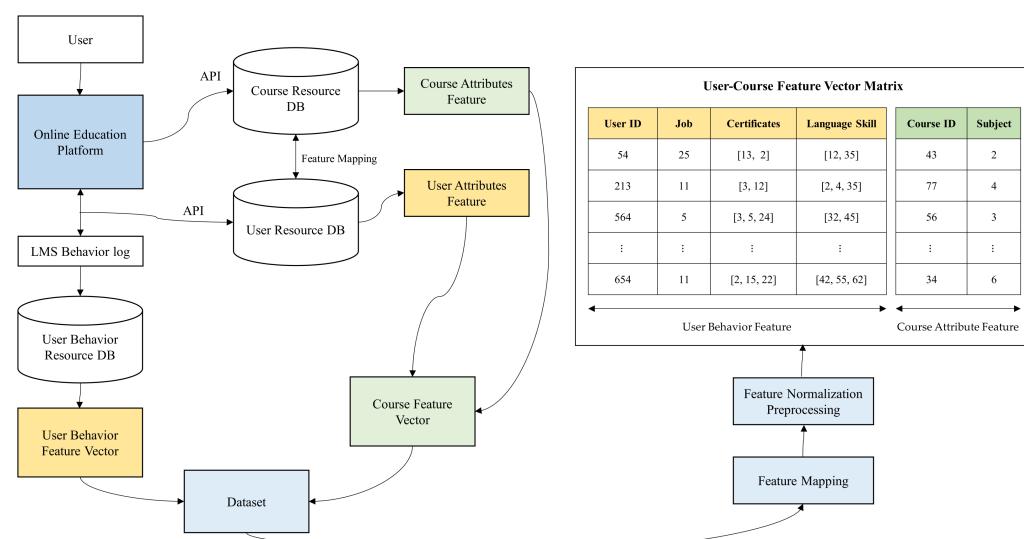
$$L(y_{uv}, \hat{y}_{uv}) = \sum_{(u,v) \in \chi} (y_{uv} - \hat{y}_{uv})^2 + \frac{\lambda}{2} \|\Theta\|^2, \quad (16)$$

where  $y_{uv}$  and  $\hat{y}_{uv}$  indicate the actual value and predicted value, respectively.  $\Theta$  indicates the set of parameters, and  $\lambda$  is a regularization parameter used to prevent overfitting. Additionally,  $\chi$  denotes a series of training datasets consisting of positive and negative feedback.

## 4. Experiments

### 4.1. Dataset and Evaluation Metric

In this section, we specifically describe the datasets used in this study. To evaluate the performance of the proposed DECOR model, we collected a dataset consisting of user behaviors and course attributes from a Korean online recruitment website and an online certificate website, respectively. The dataset collection process is shown in Figure 2. The dataset was collected from 891 users on 11,099 courses with a sparsity of 99.73%. The descriptive statistics of the collection datasets are summarized in Table 1. The dataset also contained 26,115 interactions between five criteria: jobs, certificates, language skills, and subjects, in addition to user ID and course ID information. Specifically, there were 25 job information, 15 language skill information, 235 certificate information, and 45 subject information types. Each interaction denoted whether the user had taken the course.



**Figure 2.** Dataset collection process.

**Table 1.** Statistics of the datasets and example.

Attribute Name	Attribute Type	Description	Example	Number of Attribute
User ID	String	Unique identifier ID for the user	"A23123"	891
Job ID	Integer	Desired job ID of the user	20	25
Certificate	String	Title of Certificate of the user	"OCP"	15
Language Skill	String	Type of Language skill	"HSK 6"	235
Course ID	Integer	Unique identifier ID for the course	3432	11,099
Subject	String	Subject name that user taken a course	"Department of Management"	45

To evaluate the performance of the DECOR model, we adopted the leave-one-out evaluation technique, which has been widely used in several studies [57,58]. We held out the users' latest interactions as the test dataset and utilized the remaining data for the model training dataset. This study followed a strategy that is common in current studies because it used too much time to rank all features for all users during the model evaluation [59,60]. This strategy randomly samples 50 courses that do not interact with the user and ranks the test items among the 50 total items. To evaluate the performance of the recommendation list, we will rank the list and return the top  $K$  ones as recommendations. We adopted the Hit Ratio (HR) at a rank  $K$  and Normalized Discounted Cumulative Gain

(*NDCG*) at a rank  $K$  metrics to evaluate the performance of a ranked list [61,62]. When making the recommendation list, we generated the top  $K$  items for each user. If the test item was present in the top  $K$  list, it was called a hit. When  $N$  is the number of test sets, the HR is computed as follows:

$$HR@K = \frac{\#Number\ of\ Hits@K}{N}, \quad (17)$$

The *HR* is a recall-based metric, so it does not reflect the accuracy of critical top ranks. To overcome this, we also adopted the *NDCG*, which explains the hits' position by assigning higher scores to hits with higher ranks. The *NDCG* is computed as follows:

$$NDCG@K = \sum_{i=1}^K \frac{2^{r_i} - 1}{\log_2(i + 1)}, \quad (18)$$

where  $r_i$  is the graded binary relevance of the item at position  $i$ . If the test item was present in the top  $K$  list,  $r_i$  became 1; it became 0 otherwise. For both metrics, a higher value indicates better recommendation performance. We calculated both metrics for each user and reported the average score.

#### 4.2. Baselines and Settings

This study evaluated the proposed model's performance compared with that of the most popular algorithms in recommendation studies. We used state-of-the-art baseline methods such as ItemKNN, MF, and Neural Collaborative Filtering (NCF), and these methods were divided into memory-based, model-based, and deep learning-based approaches, respectively.

ItemKNN is a standard item-based CF method based on neighborhood models in recommender systems. It has been applied to Amazon [63], MovieLens [64], and other online recommendation applications to optimize their performance. We followed the settings in the existing literature to adapt ItemKNN to an implicit dataset and tested the method with several recommendation sizes [65].

MF is an efficient model-based CF that obtains accurate recommendations, and it is easy to understand and implement. It also focuses on dimensionality reduction, where a reduced set of hidden factors contains user preferences [62].

NCF is a deep neural network-based recommender system that can capture complex nonlinear interactions between users and items. Additionally, it utilizes embedding vectors representing users and items as the inputs for capturing user-item interactions. Furthermore, it only uses implicit feedback and a pointwise loss function [32].

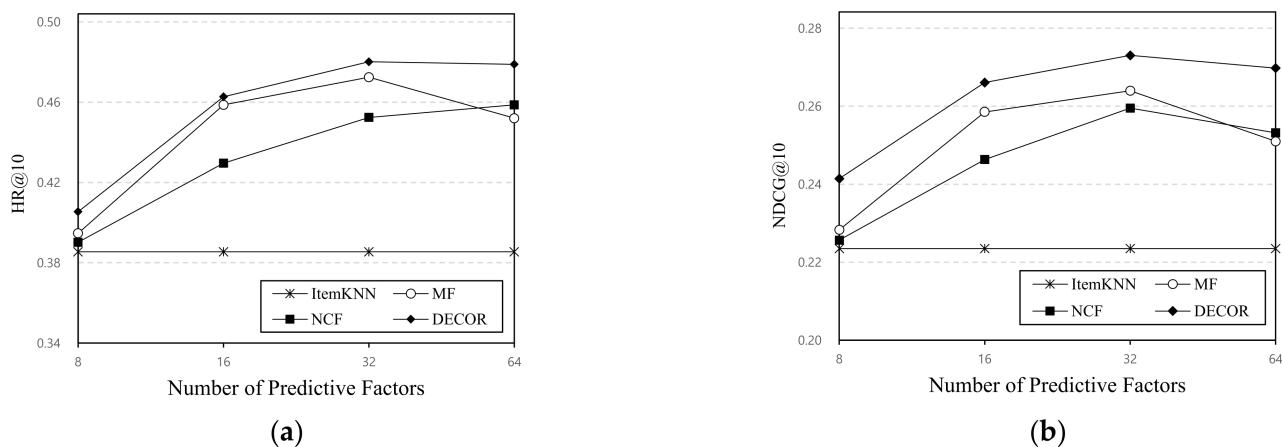
We also determined the impact of changing the number of latent factors (within {8, 16, 32, 64}) on model performance. The number of hidden layers was also equally set to {32, 16, 8} for comparison with existing studies. We set the optimal learning rate in the model training process to {0.0001, 0.0005, 0.001, 0.005} and conducted various experiments to change the parameters and optimize the proposed model. Based on the common strategy found in previous studies', the proposed model for used an Adam optimizer. This study set the batch size to {32, 64, 128, 256, 512} and then selected the optimal size. All experiments were conducted in a computer environment with 64 GB of memory and a 2080Ti GPU.

#### 4.3. Performance Evaluation

##### 4.3.1. Impact of Predictive Factors Number

In this section, we study the impact of changing the number of predictive factors on the model's recommendation performance. Like those in most previous recommender system studies, the collected real-world datasets in this study were highly sparse and high-dimensional. Therefore, we converted the input feature vector into a low-dimensional dense latent factor vector. To determine the optimal number of predictive factors, we followed the standard strategy used in previous studies by performing experiments, where

the number of predictive factors was set to 8, 16, 32 and 64 [32]. For the MF, NCF, and DÉCOR models, the numbers of predictive factors equaled the numbers latent factors. Additionally, in the ItemKNN approach, we performed experiments with several neighborhood sizes and reported the best performance. Figure 3 shows the HR@10 and *NDCG*@10 performances of the models based on the number of predictive factors. Here, the HR metric indicates whether a given test item was in the top-10 list, and the *NDCG* metric assigns a higher weight to a test item with a higher rank to describe its position.



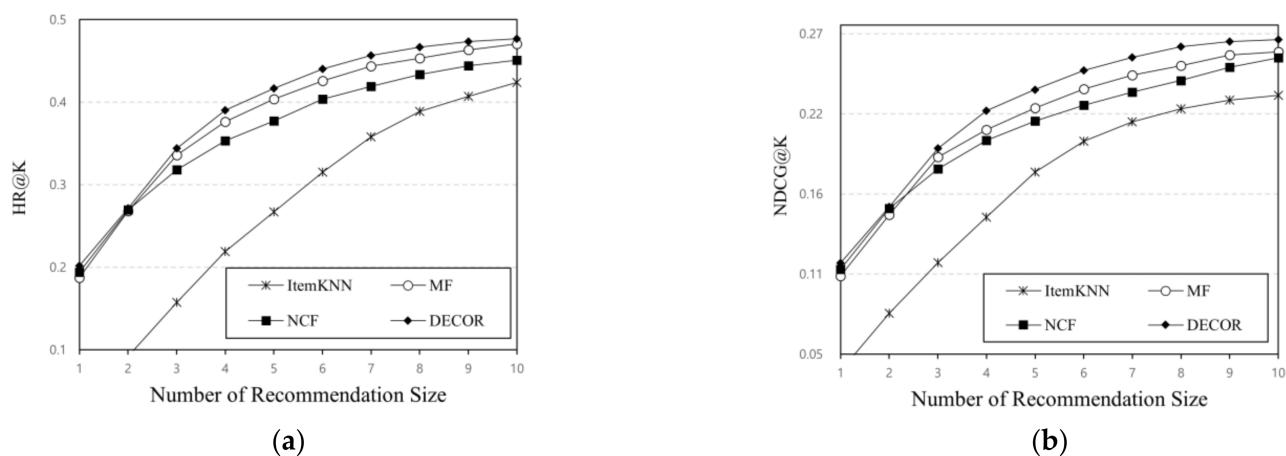
**Figure 3.** This figure shows the changes in model performance induced by changing the number of predictive factors: (a) HR@10 performance based on the number of predictive factors; (b) *NDCG*@10 performance based on the number of predictive factors.

The results of the experimental findings are discussed as follows. First, we can see that the proposed DÉCOR model exhibited the best performance on real-world datasets, outperforming the state-of-the-art approaches. Moreover, we found that both metrics indicate the same changing pattern. When the number of predictive factors was 32, the DÉCOR model indicated the best performance, outperforming the MF and NCF approaches. Furthermore, NCF and ItemKNN were followed by MF, where MF performed slightly better than NCF. Although MF exhibited a performance drop due to overfitting when the number of predictive factors was large, the performance of MF was better than that of NCF with smaller numbers of predictive factors. The results show that the DÉCOR model indicated high performance when integrating the UBE layer, capturing the user behavior features, and the CAE layer, capturing the course attribute features. Second, we found that the performance improved as the number of predictors increased. The best DÉCOR performance was reached when the number of predictive factors was 32. Additionally, when the predictive factor increased, the capture power of the latent factor-based model improved. Simultaneously, however, there was much noise, the model suffered overfitting problems, and the performance started to drop. Therefore, we need to set a correct number of predictive factors to optimize the model.

#### 4.3.2. Impact of Recommendations List Size

In this section, we study the impact of recommendation list size changes on model recommendation performance. We conducted various experiments with several recommendation list sizes ranging from 1 to 10 while utilizing the optimized number of predictive factors to create a ranked list in terms of performance. Figure 4 shows the HR and *NDCG* performances of the models based on the recommendation list size. We discuss the results of our findings as follows. First, we found that HR and *NDCG* of all approaches improved as the number of recommendation list sizes increased. Additionally, the proposed DÉCOR model demonstrated consistent improvements over other approaches when using several recommendation lists sizes. The results show that the HR and *NDCG* of the baseline approaches improved as the number of recommendations lists increased. Second, the MF

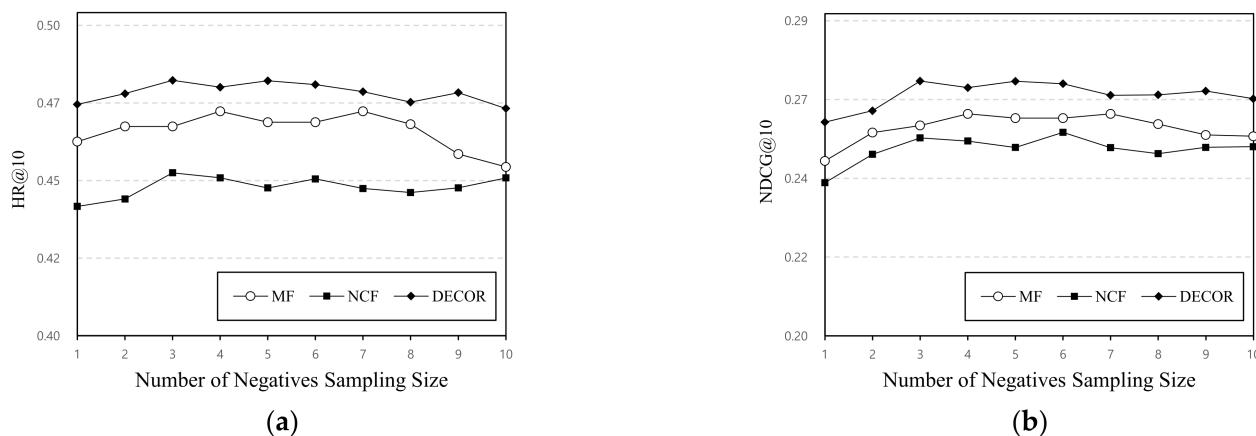
approach outperformed the NCF approach in terms of all performance improvements. As in previous studies, the MF approach achieved higher performance when ranking recommendations. Finally, the traditional ItemKNN approach exhibited lower recommendation performance than that of the model-based approach, indicating that personalized recommendation prefers a model-based approach rather than a memory-based approach.



**Figure 4.** This figure shows the impact of recommendation size on performance: (a) Evaluation of HR, where K ranges from 1 to 10; (b) Evaluation of NDCG, where K ranges from 1 to 10.

#### 4.3.3. Impact of Negative Sampling Size

In this section, we study the impact of changing the negative sampling ratio on model recommendation performance. The advantage of a pointwise log loss is that it can be set to a flexible negative sampling ratio. However, a pairwise objective function can be paired with only one negative sample. Therefore, we can easily set the number of negative samples for a pointwise loss. Negative sampling does not use all feature information when learning a model and selects and uses only some of this information. This study conducted several experiments with different numbers of negative samples ranging from 1 to 10 while utilizing the optimal number of predictive factors. Figure 5 indicates the HR@10 and NDCG@10 performances of the models with several negative sample sizes. We discuss the results as follows. First, this section studied the impact of the negative sampling ratio on the baseline model-based approach. We found that only one negative sample is insufficient for enhancing recommendation performance, and a greater negative sample should be used to improve the recommendation performance. Second, the DÉCOR approach exhibited better recommendation performance than the baseline approach under several numbers of negative samples. This result shows that the optimal number of negative samples ranges from approximately 3 to 6. When the number of negative samples was larger than 6, the performance of the DÉCOR approach decreased. We found that the performance worsened when the number of negative samples was set too high.



**Figure 5.** This figure shows the impact of a negative sampling ratio on performance: (a) HR@10 performance based on the number of negative sampling size; (b) NDCG@10 performance based on the number of negative sampling size.

## 5. Discussion and Conclusions

This study aims to solve the issue of information overload in the online education domain. Choosing a personalized course on an online education website may be incredibly tedious and complicated. Therefore, it is a real challenge to help users make the correct choices from among the many courses available that meet their requirements. Traditional methods based on data mining such as CF try to model the user's preferences from user historical behavior such as clicking, rating, and viewing [9,66]. For example, CF only uses user, course, and enrolment information to model the behavior pattern. It is almost impossible to input comprehensive knowledge about a given course and does not address the recommendation process's user requirements. Furthermore, when historical behavior records are sparse, the existing approach will result in a worse performance. This study proposed a novel DECOR model that can analyze the relationship between courses reasonably and efficiently, using all available information about the course (e.g., job opportunities, user interests, and so on). The experimental results indicate that the proposed DECOR model has better recommendation performance than traditional models, such as using only user, course, and preference. Although the amount of data we tested is not large, we believe that if we experiment with big data, better results will be obtained. This is because, according to the existing research results, when the data volume is very large, the deep learning technique produces better results than data mining.

This study focuses on the following three main aspects in the field of study of course recommender systems:

- Build a novel recommender system framework that can mitigate the information overload problem and restrict the number of user options to fewer alternatives.
- Propose a new approach for the extraction and integration of data from multiple sources, thereby mapping relevant information regarding a course to obtain comprehensive knowledge about the recommended items.
- Design and implement a deep learning-based recommender system that can analyze the relationship between courses reasonably and efficiently, using all available information about the course.
- Add significant contributions to overcoming user cold start challenges on traditional approaches. Also, such an approach integrating multiple attributes and applying deep learning techniques can improve the ability to find courses for the target user.

This study's main contribution is that it offers a personalized course recommender system that mitigates the information overload problem by building a comprehensive framework, which supports the extraction and integration of data from multiple features. The proposed framework also utilizes deep learning techniques to enhance the recommendation performance of the system and solve the cold-start problem caused by new users.

The contribution of this study in terms of the expansion of the recommender system and the development of the e-learning industry is summarized as follows.

- This study contributes to the knowledge of existing recommender systems by addressing how to solve problems in general and why they still have shortcomings. From a scientific aspect, it makes relevant contributions in the emerging domain of deep learning-based recommender systems.
- A novel recommendation framework is proposed based on a deep learning technique containing user behavior information and course attribute information. The framework includes multiple feature integration and personalized course recommendations, and it provides suitable recommendations for user's individual requirements. In summary, we developed deep learning-based hybrid recommendation algorithms designed to extract and integrate information from multiple features to improve recommendation performance by effectively solving the information overload and data sparsity problems.
- The proposed deep learning-based recommendation model is designed to extract and integrate information from multiple sources, enhancing recommendation quality and overcoming heterogeneity of course information. It can also be applied in diverse recommender system domains that change with the user's interests and attribute features.
- We develop a personalized recommender system, evaluate its effectiveness, and compare it with state-of-the-art methods on real-world datasets. The proposed approach can be applied to other online education platforms such as K-MOOC, edX and Coursera for all users.

The developed deep learning-based recommendation approach for the online education website domain improves recommendation performance. There are, however, still certain limitations encountered when applying it to online education websites. First, building deep learning-based recommendation algorithms is a challenging and complicated process that requires knowledge engineering skills. Additionally, deep learning knowledge in the online education domain requires expertise in this area. Second, course recommender system studies in online education domains have been a major challenge for many researchers. This is due to the lack of publicly available standard datasets for evaluating recommender systems. Other domains, such as movies and books, have many public datasets that can be used for evaluation purposes. However, there are not enough public datasets for training domain recommender systems. Hence, the assessment of a course recommender system is challenging. Third, this study mainly reported results on technical aspects related to course recommendation algorithms. In other words, we did not conduct behavioral studies when real users applied the algorithm proposed in this study. Future studies will improve the ability of the DECOR model to address the high-dimensionality and sparsity problems in practical online education platforms and will study more effective data reduction algorithms. Several experiments were conducted with a deep learning-based recommender system, and its performance was evaluated compared to that of other traditional recommender systems. Experimental results show that the utilized deep learning approach performs better than the conventional methods. Therefore, further study is needed to apply various other deep learning algorithms, such as a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN). Furthermore, investigating how user behavior features and course attribute features influence course recommendation algorithms' scalability and reliability will be another research area. Other feature interactions need to be considered in our future studies to develop highly personalized course recommendation algorithms. In addition, we will apply the proposed algorithm to the actual website for demonstration analysis. Through this work, we will effectively analyze the features that satisfy the actual user step by step.

**Author Contributions:** Conceptualization, J.K. and Q.L.; methodology, Q.L.; data curation, Q.L. and J.K.; writing—original draft preparation, J.K. and Q.L.; writing—review and editing, J.K.; supervision, J.K. Both authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the BK21 FOUR Program (519990913932) was funded by the Ministry of Education (MOE, Korea) and the National Research Foundation of Korea (NRF).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Kim, J.K.; Kim, H.K.; Oh, H.Y.; Ryu, Y.U. A group recommendation system for online communities. *Int. J. Inf. Manag.* **2010**, *30*, 212–219. [[CrossRef](#)]
2. Kim, H.K.; Ryu, Y.U.; Cho, Y.; Kim, J.K. Customer-driven content recommendation over a network of customers. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2011**, *42*, 48–56. [[CrossRef](#)]
3. Lu, J.; Wu, D.; Mao, M.; Wang, W.; Zhang, G. Recommender system application developments: A survey. *Decis. Support Syst.* **2015**, *74*, 12–32. [[CrossRef](#)]
4. Li, Y.; Nishimura, N.; Yagami, H.; Park, H.-S. An Empirical Study on Online Learners’ Continuance Intentions in China. *Sustainability* **2021**, *13*, 889. [[CrossRef](#)]
5. Bao, W. COVID-19 and online teaching in higher education: A case study of Peking University. *Hum. Behav. Emerg. Technol.* **2020**, *2*, 113–115. [[CrossRef](#)]
6. Bhumichitr, K.; Channarukul, S.; Saejiem, N.; Jiamthapthaksin, R.; Nongpong, K. Recommender Systems for University Elective Course Recommendation. In Proceedings of the 2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE), NakhonSiThammarat, Thailand, 12–14 July 2017; pp. 1–5.
7. Tarus, J.K.; Niu, Z.; Mustafa, G. Knowledge-based recommendation: A review of ontology-based recommender systems for e-learning. *Artif. Intell. Rev.* **2018**, *50*, 21–48. [[CrossRef](#)]
8. Tarus, J.K.; Niu, Z.; Yousif, A. A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. *Future Gener. Comput. Syst.* **2017**, *72*, 37–48. [[CrossRef](#)]
9. Zhang, H.; Huang, T.; Lv, Z.; Liu, S.; Zhou, Z. MCRS: A course recommendation system for MOOCs. *Multimed. Tools Appl.* **2018**, *77*, 7051–7069. [[CrossRef](#)]
10. Huang, C.-Y.; Chen, R.-C.; Chen, L.-S. Course-Recommendation System based on Ontology. In Proceedings of the 2013 International Conference on Machine Learning and Cybernetics (ICMLC), Tianjin, China, 14–17 July 2013; pp. 1168–1173.
11. Ibrahim, M.E.; Yang, Y.; Ndzi, D. Using Ontology for Personalised Course Recommendation Applications. In Proceedings of the International Conference on Computational Science and Its Applications (ICCSA), Trieste, Italy, 3–6 July 2017; pp. 426–438.
12. Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.* **2019**, *52*, 1–38. [[CrossRef](#)]
13. Park, D.H.; Kim, H.K.; Choi, I.Y.; Kim, J.K. A literature review and classification of recommender systems research. *Exp. Syst. Appl.* **2012**, *39*, 10059–10072. [[CrossRef](#)]
14. Zhang, H.; Yang, H.; Huang, T.; Zhan, G. DBNCF: Personalized Courses Recommendation System based on DBN in MOOC Environment. In Proceedings of the 2017 International Symposium on Educational Technology (ISET), Hong Kong, China, 27–29 June 2017; pp. 106–108.
15. Lofty, M.; Salama, A.; El-Ghareeb, H.; El-dosuky, M. Subject recommendation using Ontology for computer science ACM curricula. *Int. J. Inf. Sci. Intell. Syst.* **2014**, *3*, 199–205.
16. Choi, I.Y.; Oh, M.G.; Kim, J.K.; Ryu, Y.U. Collaborative filtering with facial expressions for online video recommendation. *Int. J. Inf. Manag.* **2016**, *36*, 397–402. [[CrossRef](#)]
17. Su, X.; Khoshgoftaar, T.M. A survey of collaborative filtering techniques. *Adv. Artif. Intell.* **2009**, *2009*, 1–19. [[CrossRef](#)]
18. Li, X.; Wang, M.; Liang, T.-P. A multi-theoretical kernel-based approach to social network-based recommendation. *Decis. Support Syst.* **2014**, *65*, 95–104. [[CrossRef](#)]
19. Cho, Y.H.; Kim, J.K.; Kim, S.H. A personalized recommender system based on web usage mining and decision tree induction. *Exp. Syst. Appl.* **2002**, *23*, 329–342. [[CrossRef](#)]
20. Konstan, J.A.; Miller, B.N.; Maltz, D.; Herlocker, J.L.; Gordon, L.R.; Riedl, J. GroupLens: Applying collaborative filtering to usenet news. *Commun. ACM* **1997**, *40*, 77–87. [[CrossRef](#)]
21. Pazzani, M.J. A framework for collaborative, content-based and demographic filtering. *Artif. Intell. Rev.* **1999**, *13*, 393–408. [[CrossRef](#)]
22. Pazzani, M.J.; Billsus, D. Content-based recommendation systems. In *The Adaptive Web*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 325–341.

23. Balabanović, M.; Shoham, Y. Fab: Content-based, collaborative recommendation. *Commun. ACM* **1997**, *40*, 66–72. [[CrossRef](#)]
24. Kim, H.K.; Oh, H.Y.; Gu, J.C.; Kim, J.K. Commanders: A recommendation procedure for online book communities. *Electron. Commer. Res. Appl.* **2011**, *10*, 501–509. [[CrossRef](#)]
25. Herlocker, J.L.; Konstan, J.A.; Terveen, L.G.; Riedl, J.T. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **2004**, *22*, 5–53. [[CrossRef](#)]
26. Breese, J.S.; Heckerman, D.; Kadie, C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, USA, 24–26 July 1998; pp. 43–52.
27. Bobadilla, J.; Ortega, F.; Hernando, A.; Gutiérrez, A. Recommender systems survey. *Knowl. Based Syst.* **2013**, *46*, 109–132. [[CrossRef](#)]
28. Isinkaye, F.O.; Folajimi, Y.O.; Ojokoh, B.A. Recommendation systems: Principles, methods and evaluation. *Egypt. Inform. J.* **2015**, *16*, 261–273. [[CrossRef](#)]
29. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; Volume 1.
30. Zhang, Q.; Yang, L.T.; Chen, Z.; Li, P. A survey on deep learning for big data. *Inf. Fusion* **2018**, *42*, 146–157. [[CrossRef](#)]
31. Shafqat, W.; Byun, Y.-C. Incorporating Similarity Measures to Optimize Graph Convolutional Neural Networks for Product Recommendation. *Appl. Sci.* **2021**, *11*, 1366. [[CrossRef](#)]
32. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.-S. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, 3–7 April 2017; pp. 173–182.
33. Lian, J.; Zhang, F.; Xie, X.; Sun, G. CCCFNet: A Content-Boosted Collaborative Filtering Neural Network for Cross Domain Recommender Systems. In Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, 3–7 April 2017; pp. 817–818.
34. Xue, F.; He, X.; Wang, X.; Xu, J.; Liu, K.; Hong, R. Deep item-based collaborative filtering for top-n recommendation. *ACM Trans. Inf. Syst.* **2019**, *37*, 1–25. [[CrossRef](#)]
35. Fu, M.; Qu, H.; Yi, Z.; Lu, L.; Liu, Y. A novel deep learning-based collaborative filtering model for recommendation system. *IEEE Trans. Cybern.* **2018**, *49*, 1084–1096. [[CrossRef](#)]
36. Kiran, R.; Kumar, P.; Bhasker, B. DNNRec: A novel deep learning based hybrid recommender system. *Exp. Syst. Appl.* **2020**, *144*, 113054. [[CrossRef](#)]
37. Lian, J.; Zhou, X.; Zhang, F.; Chen, Z.; Xie, X.; Sun, G. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1754–1763.
38. He, X.; Chua, T.-S. Neural Factorization Machines for Sparse Predictive Analytics. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, 7–11 August 2017; pp. 355–364.
39. Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M. Wide & Deep Learning for Recommender Systems. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, Boston, MA, USA, 15 September 2016; pp. 7–10.
40. Covington, P.; Adams, J.; Sargin, E. Deep Neural Networks for YouTube Recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 191–198.
41. Alashkar, T.; Jiang, S.; Wang, S.; Fu, Y. Examples-Rules Guided Deep Neural Network for Makeup Recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 941–947.
42. Ibrahim, M.E.; Yang, Y.; Ndzi, D.L.; Yang, G.; Al-Maliki, M. Ontology-based personalized course recommendation framework. *IEEE Access* **2018**, *7*, 5180–5199. [[CrossRef](#)]
43. Manouselis, N.; Drachsler, H.; Vuorikari, R.; Hummel, H.; Koper, R. Recommender systems in technology enhanced learning. In *Recommender Systems Handbook*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 387–415.
44. Jing, X.; Tang, J. Guess you like: Course Recommendation in MOOCs. In Proceedings of the International Conference on Web Intelligence, Leipzig, Germany, 23–26 August 2017; pp. 783–789.
45. Al-Badarenah, A.; Alsakran, J. An automated recommender system for course selection. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*, 166–175. [[CrossRef](#)]
46. Obeid, C.; Lahoud, I.; El Khoury, H.; Champin, P.-A. Ontology-Based Recommender System in Higher Education. In Proceedings of the Companion Proceedings of the The Web Conference 2018, Lyon, France, 23–27 April 2018; pp. 1031–1034.
47. Carballo, F.O.G. Masters' courses recommendation: Exploring collaborative filtering and singular value decomposition with student profiling. Master's Thesis, Instituto Superior Técnico, Lisboa, Portugal, 2014.
48. Ray, S.; Sharma, A. A Collaborative Filtering based Approach for Recommending Elective Courses. In Proceedings of the International Conference on Information Intelligence, Systems, Technology and Management, Gurgaon, India, 10–12 March 2011; pp. 330–339.
49. Farzan, R.; Brusilovsky, P. Social Navigation Support in a Course Recommendation System. In Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Dublin, Ireland, 21–23 June 2006; pp. 91–100.
50. Chen, C.-M.; Lee, H.-M.; Chen, Y.-H. Personalized e-learning system using item response theory. *Comput. Educ.* **2005**, *44*, 237–255. [[CrossRef](#)]
51. Salahli, M.A.; Özdemir, M.; Yasar, C. Concept Based Approach for Adaptive Personalized Course Learning System. *Int. Educ. Stud.* **2013**, *6*, 92–103. [[CrossRef](#)]

52. Punj, G.N.; Moore, R. Smart versus knowledgeable online recommendation agents. *J. Interac. Mark.* **2007**, *21*, 46–60. [[CrossRef](#)]
53. O’Mahony, M.P.; Smyth, B. A Recommender System for Online Course Enrolment: An Initial Study. In Proceedings of the 2007 ACM Conference on Recommender Systems, Minneapolis, MN, USA, 19–20 October 2007; pp. 133–136.
54. Xiaoquan, Z.; Zhang, X. Civil Engineering Professional Courses Collaborative Recommendation System based on Network. In Proceedings of the 2009 First International Conference on Information Science and Engineering, Nanjing, China, 26–28 December 2009; pp. 3253–3256.
55. Liu, J.; Wang, X.; Liu, X.; Yang, F. Analysis and Design of Personalized Recommendation System for University Physical Education. In Proceedings of the 2010 International Conference on Networking and Digital Society, Wenzhou, China, 30–31 May 2010; pp. 472–475.
56. Guo, H.; Tang, R.; Ye, Y.; Li, Z.; He, X. DeepFM: A factorization-machine based neural network for CTR prediction. *arXiv* **2017**, arXiv:1703.04247.
57. He, X.; Zhang, H.; Kan, M.-Y.; Chua, T.-S. Fast Matrix Factorization for Online Recommendation with Implicit Feedback. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, Pisa, Italy, 17–21 July 2016; pp. 549–558.
58. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* **2012**, arXiv:1205.2618.
59. Elkahky, A.M.; Song, Y.; He, X. A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 278–288.
60. Koren, Y. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 426–434.
61. He, X.; Chen, T.; Kan, M.-Y.; Chen, X. Trirank: Review-Aware Explainable Recommendation by Modeling Aspects. In Proceedings of the 24th ACM International Conference on Information and Knowledge Management, Melbourne, Australia, 18–23 October 2015; pp. 1661–1670.
62. Shani, G.; Gunawardana, A. Evaluating recommendation systems. In *Recommender Systems Handbook*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 257–297.
63. Linden, G.; Smith, B.; York, J. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput.* **2003**, *7*, 76–80. [[CrossRef](#)]
64. Vig, J.; Sen, S.; Riedl, J. Tagsplanations: Explaining Recommendations Using Tags. In Proceedings of the 14th International Conference on Intelligent User Interfaces, Sanibel Island, FL, USA, 8–11 February 2009; pp. 47–56.
65. Hu, Y.; Koren, Y.; Volinsky, C. Collaborative Filtering for Implicit Feedback Datasets. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 263–272.
66. Ricci, F.; Rokach, L.; Shapira, B. Introduction to recommender systems handbook. In *Recommender Systems Handbook*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1–35.