

# Deep Biased Matrix Factorization for Student Performance Prediction

Thanh-Nhan Huynh-Ly<sup>1,2,3</sup>, Huy-Thap Le<sup>1</sup> and Nguyen Thai-Nghe<sup>4,\*</sup>

<sup>1</sup>Department of Information Technology, Lac Hong University, Dong Nai Province, Vietnam

<sup>2</sup>Department of Information Technology, An Giang University, An Giang Province, Vietnam

<sup>3</sup>Vietnam National University Ho Chi Minh City, Vietnam

<sup>4</sup>College of Information and Communication Technology, Can Tho University, Can Tho City, Vietnam

## Abstract

In universities that use the academic credit system, selecting elective courses is a crucial task that can have a significant impact on a student's academic performance. Students who perform poorly in their courses may receive formal warnings or even face expulsion from the university. Thus, a well-designed study plan from a course recommendation system can play an essential role in achieving good academic performance. Additionally, early warnings regarding challenging courses can help students better prepare and improve their chances of success. Therefore, predicting student performance is a vital component of both the course recommendation system and the academic advisor's role. To this end, numerous studies have addressed the prediction of student performance using various approaches such as association rules, machine learning, and recommender systems. More recently, personalized machine learning approaches, particularly the matrix factorization technique, have been used in the course recommendation system. However, the accuracy of these approaches in predicting student performance still needs improvement. To address this issue, this study proposes an approach called Deep Biased Matrix Factorization, which carries out deep factorization via multi-layer to enhance prediction accuracy. Experimental results on an educational dataset have demonstrated that the proposed approach can significantly improve the accuracy of student performance prediction. By using this approach, universities can better recommend elective courses to their students as well as predict student performance, which can help them make informed decisions and achieve better academic outcomes.

**Keywords:** Educational data mining; deep matrix factorization; courses recommendation, student performance prediction.

Received on 15 March 2023, accepted on 14 April 2023, published on 20 April 2023

Copyright © 2023 Thanh-Nhan Huynh-Ly *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [CC BY-NC-SA 4.0](#), which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/eetcasa.v9i1.3147

## 1. Introduction

In a university setting, student performance prediction refers to the process of using data about a student's past academic performance to predict their future performance in a particular course or program. The goal is to identify students who may be at risk of not performing well and provide them with additional support or resources to help them succeed. In addition, universities can identify courses and programs that are likely to be a good fit for the student and align with their academic goals and interests [1].

Accurately predicting student performance can be very useful for universities and educators, as it can help ensure that students are placed in the right courses and programs and receive the support they need to succeed [2].

In recent years, the growing number of students who face academic difficulties, such as poor grades and academic probation, has become a matter of concern for educational institutions. For some students, these issues even lead to dropping out of school, negatively affecting their future career prospects and social well-being. The reasons for such difficulties are diverse, but one of the most significant factors is the student's inability to choose the right courses, which can result in overloading, unbalanced

\*Corresponding author. Email: [ntnghe@cit.ctu.edu.vn](mailto:ntnghe@cit.ctu.edu.vn)

curriculums, and mismatched interests. This issue is particularly problematic in the context of elective courses, where students have a wide range of options to choose from and often lack guidance and knowledge about the potential benefits and challenges of each course.

To address this problem, course recommendation systems have been proposed as a solution to provide students with personalized guidance on selecting courses that fit their interests, goals, and abilities. These systems use various methods, including collaborative filtering, machine learning, and data mining, to analyze student performance data, course attributes, and other relevant information and generate recommendations that optimize the match between the student and the course [3]. The potential benefits of such systems are numerous, including enhancing academic performance, reducing dropout rates, improving student satisfaction, and reducing the burden on academic advisors. However, the accuracy and effectiveness of these systems depend on several factors, such as the quality of the data, the complexity of the algorithm, and the validity of the evaluation metrics. Therefore, there is a need for ongoing research and development to improve the performance and usability of course recommendation systems in universities.

Several studies employ the learner's behavior or grades to predict student performance [4][5]. While the former is an implicit method that predicts student performance by observing their learning activities through the application system, the latter is a more explicit and straightforward method that relies on the student grading system available in all schools. As a result, using grades as a predictor of student performance is widely used in many research studies.

Effective academic advising is important for improving student retention, and new techniques for predicting student performance and the risk of failing or dropping a class can help address this issue. Personalized prediction approaches based on recommender systems are effective for accurately forecasting student grades in future courses and in-class assessments. For example, a study [6] with 772 students found that personalized prediction was much more effective than general rule prediction for the whole group of students, using video learning analytics and data mining techniques to predict students' overall performance at the end of the semester. This demonstrates that recommender systems for personalized advising are better than traditional data mining for suggesting general predictions. While many researchers have addressed student performance prediction using recommender systems, there is still room for improvement in these approaches [7].

In the field of artificial intelligence, specifically in machine learning, having a sufficiently large dataset is crucial for effective data mining. Fortunately, an educational data mining competition was held, providing a free dataset aligned with the goals of promoting educational data mining. Although the competition has ended, researchers interested in this field can still obtain the dataset with permission [8]. As a result, many studies in

educational data mining have utilized this dataset, including the study being used.

Deep Learning (DL) has become increasingly popular and has shown significant improvement in various domains, such as cybersecurity, natural language processing, bioinformatics, robotics, and medical information processing. As such, there is a growing need to have a more comprehensive understanding of DL and its techniques. In a recent study [9], the authors propose a holistic approach to provide a more appropriate starting point for developing a complete understanding of DL. They discuss the importance of DL and present the architecture of DL techniques and networks, which can be applied to other methods. This approach can be useful in developing advanced models for predicting student performance and recommending elective courses. With the increasing availability of educational data and the application of DL techniques, it is possible to develop accurate and personalized course recommendation systems that can greatly benefit students and universities.

The architecture of deep learning techniques has been successfully applied in various fields, including entertainment, and has the potential to be adapted to other methods. For instance, a previous study utilized deep learning architecture in matrix factorization to improve prediction accuracy in the entertainment industry [10]. As every method is suitable for a particular issue and data sample, the approach demonstrated positive results in the entertainment field and may be similarly effective in the education sector [11]. To this end, this study proposes a deep learning-based approach, called Deep Biased Matrix Factorization, to enhance the accuracy of student performance prediction. This method involves using multi-layered deep factorization to improve prediction accuracy, potentially enhancing course recommendation systems in educational settings.

## 2. Related work

In the field of educational data mining, predicting student performance is a critical task. A wide range of data mining techniques has been employed for this purpose, each with its own strengths and limitations. A survey was conducted to provide a comprehensive overview of the intelligent models and paradigms used in education [12]. The survey identifies various challenges in predicting student performance, such as the high dimensionality of educational data, class imbalance, and the lack of labelled data. It also discusses the pros and cons of several data mining algorithms, including traditional methods like decision trees, Bayesian models, and instance-based models, as well as more recent techniques like neural networks and support vector machines. Finally, the survey provides recommendations for future research in educational data mining, including the need for more transparent and interpretable models, the use of multi-modal data, and the development of hybrid models that combine different techniques.

In the field of educational data mining, many studies have explored the use of decision tree algorithms for predicting student performance. One such study, conducted by the author of [13], compared three different decision tree algorithms and found that J48 was the most effective for classifying and predicting student actions. The study also noted that the decision tree graphs' structure was influenced by the number of input attributes and the end class attributes. Another study examined using k-NN and decision tree classification methods to predict employee performance using internal data. Additionally, researchers have conducted comparative studies, such as [14], which compared the effectiveness of decision tree and Bayesian network algorithms for predicting student academic performance. These studies highlight the importance of selecting appropriate algorithms for specific prediction tasks and suggest avenues for further research in educational data mining.

In recent years, predicting student performance in higher education has gained significant attention, and various techniques have been proposed for this task [15]. One such technique is the association rule mining method in data mining, which has been employed to identify hidden relationships and patterns among student data to predict their academic performance. This method is effective in improving the accuracy of student performance prediction and can be used to provide personalized academic advising and support for students. For example, in a study [16], an association rule mining-based model was developed to predict student performance using data collected from a learning management system. The results demonstrated that the proposed model outperformed other commonly used machine learning algorithms in predicting student performance. This highlights the potential of using association rule mining for student performance prediction and the importance of exploring this technique further.

Additionally, another study has proposed an academic advising system using case-based reasoning (CBR) that recommends the most suitable major for each student by comparing the historical case with the student case, thus providing personalized advice [17].

The authors in [18] address developing a recommender system that suggests a set of learning objects to multiple students. In order to handle recommendations for groups, they represent the recommendation process as a non-cooperative game, intending to achieve a Nash equilibrium. They showcase the efficacy of their model through a case study experiment. In addition, they create a system that helps university students select elective courses by utilizing a hybrid multi-criteria recommendation system that incorporates genetic optimization techniques [19].

Rivera A.C. et al. [20] conducted a systematic mapping study of education recommender systems (RS) and identified several statistical methods to address the problem of predicting student performance using RS. In their paper [21], the authors propose several methods for building course recommendation systems, including student k-nearest neighbours, course-kNN, standard MF, and biased MF. These methods are analyzed and validated

using an actual data set before selecting the appropriate methods. The authors presented a comprehensive framework for building a course recommendation system. However, it is important to note that this study mainly focuses on the application systems and relies on baseline methods. Further research is needed to explore more advanced techniques and evaluate their performance.

There has been increasing interest in integrating social networks into RS, as studies such as [22] have shown that prediction accuracy can be improved by utilizing information from users' social networks. To this end, various methods have been proposed to integrate social networks into the MF. Several experiments have confirmed that social networks can provide independent sources of information that can be effectively utilized to enhance the quality of recommendations. Moreover, the study [23] demonstrated that incorporating the relationship between classroom members into the training model can significantly improve prediction accuracy. However, it should be noted that this algorithm only applies to a dataset with user relationships.

In addition, another paper [24] proposed a novel approach for incorporating the relationships between courses (e.g., knowledge/skills) into the MF, which can help to solve the PSP problem. This approach involves gathering information about course relationships and using this information to enrich the recommendation system. Such methods have shown great potential in improving the performance of RS, particularly in situations where traditional methods may not be effective.

Recently, there has been a growing interest in applying knowledge transfer between domains. For example, Tsiakmaki M. et al. [25] investigated the use of transfer learning in improving the performance of a learning model. Transfer learning is a machine learning approach that leverages knowledge obtained from one problem to improve the performance of a learning model on another related problem. In another study [26], deep learning models, such as Long Short-Term Memory and Convolutional Neural Networks, were proposed to predict student performance in educational data mining. The authors optimized these models by using various data preprocessing techniques, such as Quantile Transforms and MinMax Scaler, and applied robust machine learning approaches, such as Linear Regression, for prediction tasks. However, these studies did not incorporate personalized prediction using data mining techniques.

Collaborative filtering is a widely used technique for making recommendations in recommendation systems. To improve the quality of predictions and recommendations made to the user, the Matrix Factorization (MF) technique has been extended with a Deep Learning paradigm to create Deep Matrix Factorization (DeepMF) [27]. DeepMF uses a layered architecture that refines an MF model successively, with each layer using the knowledge acquired from the previous layer as input. The main objective of DeepMF is to learn high-level representations of user-item interactions and to capture the non-linear relationships between users and items. DeepMF effectively improves the

accuracy of recommendation systems, making it a promising approach for enhancing collaborative filtering. Another study has validated the effectiveness of DeepMF in a student performance prediction system [11].

The main aim of this study is to enhance the precision of predicting student performance through the extension of the Deep Matrix Factorization (DeepMF) model using Biased Matrix Factorization instead of the standard Matrix Factorization in the education domain. To achieve this aim, the paper offers an overview of the problem formulations used in predicting students' grades and explains the fundamental methods employed, including matrix factorization and biased-matrix factorization. The proposed approach integrates a deep factorizing architecture with the Biased Matrix Factorization to achieve a higher accuracy rate. The study then presents the findings and comparative analysis to evaluate the effectiveness of the proposed method in enhancing the precision of student performance prediction.

### 3. Proposed Method

#### 3.1. Problem Definition

The problem of predicting student performance has been mapped to a recommendation prediction task in several studies, including [21], [23], [24]. In a recommender system, there are three primary elements: user, item, and rating, where the task is to predict the user's rating for all unrated items and recommend the top-N highest predicted scores. Similarly, in the PSP problem, there are three virtual objects: student, course, and performance (correct or incorrect), and the task is to predict the course results that the students have not learned or solved in this context. Figure 1 depicts a similar mapping between the PSP and RS, with student, course, and grading becoming user, item, and rating, respectively.

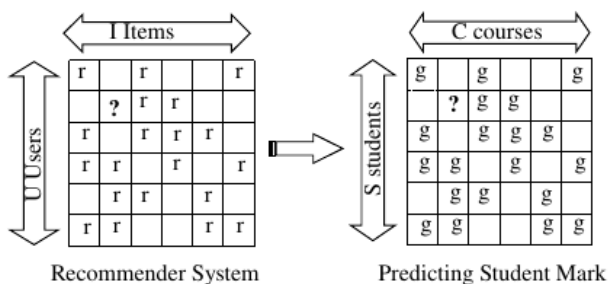


Figure 1. The similar mapping between PSP and RS

The availability of student scoring management systems in universities presents an opportunity to predict student performance using computer science methods. Despite the potential benefits, these systems have not been effectively utilized for this purpose. A common challenge is diversity in dataset structures, though they typically contain three key fields: student ID, course ID, and performance. By leveraging computer science methods, such as machine

learning and data mining, it is possible to uncover valuable insights that can aid in predicting student performance. However, it is important to carefully select the appropriate techniques and ensure the data is properly processed to achieve accurate results.

The process of factorizing students and problems based on their performance is illustrated in Figure 2. Throughout the rest of the paper, we will use the terms "course," "problem," and "task" interchangeably.

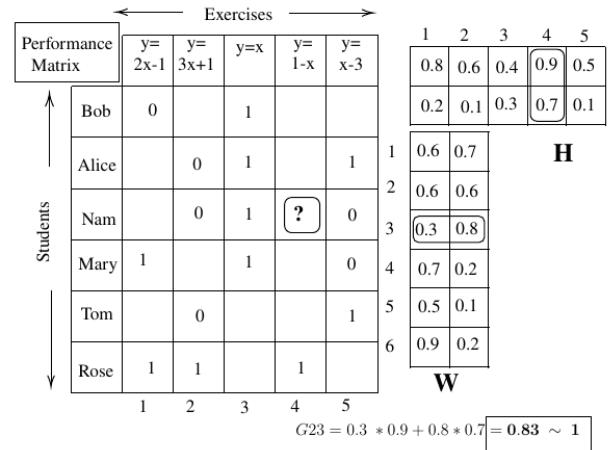


Figure 1. An example of factorizing on students and exercises

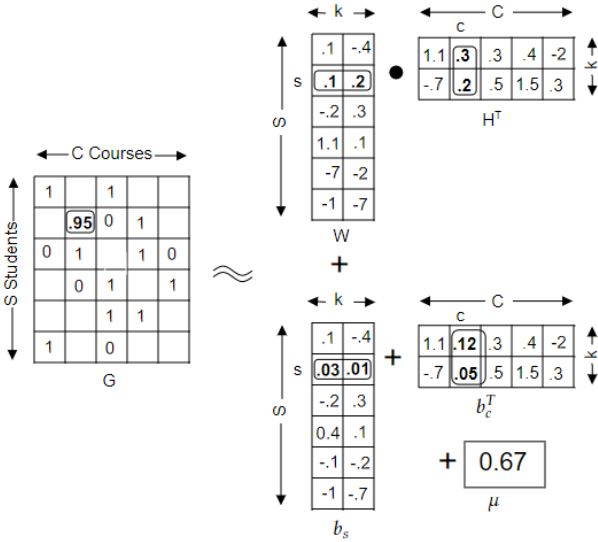
In the past few years, the research community has dedicated significant efforts to enhance the accuracy of predictive models by incorporating information from independent sources. This information integration has been demonstrated to improve results in several studies. Furthermore, some researchers have successfully applied models from other techniques to supplement their existing models and achieved positive outcomes. These endeavours have advanced the field's understanding of how various data sources and models can be combined to enhance prediction performance, showcasing the potential for further research in this direction.

#### 3.2. Biased Matrix Factorization Method

There are two main types of recommender systems: collaborative filtering and content-based filtering [28]. Content-based filtering systems use information about both the user and the item to find matches between users and items. On the other hand, collaborative filtering systems rely on the past behavior of users to recommend items in the future. Collaborative filtering can be further divided into two types: user-based collaborative filtering and item-based collaborative filtering. Similar users are identified in user-based collaborative filtering, and recommendations are made based on the items they rated highly. In item-based collaborative filtering, items are compared, and recommendations are made based on similar items that a user has shown interest in.

Collaborative filtering approaches can be classified into two main categories: memory-based and model-based collaborative filtering. Memory-based approaches, also known as neighborhood-based methods, rely on the user's or item's neighborhood for prediction. Examples of memory-based approaches include User-kNN and Item-kNN methods. On the other hand, model-based approaches typically use statistical models with some parameterization, such as latent factor models. Among the model-based approaches, matrix factorization (MF) has gained popularity, particularly after being used in the Netflix Prize competition.

The Matrix Factorization technique has gained popularity due to its scalability and high predictive performance. This approach involves creating a low-rank matrix approximation of the rating matrix, which can be enhanced by integrating additional terms to improve prediction accuracy. By combining the MF and the bias, a more advanced model called Biased Matrix Factorization (Biased-MF) can be created [21]. The main idea of matrix factorization is to approximate the matrix  $G^{(|S| \times |C|)}$  by an inner product of two smaller matrices (W and H). Biased-MF incorporates two small matrices ( $b_s, b_c$ ) and the global mean  $\mu$  to produce predictive ratings, making it more scalable. A model that factorizes the matrix with the biased term and generates a predicted grading for students to learn a course is shown in Fig. 3. The gradings in this model include the global mean, user-specific bias, item-specific bias, and the true interaction between users and items.



**Figure 2.** The prediction process of Biased MF

In the context of educational data mining, a popular method for predicting student performance involves the use of a matrix factorization technique. This approach involves the use of a matrix  $G = (g_{s,c}) \in G_{S \times C}$  that collects the grades  $g$  earned by students  $s$  for courses  $c$ . Typically, these grades are represented as float values between 0 and 1, or as an empty value  $\emptyset$  if the student has not taken the course. The goal is to factorize  $G$  into two smaller matrices,

$W$  and  $H^T$ , of dimensions  $s \times k$  and  $k \times c$ , respectively, where  $k$  is a lower rank than  $s$  and  $c$ . These matrices can be seen as projections or co-projections of the  $s$  students and  $c$  courses into a  $k$ -dimensional latent space. The elements of these matrices,  $w_{sk}$  and  $h_{ck}$ , respectively, can be used to predict the grade/mark  $g$  for a student  $s$  to learn a course  $c$ . To achieve this prediction, the dot product of the corresponding row of  $W$  and column of  $H$  is computed, which represents the predicted grade/mark.

$$\widehat{g}_{sc} = \sum_{k=1}^K w_{sk} h_{ck} = W_s H_c^T \quad (1)$$

In order to capture the interplay between users and items, the BiasedMF approach can be used within the context of Recommender Systems. This approach can address the "user effect" (also known as "user bias") and "item effect" (or "item bias") that can be present. In the realm of education, the user and item biases refer to the student and course biases, respectively. The student bias represents how proficient or unskilled a student is in completing a course, while the course bias captures how challenging or simple the course is to complete successfully. By incorporating these biases into the model, predictions can be improved in accuracy. The prediction function for a given student  $s$  on a given course  $c$  can be formulated as follows:

$$\widehat{g}_{sc} = \mu + b_s + b_c + W_s H_c^T \quad (2)$$

where  $\widehat{g}_{sc}$  represents the predicted performance of student  $s$  on course  $c$ ,  $\mu$  is the overall average performance in (3),  $b_s$  is the bias term for student  $s$  in (4),  $b_c$  is the bias term for course  $c$  in (5),  $W_s$  is the weight vector for student  $s$ , and  $H_c^T$  is the weight transpose vector for course  $c$ .

$$\mu = \frac{\sum_{(s,c,g) \in D^{train}} g}{|D^{train}|} \quad (3)$$

$$b_s = \frac{\sum_{(s',c,g) \in D^{train} | s'=s} (g-\mu)}{|\{(s',c,g) \in D^{train} | s'=s\}|} \quad (4)$$

$$b_c = \frac{\sum_{(s,c',g) \in D^{train} | c'=c} (g-\mu)}{|\{(s,c',g) \in D^{train} | c'=c\}|} \quad (5)$$

The Root Mean Square Error (RMSE) is a widely used criterion for optimizing  $W$  and  $H$  parameter values in matrix factorization models. RMSE measures the difference between the predicted and actual values of the ratings or grades in a recommender system. By minimizing the RMSE, we can obtain the optimal parameter values that lead to the best predictive performance of the model. The RMSE value is computed by taking the square root of the average of the squared differences between the predicted and actual ratings. Mathematically, the RMSE can be computed using the equation below:

$$RMSE = \sqrt{\frac{1}{|D^{test}|} \sum_{s,c,g \in D^{test}} (g_{sc} - \widehat{g}_{sc})^2} \quad (6)$$

The Matrix Factorization (MF) technique [18] involves optimizing  $W$  and  $H$  parameters during the model's training.  $W$  and  $H$  matrices are initialized with random values, typically from a normal distribution. Additionally,

an error function is incorporated to prevent over-fitting of the model. The error function can be expressed as follows:

$$O^{Biased-MF} = \sum_{(s,c,g) \in D^{train}} (g_{sc} - \mu - b_s - b_c - \sum_{k=1}^K w_{sk} h_{ck})^2 + \lambda (\|W\|_F^2 + \|H\|_F^2 + b_s^2 + b_c^2) \quad (7)$$

Where  $\|\cdot\|_F^2$  is a Frobenius<sup>†</sup> norm,  $\lambda$  is a regularization weight. The error function  $O^{Biased-MF}$  can be derived to the  $w_s$  and  $h_c$  resulting in the following updated rules for learning the model parameters. The  $w_{sk}$  and  $h_{ck}$  are updated by the equations below (where  $e_{sc} = g_{sc} - \hat{g}_{sc}$ , and  $w'_{sk}$  is the updated value of  $w_{sk}$ , and  $h'_{ck}$  is the updated value of  $h_{ck}$ ) The values of the  $w'_{sk}$  and  $h'_{ck}$  are carried out respectively.

$$w'_{sk} = w_{sk} + \beta(2e_{sc}h_{ck} - \lambda w_{sk}) \quad (8)$$

$$h'_{ck} = h_{ck} + \beta(2e_{sc}w_{sk} - \lambda h_{ck}) \quad (9)$$

The bias terms are typically initialized to 0 and then updated during the model training process, with the magnitude of the update being proportional to the learning parameters that are used for updating the main matrices  $W$  and  $H$ . Mathematically, the updated bias terms can be computed using the equation below:

$$b'_{sk} = b_{sk} + \beta(2e_{sc} - \lambda b_{sk}) \quad (10)$$

$$b'_{ck} = b_{ck} + \beta(2e_{sc} - \lambda b_{ck}) \quad (11)$$

Where  $\beta$  is the learning rate. We update the values of  $W$  and  $H$  iteratively until the error converges to its minimum ( $O_{n-1}^{MF} - O_n^{MF} < \varepsilon$ ) or reaching a predefined number of iterations. Finally, the performance of student  $s$  on courses  $c$  is now determined by equation (12):

$$\hat{g}_{sc} = \mu + b_s + b_c + \sum_{k=1}^K w_{sk} h_{ck} = \mu + b_s + b_c + w_s h_c^T \quad (12)$$

#### Algorithm Biased-MF-StudentPerformancePrediction

1. Let  $s \in S$  be a student,  $c \in C$  a course,  $g \in G$  a grade,  $\mu$  be an average grade from  $D^{train}$ .
2. Let  $w_{sk}, h_{ck}$  be latent factors of students, courses
3.  $W \leftarrow N(0, \sigma^2)$  and  $H \leftarrow N(0, \sigma^2)$
4. Initialize  $b_s$  and  $b_c$  to zero.
5. While (the stopping condition is NOT met) do
6.   for each  $(s, c, g_{sc})$  from  $D^{train}$
7.      $\hat{g}_{sc} \leftarrow \mu + b_s + b_c + w_s h_c^T$
8.      $e_{sc} = g_{sc} - \hat{g}_{sc}$
9.     For  $k = 1..K$  do
10.          $w'_{sk} = w_{sk} + \beta(2e_{sc}h_{ck} - \lambda w_{sk})$
11.          $h'_{ck} = h_{ck} + \beta(2e_{sc}w_{sk} - \lambda h_{ck})$
12.          $b'_{sk} = b_{sk} + \beta(2e_{sc} - \lambda b_{sk})$
13.          $b'_{ck} = b_{ck} + \beta(2e_{sc} - \lambda b_{ck})$
14.     End for
15.   End for
16. End while

<sup>†</sup> [https://en.wikipedia.org/wiki/Matrix\\_norm#Frobenius\\_norm](https://en.wikipedia.org/wiki/Matrix_norm#Frobenius_norm)

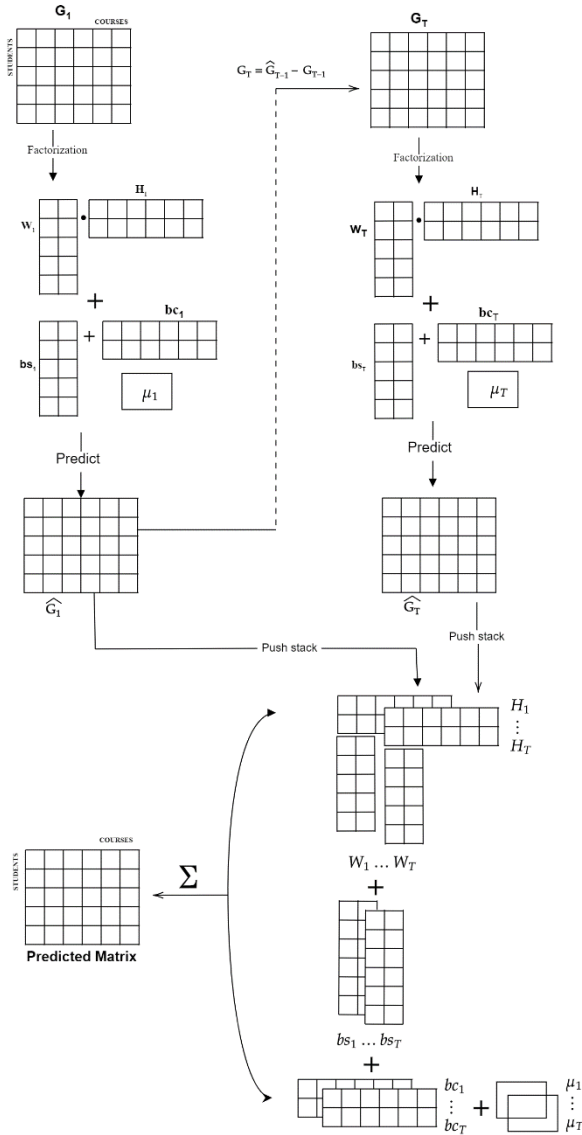
### 3.3 Deep Biased Matrix Factorization method

In the current landscape of predicting student performance, a common strategy is to enhance the accuracy of the predictions by utilizing various techniques of Matrix Factorization, including Biased-MF [21], Social-MF [23], and CRMF [24]. These techniques aim to pose an optimization problem by defining an error function that integrates multiple sources of information to evaluate the model's performance. The error function measures the divergence of the model from the actual data, and the objective is to minimize this error. Ultimately, the model with lower errors is considered more effective in predicting student performance.

One approach to improving the accuracy of matrix factorization-based recommendation systems is to use a deep factorizing architecture that combines multiple stages of biased matrix factorization methods. This approach involves recursively applying biased matrix factorization to the input matrix, with each iteration refining the approximation until the desired level of accuracy is achieved. The complete model can be viewed as a stack of models, with each stage using a biased matrix factorization technique to estimate the optimal values of the latent factors. By integrating multiple stages of biased matrix factorization, the model can capture more complex patterns in the data and improve the quality of recommendations.

Our objective is to expand on the paradigm of Deep Matrix Factorization as presented in the research papers [11][27] by introducing a novel integrated model named Deep Biased Matrix Factorization (DBMF). Figure 4 provides a visual illustration of the functioning of the DBMF model. The model's initialization is based on the input of the Biased Matrix Factorization in a Collaborative Filtering-based recommender system, which involves a matrix  $X$  comprising the grades/marks of students for various courses/exercises, along with their corresponding bias terms. DBMF model improves the quality of predictions by repeatedly training and refining the output.

As in the Biased-MF method, this matrix  $X$  will also be called  $X = G^1$  which is the beginning of the process. Approximating a matrix  $G_{|S| \times |C|}^1$  by a product of two smaller matrices,  $W^1$  and  $H^1$ , is a form  $\hat{G}^1 \approx W^1 \cdot H^1$ . The matrix  $\hat{G}^1$  provides all the predicted gradings stored in the stack at the first step. At this step, the recursive is begins. A new matrix  $G^2 = G^1 - \hat{G}^1$  is built by computing the attained errors between the original gradings matrix  $X$  and the predicted gradings stored in  $\hat{G}^1$ . A factorization again approximates this new matrix  $G^2$  into two new small-rank matrices  $\hat{G}^2 \approx W^2 \cdot H^2$ , which produces the errors in the second step  $G^3 = G^2 - \hat{G}^2$ . This process is repeated many times by generating and factorizing successive error matrices  $G^1 \dots G^T$ . Presumably, this sequence of error matrices converges to zero, so we get precise predictions as we add new layers to the model.



**Figure 3.** Graphical Model of the DBMF technique for Predicting Student Performance

Similar to the Biased-MF method presented in the baseline *methods* section above. Two small  $k$ -rank matrices are trained such that the product  $W^1 \cdot H^2$  is a good approximation of the rating matrix  $G^1 = X$ , that is, in the usual Euclidean distance. The term  $W^1 \in G^{1|s| \times |k^1|}$  is a matrix where each row  $s$  is a vector  $w_s$  (rendering the student  $s$ ) and has  $k^1$  latent factors.

Similarly, the term  $H^1 \in G^{1|c| \times |k^1|}$  is a matrix where each row  $c$  is a vector  $h_c$  (rendering the course  $c$ ) and has  $k^1$  latent factors. The approximation can be expressed as:

$$G^1 \approx \hat{G}^1 = W^1 \cdot H^1 \quad (13)$$

To implement the deep factorizing model, we subtract the approximation performed by  $\hat{G}^1$  to the original matrix  $X$ , to obtain a new sparse matrix  $G^2$  that contains the prediction error at the first iteration:

$$G^2 = X - \hat{G}^1 \approx G^1 - W^1 \cdot H^1 \quad (14)$$

Note that positive values in the matrix  $G^1$  mean that the prediction is low and needs to be increased. Similarly, the negative values in the matrix  $G^1$  mean that the prediction is high and needs to be decreased. Indeed, this adjustment is the main idea of applying the deep factorizing approach. To do this, we need to perform a new factorization to the error matrix  $G^2$  in such a way that

$$G^2 \approx \hat{G}^2 = W^2 \cdot H^2 \quad (15)$$

The approximation process in each step is performed similarly. However, two matrices  $W^2$  and  $H^2$  have orders  $s \times k^2$  and  $k^2 \times c$  for a definite number of latent factors  $k^2$ . Note that we should take  $k^2 \neq k^1$  to get various resolutions in the factorization. In the general case, if we computed at  $t-1$  steps of the deep factorizing, the  $t^{\text{th}}$  matrix of errors can be expressed:

$$G^t = G^{t-1} - \hat{G}^{t-1} = G^{t-1} - W^{t-1} \cdot H^{t-1} \quad (16)$$

At the step  $t$ , we are also factorizing into matrices  $W^t$  and  $H^t$  have the  $k^t$  latent factors until the sequence of error matrices converge to zero.

$$G^t \approx \hat{G}^t = W^t \cdot H^t \quad (17)$$

Once the deep factorization process ends after  $T$  steps, the original grading matrix  $X$  can be reconstructed by adding the estimates of the errors as

$$X \approx \hat{X} = \hat{G}^1 + G^2 = \hat{G}^1 + \hat{G}^2 + G^3 = \dots = \hat{G}^1 + \hat{G}^2 + \hat{G}^3 + \dots + \hat{G}^T = W^1 \cdot H^1 + W^2 \cdot H^2 + \dots + W^T \cdot H^T = \sum_{t=1}^T W^t \cdot H^t \quad (18)$$

For any step  $t = 1, \dots, T$  the factorization  $G^t \approx W^t \cdot H^t$  is sought by the standard method of minimizing the euclidean distance between  $G^t$  and  $W^t \cdot H^t$  by gradient descent with regularization. The error function for the DBMF now becomes:

$$O^{DBMF} = \sum_{t=1}^T \left( \sum_{(s,c,g) \in D^{train}} (g_{sc}^t - \sum_{k=1}^K w_{sk}^t h_{ck}^t)^2 + \lambda^t (\|W^t\|_F^2 + \|H^t\|_F^2) \right) \quad (19)$$

Where the term  $\lambda^t$  is the regularization hyper-parameter of step  $t$  to avoid overfitting. The error function  $O^{DBMF}$  can be derived to  $w_s^t$  and  $h_c^t$  resulting in the following updated rules for training the model parameters. The  $w_{sk}^t$  and  $h_{ck}^t$  are updated by the equations below (where  $e_{sc}^t = g_{sc}^t - \hat{g}_{sc}^t$ , and  $w_{sk}^{t'}$  is the updated value of  $w_{sk}^t$ , and  $h_{ck}^{t'}$  is the updated value of  $h_{ck}^t$ ). With the new error function, the values of  $w_{sk}^{t'}$ ,  $h_{ck}^{t'}$  and bias terms are updated, respectively

$$h_{ck}^{t'} = h_{ck}^t + \beta^t (2e_{sc}^t w_{sk}^t - \lambda h_{ck}^t) \quad (20)$$

$$w_{sk}^{t'} = w_{sk}^t + \beta^t (2e_{sc}^t h_{ck}^t - \lambda w_{sk}^t) \quad (21)$$

$$b_{sk}^{t'} = b_{sk}^t + \beta (2e_{sc} - \lambda b_{sk}^t) \quad (22)$$

$$b_{ck}^{t'} = b_{ck}^t + \beta (2e_{sc} - \lambda b_{ck}^t) \quad (23)$$

Where  $\beta^t$  is the learning rate hyper-parameter of step  $t$  to control the learning speed. In this way, after finishing the nested factorization, all the predicted ratings are collected in the matrix  $\hat{X} = (\hat{g}_{sc})$ , where the predicted grading of the student  $s$  to the course  $c$  is given by:

$$\hat{g}_{sc} = \sum_t^T w_s^t * (h_c^t)^T + \sum_t^T \mu + \sum_t^T b_s + \sum_t^T b_c \quad (24)$$

Note that this method consists of successive repetitions of the BiasedMF process using the results of the previous BiasedMF as input. All the parameters are stored in the stack, so a recursive implementation is easily used.

### 3.4 Proposed Algorithm - DBMF

The proposed method, which integrates deep factorizing architecture, is described in detail in the function "Deep-Biased-Matrix-Factorization (DBMF)". The algorithm takes the original matrix  $X$  and the model parameters as inputs. The output of the algorithm is a stack that contains the set  $\langle W, H, b_s, b_c, \mu \rangle$ .

It is worth noting that the parameters used in each factorization of the DBMF algorithm are stacked in a way that each factorization uses a different set of parameters. The parameters used in the first factorization are placed at the top of the stack, followed by the parameters used in the second factorization and so on, until the parameters used in the last factorization are placed at the bottom of the stack. This enables us to set the stopping criterion of the algorithm as the depth, typically around four layers.

The DBMF employs a recursive process to factorize student and course data using stacks of stochastic gradient descent with  $k$  latent factors, a  $\beta$  learning rate, a  $\lambda$  regularization weight, a stopping condition, and a depth. In each depth, parameters are popped to carry out the block of Biased-MF statements in lines 1-17, which includes the standard MF method and bias terms. The complete training models are recursively pushed to the stack in lines 20-21.

#### Algorithm DBMF-StudentPerformancePrediction

**Input:**  $D^{train}$ ,  $K$ , betas, lamdas, stopping condition, depth

**Output:**  $W, H$

1. Let  $s \in S$  be a student,  $c \in C$  a course,  $g \in G$  a grade
2. Let  $w_{sk}, h_{ck}$  be latent factors of students, courses
3.  $W \leftarrow N(0, \sigma^2)$  and  $H \leftarrow N(0, \sigma^2)$
4.  $k, t, \beta, \lambda \leftarrow \text{pop}(K, T, \text{Betas}, \text{Lamdas})$
5. Initialize  $b_s$  and  $b_c$  to zero.
6. while (the stopping condition is NOT met) do
7.     for each  $(s, c, g_{sc})$  from  $D^{train}$
8.          $\hat{g}_{sc} \leftarrow \mu + b_s + b_c + w_s h_c^T$
9.          $e_{sc} = g_{sc} - \hat{g}_{sc}$
10.        for  $k = 1..K$  do
11.             $w'_{sk} = w_{sk} + \beta(2e_{sc}h_{ck} - \lambda w_{sk})$
12.             $h'_{ck} = h_{ck} + \beta(2e_{sc}w_{sk} - \lambda h_{ck})$
13.             $b'_{sk} = b_{sk} + \beta(2e_{sc} - \lambda b_{sk})$
14.             $b'_{ck} = b_{ck} + \beta(2e_{sc} - \lambda b_{ck})$
15.        end for
16.     end for
17. end while
18. if (is-empty( $K$ )) return new stack ( $\langle W, H, b_s, b_c, \mu \rangle$ )
19. else
20.      $G' = G - (W \cdot H + \mu + b_s + b_c)$
21.     Params-Stack= DBMF ( $G', K, T, \text{Betas}, \text{Lamdas}$ )
22.     Return push ( $\langle W, H, b_s, b_c, \mu \rangle$ , Params-Stack)
23. end else.
24. end function.

## 4. Result

### 4.1. Dataset

The ASSISTments dataset, made available by the ASSISTments Platform, is a collection of data gathered from a web-based tutoring system that aims to help students learn mathematics and provide teachers with an assessment of their students' progress. This platform allows teachers to write customized questions, each of which includes associated hints, solutions, and web-based videos. After preprocessing, the ASSISTments dataset consists of 1011079 gradings (ratings) given by 8519 students (users) on 35978 tasks (items).

This investigation utilizes the ASSISTments dataset, a web-based math tutoring system that was initially developed in 2004 through a collaborative effort between the Worcester Polytechnic Institute and Carnegie Mellon University. The system was designed to provide students with personalized assistance and automated assessments of their performance at a detailed level [7]. ASSISTments is a popular tool middle and high school students use for their daily learning, homework, and preparation for the MCAS (Massachusetts Comprehensive Assessment System) tests. A snapshot of the ASSISTments dataset is depicted in Figure 5, which includes essential fields required for data mining, such as "User\_id," "Problem\_id," and "Correct."

	A	B	C	D	E	F	G
1	user_id	student_class_id	assignment_id	assistent_id	problem_id	skill_id	correct
2	77759	12138	245748	12914	12914	231	1
3	77759	12138	245748	15320	15320	231	1
4	77759	12138	245748	14529	14529	231	1
5	77912	12138	245698	1159	1159	100	0
6	77912	12138	245698	1647	1647	93	1
7	77912	12138	245698	2705	2705	100	1
8	77912	12138	245698	2186	2186	35	1
9	77912	12138	245698	1653	1653	35	1

Figure 4. A snapshot of the data sample

### 4.2 Evaluation

The primary objective of this study is to predict student marks using rating prediction, which falls under explicit feedback. As a standard measure for evaluating the models' performance in the Recommender Systems (RS) field, we have chosen Root Mean Squared Error (RMSE). We have adopted the hold-out approach for experimentation with the models, which splits the dataset into two parts, using two-thirds of the data for training and one-third for testing.

The accuracy of the prediction heavily relies on the parameters fed into the algorithm. If the parameters are unsuitable, the algorithm would produce inaccurate predictions, even if theoretically correct. Consequently, identifying the optimal parameters for the model is of utmost importance. Selecting the best set of hyperparameters entails finding the optimal values that



strike a balance between overfitting and underfitting the training data, and can significantly impact the model's predictive power. Thus, selecting the right set of hyperparameters is crucial in enhancing the model's performance.

Optimizing hyper-parameters is critical to improving the performance of machine learning models. In this study, we employed a parameter optimization technique known as hyper-parameters search to search for the best parameter values for the models used [21]. The hyper-parameters search method involves two stages: raw search, which is used to find the best hyper-parameters in the long data segments, and smooth search, which is used to find nearby optimal hyper-parameters for the short data segments. We utilized RMSE as the evaluation metric for the models, and the hyper-parameter search results for the ASSISTments dataset are presented in Table 1. This approach allows us to find the optimal hyper-parameters for the models, which in turn enhances the performance of the models in predicting student marks accurately.

TABLE I. Hyper-Parameters On The Assistments Dataset

Methods	Hyper Parameter
MF	$\beta=0.03$ , #iter=50, K=4, $\lambda=0.05$
Biased-MF	$\beta=0.0015$ , #iter=50, K=2, $\lambda=0.1$
DBMF	Deep (T)= 4 numFactors (K)= [5, 7, 5, 7]; numIters (Iter)= [50, 50, 50, 50]; learningRate ( $\beta$ ) = [0.5, 0.05, 0.5, 0.05]; regularization ( $\lambda$ ) = [0.03, 0.03, 0.03, 0.03])

Once the optimal hyper-parameters have been determined using the hyper-parameter search method, the selected parameters are utilized for training and testing the individual models. However, incorporating deep factorization in the models increases the training time, which can become increasingly time-consuming as the depth of the model increases.

### 4.3 Experimental Result

In this study, we have evaluated the performance of our proposed method, which integrates deep factorizing architecture into Biased-MF called (DBMF), against two baseline methods. The baseline algorithms, including MF and Biased-MF, are widely used and implemented in various open-source libraries, such as LibRec, MyMediaLite, and Collaborative Filtering For Java (CF4J). By leveraging these existing libraries, we could

extend and implement our proposed algorithm, DBMF, and compare its performance with the baseline algorithms.

We carried out three experiments to compare the proposed approach, DBMF, with other methods. The experimental results are presented in Figure 6, and they show that the proposed approach achieves the smallest RMSE of 0.332 on the dataset, which is the best result compared to the other methods. The smaller the RMSE, the better the model; hence, the proposed approach (DBMF) is considered the most accurate and effective method for predicting student performance.

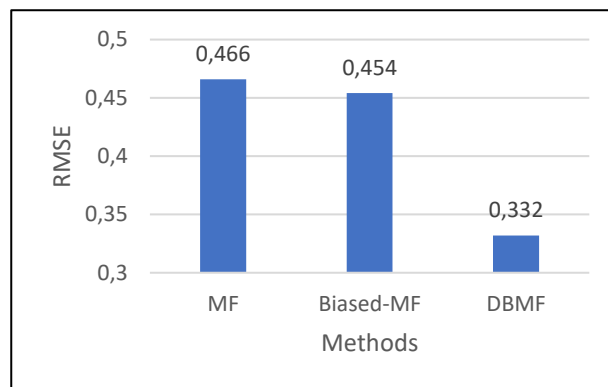


Figure 5. Experiment results on the ASSISTment dataset

## 5. Conclusion

The present study introduces an innovative approach to improving the accuracy of student performance prediction by leveraging the deep factorizing architecture to enhance the Biased-MF method. Applying deep factorizing principles can refine the model's output through successive training, significantly improving the prediction results. Experimental results on the published competition datasets reveal that this method is highly effective and outperforms the baseline approaches, including MF and Biased-MF.

While applying deep factorizing architecture to matrix factorization may slow down the training process, it can be addressed by implementing parallel algorithms. This study focuses on deep factorizing architecture for biased matrix factorization without using other complex integrating techniques. Further research could involve exploring meta-data integration to enhance the approach's effectiveness. Another potential avenue for future work is using graphics processing units (GPUs) to expedite the algorithm's execution.

In summary, the current work highlights the potential of utilizing deep factorizing architecture to enhance the accuracy of student performance prediction. By leveraging this technique, researchers and educators can better understand students' learning outcomes and tailor their teaching methods to meet individual needs. The proposed approach is simple to implement and can significantly improve prediction accuracy, making it a promising avenue for future research and development in this field.

## References

- [1] B. Albreiki, N. Zaki, and H. Alashwal, "A systematic literature review of student' performance prediction using machine learning techniques," *Educ. Sci.*, vol. 11, no. 9, 2021, doi: 10.3390/educsci11090552.
- [2] R. a. Sottolare, A. C. Graesser, X. Hu, and K. Brawner, "Design Recommendations for Intelligent Tutoring Systems Volume 3: Authoring Tools & Expert Modeling Techniques," *Book*, vol. 3, no. JUNE, pp. 1–388, 2015.
- [3] Q. Li and J. Kim, "A deep learning-based course recommender system for sustainable development in education," *Appl. Sci.*, vol. 11, no. 19, 2021, doi: 10.3390/app11198993.
- [4] K. Fahd, S. J. Miah, and K. Ahmed, "Predicting student performance in a blended learning environment using learning management system interaction data," *Appl. Comput. Informatics*, 2021, doi: 10.1108/ACI-06-2021-0150.
- [5] X. Tan, J. She, S. Chen, S. Ohno, and H. Kameda, "Analysis of Student Learning Behavior based on Moodle Log Data," in *International Conference on Human System Interaction, HSI*, Jul. 2021, vol. 2021-July, doi: 10.1109/HSI52170.2021.9538680.
- [6] R. Hasan, S. Palaniappan, S. Mahmood, A. Abbas, K. U. Sarker, and M. U. Sattar, "Predicting Student Performance in Higher Educational Institutions Using Video Learning Analytics and Data Mining Techniques," *Appl. Sci. 2020, Vol. 10, Page 3894*, vol. 10, no. 11, p. 3894, Jun. 2020, doi: 10.3390/APP10113894.
- [7] S. U. Masruroh, D. Rosyada, Zulkifli, Sururin, and N. A. R. Vitalaya, "Adaptive Recommendation System in Education Data Mining using Knowledge Discovery for Academic Predictive Analysis: Systematic Literature Review," 2021, doi: 10.1109/CITSM52892.2021.9588895.
- [8] T. Patikorn, R. S. Baker, and N. T. Heffernan, "ASSISTments Longitudinal Data Mining Competition Special Issue: A Preface," *J. Educ. Data Min.*, vol. 12, no. 2, pp. i–xi, Aug. 2020, doi: 10.5281/ZENODO.4008048.
- [9] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data 2021 81*, vol. 8, no. 1, pp. 1–74, Mar. 2021, doi: 10.1186/S40537-021-00444-8.
- [10] J. Bobadilla, S. Alonso, and A. Hernando, "Deep learning architecture for collaborative filtering recommender systems," *Appl. Sci.*, vol. 10, no. 7, 2020, doi: 10.3390/app10072441.
- [11] T. N. Huynh-Ly, H. T. Le, and N. Thai-Nghe, "Integrating Deep Learning Architecture into Matrix Factorization for Student Performance Prediction," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2021, vol. 13076 LNCS, doi: 10.1007/978-3-030-91387-8\_26.
- [12] H. Nawang, M. Makhtar, and W. M. A. F. W. Hamzah, "A systematic literature review on student performance predictions," *International Journal of Advanced Technology and Engineering Exploration*, vol. 8, no. 84. Accent Social and Welfare Society, pp. 1441–1453, Nov. 01, 2021, doi: 10.19101/IJATEE.2021.874521.
- [13] M. I. Hoque, A. kalam Azad, M. Abu Hurayra Tuhin, and Z. U. Salehin, "University students result analysis and prediction system by decision tree algorithm," *Adv. Sci. Technol. Eng. Syst.*, vol. 5, no. 3, 2020, doi: 10.25046/aj050315.
- [14] R. Ghorbani and R. Ghousi, "Comparing Different Resampling Methods in Predicting Students' Performance Using Machine Learning Techniques," *IEEE Access*, vol. 8, pp. 67899–67911, 2020, doi: 10.1109/ACCESS.2020.2986809.
- [15] C. Romero, S. Ventura, M. Pechenizkiy, and R. SJD Baker, "Handbook of Educational Data Mining."
- [16] L. M. Crivei, G. Czibula, and A. Mihai, "A Study on Applying Relational Association Rule Mining Based Classification for Predicting the Academic Performance of Students," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11775 LNAI, doi: 10.1007/978-3-030-29551-6\_25.
- [17] P. Dixit, H. Nagar, and S. Dixit, "Student performance prediction using case based reasoning knowledge base system (CBR-KBS) based data mining," *Int. J. Inf. Educ. Technol.*, vol. 12, no. 1, 2022, doi: 10.18178/ijiet.2022.12.1.1583.
- [18] A. Zia and M. Usman, "Elective learning objects group recommendation using non-cooperative game theory," *Proc. - 2018 Int. Conf. Front. Inf. Technol. FIT 2018*, pp. 194–199, Jan. 2019, doi: 10.1109/FIT.2018.00041.
- [19] A. Esteban, A. Zafra, and C. Romero, "Helping university students to choose elective courses by using a hybrid multi-criteria recommendation system with genetic optimization," *Knowledge-Based Syst.*, vol. 194, p. 105385, Apr. 2020, doi: 10.1016/J.KNOSYS.2019.105385.
- [20] A. C. Rivera, M. Tapia-Leon, and S. Lujan-Mora, "Recommendation Systems in Education: A Systematic Mapping Study," *Adv. Intell. Syst. Comput.*, vol. 721, pp. 937–947, Jan. 2018, doi: 10.1007/978-3-319-73450-7\_89.
- [21] H. L. Thanh-Nhan, H. H. Nguyen, and N. Thai-Nghe, "Methods for building course recommendation systems," *Proc. - 2016 8th Int. Conf. Knowl. Syst. Eng. KSE 2016*, pp. 163–168, Nov. 2016, doi: 10.1109/KSE.2016.7758047.
- [22] R. Chen, Y. S. Chang, Q. Hua, Q. Gao, X. Ji, and B. Wang, "An enhanced social matrix factorization model for recommendation based on social networks using

- social interaction factors,” *Multimed. Tools Appl.*, vol. 79, no. 19–20, pp. 14147–14177, May 2020, doi: 10.1007/s11042-020-08620-3.
- [23] H. L. Thanh-Nhan, L. Huy-Thap, and N. Thai-Nghe, “Toward integrating social networks into intelligent tutoring systems,” *Proc. - 2017 9th Int. Conf. Knowl. Syst. Eng. KSE 2017*, vol. 2017-Janua, pp. 112–117, Nov. 2017, doi: 10.1109/KSE.2017.8119444.
- [24] T. N. Huynh-Ly, H. T. Le, and T. N. Nguyen, “Integrating courses’ relationship into predicting student performance,” *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, pp. 6375–6383, Jul. 2020, doi: 10.30534/IJATCSE/2020/320942020.
- [25] M. Tsiakmaki, G. Kostopoulos, S. Kotsiantis, and O. Ragos, “Transfer Learning from Deep Neural Networks for Predicting Student Performance,” *Appl. Sci. 2020, Vol. 10, Page 2145*, vol. 10, no. 6, p. 2145, Mar. 2020, doi: 10.3390/APP10062145.
- [26] T. T. Dien, S. H. Luu, N. Thanh-Hai, and N. Thai-Nghe, “Deep Learning with Data Transformation and Factor Analysis for Student Performance Prediction,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 8, pp. 711–721, 2020, doi: 10.14569/IJACSA.2020.0110886.
- [27] R. Lara-Cabrera, Á. González-Prieto, and F. Ortega, “Deep Matrix Factorization Approach for Collaborative Filtering Recommender Systems,” *Appl. Sci. 2020, Vol. 10, Page 4926*, vol. 10, no. 14, p. 4926, Jul. 2020, doi: 10.3390/APP10144926.
- [28] F. Ricci, L. Rokach, and B. Shapira, “Introduction to Recommender Systems Handbook,” in *Recommender Systems Handbook*, Springer US, 2011, pp. 1–35.