Journal of Theoretical and Applied Information Technology
15th February 2024. Vol.102. No 3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

OBJECT-ORIENTED ONLINE COURSE RECOMMENDATION SYSTEMS BASED ON DEEP NEURAL NETWORKS

HAO LUO[1,2], NOR AZURA HUSIN[1], SINA ABDIPOOR[1], TEH NORANIS MOHD ARIS[1], MOHD YUNUS SHARUM[1] ,MASLINA ZOLKEPLI[1]

[1]Faculty of computer science and information technology, Universiti Putra Malaysia, Malaysia

[2]Department of big data, BaoShan University, China

E-mail: [1,3]{gs57436, abdipoor.sina}@student.upm.edu.my, [2,4,5,6]{n_azura, nuranis, m_yunus, masz }@upm.edu.my

ABSTRACT

In the era of widespread online learning platforms, students commonly face the challenge of navigating an extensive array of available courses. Identifying relevant and fitting options aligned with students' educational objectives and interests is highly complex. The impact of system maintainability and scalability on escalated development costs is often neglected in the literature. To tackle these issues, this paper introduces a comprehensive analysis and design of an object-oriented online course recommendation system. Employing a deep neural network algorithm for course recommendation, our system adeptly captures user preferences, course attributes, and intricate relationships between them. This methodology facilitates the delivery of personalized course recommendations precisely tailored to individual needs and preferences. The incorporation of object-oriented design principles such as encapsulation, inheritance, and polymorphism ensure modularity, maintainability, and extensibility, thereby easing future system enhancements and adaptations. The main contribution of this paper is to propose a new idea of an adaptive learning system that combines deep learning for personalized recommendations with object-oriented design for scalability and continuous improvement. This practical solution demonstrably enhances online learning experiences by tailoring recommendations to individual needs and evolving trends. Evaluation of the proposed system's performance utilizes real-world online course datasets, demonstrating its efficacy in furnishing accurate and personalized course recommendations, ultimately enhancing the overall learning experience for students.

Keywords: *Object-oriented, Deep Neural Network, Online Course Recommendation, System Design*

1. INTRODUCTION

The emergence of online learning platforms has revolutionized education, providing students with unprecedented access to a vast array of courses. However, this abundance of options has also created a paradox of choice, making it challenging for students to identify relevant and suitable courses that align with their interests and learning goals [1]. While traditional recommendation methods offer assistance, they often fall short, neglecting personalized needs and struggling with scalability and adaptability. To address this challenge, we present a comprehensive analysis and design of an object-oriented online course recommendation system.

Our proposed system aims to overcome the limitations of traditional recommendation methods by employing Deep Neural Networks (DNNs) to capture the complex relationships between users, courses, and their interactions [2]. DNNs are capable of extracting meaningful patterns from large amounts of data, enabling them to generate personalized recommendations that accurately reflect individual preferences and learning needs.

To ensure the system's scalability, maintainability, and adaptability, we adopt an object-oriented design approach [3]. This approach promotes modularity by encapsulating the system's functionality into well-defined and independent objects [4]. Inheritance enables code reuse and facilitates the creation of specialized recommendation modules [5]. Polymorphism allows for dynamic adaptation to different recommendation scenarios, ensuring the system's flexibility in handling diverse user preferences and course characteristics.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The effectiveness of our proposed system is evaluated using real-world online course datasets. Our experimental results demonstrate that the system accurately identifies relevant courses for students, significantly enhancing their learning experience. The system's object-oriented design further demonstrates its extensibility, paving the way for future enhancements and adaptations to accommodate evolving user needs and advancements in recommendation algorithms.

This paper presents a novel object-oriented online course recommendation system that addresses the challenges of recommending relevant courses to students in the vast landscape of online courses. The key contributions of our proposed methodology include:

- Personalized Recommendations: The system utilizes DNNs to capture user preferences, course characteristics, and intricate relationships between them, enabling personalized course recommendations that accurately match individual needs and learning goals.
- Modular and Extensible Design: Employing object-oriented design principles promotes modularity, maintainability, and extensibility, facilitating future enhancements and adaptations to accommodate evolving user needs and advancements in recommendation algorithms.
- Enhanced Learning Experience: Experimental results demonstrate the system's effectiveness in providing accurate and personalized course recommendations, significantly improving the learning experience for students.
- Scalability and Adaptability: The object-oriented design approach promotes scalability and adaptability, enabling the system to handle diverse user preferences and course characteristics while also accommodating future growth in the number of users and courses.

In the following sections, we delve into the detailed analysis and design of our object-oriented online course recommendation system. We will discuss the theoretical foundations of the system, the implementation of the DNN-based recommendation algorithm, and the object-oriented design principles employed to ensure the system's modularity, maintainability, and extensibility. We will also present the experimental results that validate the effectiveness of our system and highlight its potential to revolutionize the way students navigate the vast landscape of online courses.

The rest of this paper is organized as follows. Section 2 introduces the object-oriented approach and DNN-based recommendation system from the literature. Section 3 analyzes the requirements of the online course recommendation system. Section 4 explains the architecture, components, and functionalities of the DNN-based recommendation system, providing a detailed flow chart of its module. Section 5 introduces the process of the deep neural network recommendation model for course recommendations, along with a detailed exposition of the underlying DNN algorithm. Finally, Section 6 explores promising avenues for future research, while Section 7 summarizes the key findings and contributions of this work.

2. RELATED WORK

This section reviews object-oriented approaches and DNN-based recommendation systems as two key domains relevant to our proposed hybrid online course recommendation system.

2.1 Object-oriented Approach

Object-oriented methods have been widely applied in the design and implementation of recommendation systems, offering a structured and modular approach to handling complex data structures. Researchers[6]-[7] have highlighted the benefits of adopting object-oriented methodologies for building recommendation systems, emphasizing the encapsulation of functionality and data within well-defined objects.

Research proposed by [8] demonstrated the effectiveness of object-oriented design in recommendation systems by employing item hierarchies to represent user profiles, and recommendation algorithms. Their work showcased how encapsulating recommendation algorithms as objects facilitated modularity and maintainability, making it easier to adapt and extend the system with new features.

Moreover, another research [9] delved into the application of object-oriented programming in collaborative filtering-based recommendation systems. Their study revealed that representing users, items, and user-item interactions as objects improved code organization and reusability. The use

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

of inheritance and polymorphism in their object-oriented design allowed for the creation of flexible and scalable recommendation models.

These studies collectively emphasize the advantages of object-oriented methods in recommendation system development, including enhanced modularity, maintainability, and adaptability. By leveraging the principles of encapsulation, inheritance, and polymorphism, researchers have successfully designed recommendation systems that are not only effective in providing personalized suggestions but are also structurally robust and extensible.

2.2 DNNs-based Recommendation System

The integration of DNNs in online course recommendation systems has gained significant attention in recent literature. Research works, such as [10]-[11], have explored the application of deep learning techniques to enhance the performance and accuracy of recommendation models.

The work of [12] exemplifies this trend by introducing a deep autoencoder-based collaborative filtering method for online courses. They demonstrate that the hierarchical feature learning capabilities of autoencoders enable the model to capture complex user-item interactions and extract latent patterns, ultimately improving recommendation accuracy. This study underscores the potential of DNNs in uncovering intricate relationships within large-scale online education datasets.

Furthermore, researchers [13]-[14] delved into the use of Neural Collaborative Filtering (NCF) for personalized recommendation in online courses. The NCF model integrated multi-layer perceptron's with matrix factorization, offering a powerful framework for capturing both explicit and implicit user preferences. They emphasized that the non-linear transformations enabled by deep neural networks significantly enhanced the model's ability to discern subtle patterns in user-item interactions, contributing to more accurate and personalized recommendations.

These studies, along with others[15]-[16], collectively endorse the effectiveness of DNNs in recommendation systems. By leveraging hierarchical feature learning and non-linear transformations, DNN-based models can extract intricate patterns from complex data, providing users with more personalized and context-aware item recommendations [17]-[18]. The incorporation of deep learning techniques holds promise for advancing the state-of-the-art in online education recommendation systems due to their ability to model intricate dependencies and representations within the data.

3. SOFTWARE SYSTEM REQUIREMENTS ANALYSIS

Requirement analysis is an important part of the system implementation process and is the basis for system design and implementation [19]. It encompasses defining the system's functional modules and delineating the performance expectations for its non-functional aspects.

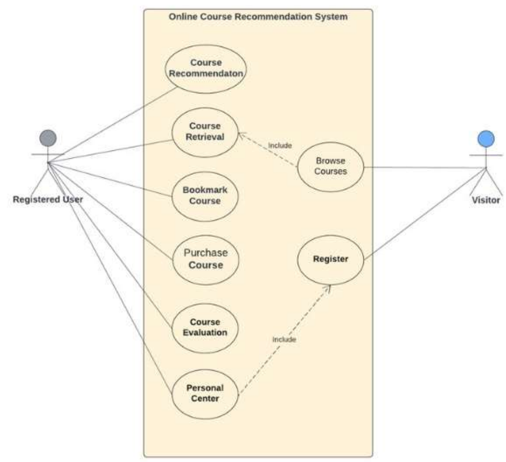

Figure 1: The use case diagram of online course recommendation system

The front-end functional module of the course recommendation system must address user requirements for course display and selection, providing personalized course recommendations [20]. On the other hand, the back-end module oversees the management of system courses, course categories, and user roles. Users are categorized as either registered or non-registered, with the latter limited to viewing and searching courses as guests. Figure 1 illustrates the system's use case diagram, depicting actors and various use cases. System actors include registered users and non-registered users, with the latter referred to as visitors in this research. The use cases encompass eight components: course recommendation, course retrieval, joining a course, bookmarking a course, course evaluation, personal center (including registration), browsing courses, and registration. Course browsing incorporates course searching, and the personal center integrates the registration use case. While registered users can

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

interact with all use cases, non-registered users are restricted to browsing courses; registration is a prerequisite for further engagement.

Based on an analysis of actual user needs, an effective online course recommendation system must fulfill the following requirements:

- **Usability**: The system should function seamlessly, with all modules performing as intended. This principle aligns with user-centric design principles [21]. Notably, the independent nature of the recommendation module enhances modularity and facilitates maintenance.
- **Recommendation Accuracy**: The system should prioritize accurate course recommendations, achieved through advanced algorithms leveraging user data like course collections and ratings [22]. Further refinement through course attribute analysis ensures personalized and satisfying recommendations.
- **Scalability**: The system should be designed for adaptability and growth. Its modular architecture, comprising separate back-end management and front-end functionalities [23], enables smooth integration with other online course systems, expanding recommendation capabilities.
- **Fault Tolerance and Stability**: Rigorous code testing at the module level is crucial. We utilize Tracy CI for continuous integration to swiftly identify and address potential issues [24]. Additionally, comprehensive exception handling, meticulous error log tracking with troubleshooting capabilities, and system status monitoring enhance stability and reliability [25].

4. MODULE DESIGN AND IMPLEMENTATION

4.1 System Architecture

The system is created using the MVC design pattern [26], encompassing page presentation, logic control, and data models. It is divided into front-end interface display and logical display, with each layer independently implementing specific functionalities, thereby achieving code reuse and enhancing the system's stability and scalability [27]. The model layer involves mapping and processing data tables, while the view layer utilizes Django templates to implement page layout and display, incorporating CSS styles. The control layer handles user requests through the view.py file in Django [28].

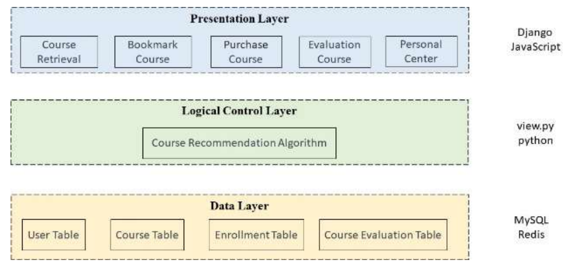

Figure 2: The architecture of the course recommendation system

Figure 2 illustrates the architecture of the course recommendation system. Specifically, the system structure of a course recommendation system includes three layers: data layer, logic control layer, and presentation layer. The bottom layer of the system structure is the data layer, which mainly stores various data, such as user tables, course tables, registration tables, and course evaluation tables. The middle layer of the system structure is the logic control layer. The function of logic control is to obtain data from the data layer, and then perform operations such as data cleaning, preprocessing, and feature extraction. Then, according to the user's interests and needs, use the recommendation algorithm to generate a recommendation list. The logic control layer is mainly composed of course recommendation algorithms. The top layer of the system structure is the presentation layer. The main function of the presentation layer is to respond to user requests, display the recommendation list generated by the logic control layer to users, and provide user-friendly interaction interfaces. The presentation layer mainly includes course search, course collection, course payment, course evaluation, and personal center functions.

Based on the requirements analysis [29] and the system architecture diagram [30], it can be inferred that the course recommendation system comprises front-end functionalities and a back-end management module. The front end includes modules such as login, course search, learning materials, and browsing history. Statistical-based course recommendation methods in the course recommendation system encompass premium courses, highly-rated courses, the latest courses, and popular courses. Personalized course recommendations for users are generated based on their course enrollment records and recommendation algorithms, aiming to suggest courses that align with individual interests [31]. This research combining these two types of course recommendations aim to

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

provide users with more opportunities for course selection, enabling a multi-level course selection approach to better meet users' needs.

4.2 System Module Design

In this section, we provide detailed descriptions of several representative modules in the system design and present flowcharts illustrating the processing steps for each module.

4.2.1 User Registration and Login Module

The registration and login module serves as the entry point to the online course recommendation system, as illustrated in Figure 3 depicting the flow of the registration and login process. The user registration module allows users to register, facilitating the creation of user preference models for course recommendations. Users can register by providing information such as username, name, password, major, and university. Upon successful registration, user information is automatically added to the database table, enabling them to log in seamlessly. If the information is already in the database, the system will prompt that the user has already registered. Otherwise, the system will prompt that the registration was successful and the user can log in.

In the user login module, users input their username and password on the login interface to access the system. If the input is correct, users can successfully enter the system; if incorrect, the system prompts that the username or password is incorrect, resulting in a login failure. If the username or password is incorrect, the system will prompt the user to re-enter the correct credentials. Repeating the previous steps will complete the login process.

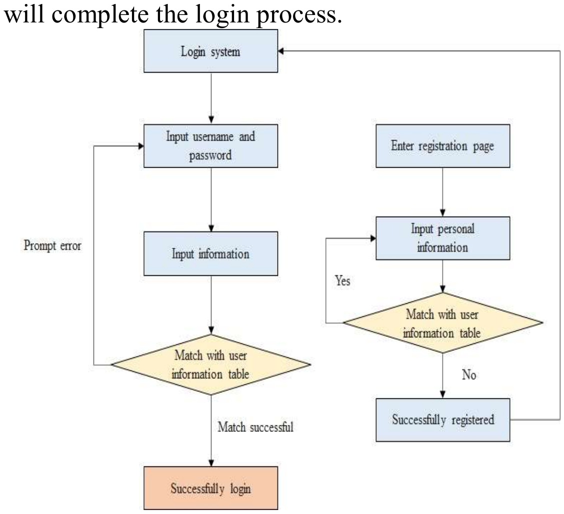

Figure 3: Login and registration module flow chart

4.2.2 Course Search Module

This module refers to obtaining the courses that users need through a series of search operations. Clicking on the search box redirects the user to the course search page, where users input keywords for the courses they wish to find. Users can perform searches based on teacher names or course titles. A dropdown list is provided below the search box, allowing users to select the other keywords just as college, class location, and practical aspects of the course. After the user enters a keyword, the system will search the database for any course that matches the keyword. If the system finds a match, it will return a list of courses that match the keyword. If the system does not find a match, it will return a message stating that no courses were found. As illustrated in Figure 4, depicting the process of course search.

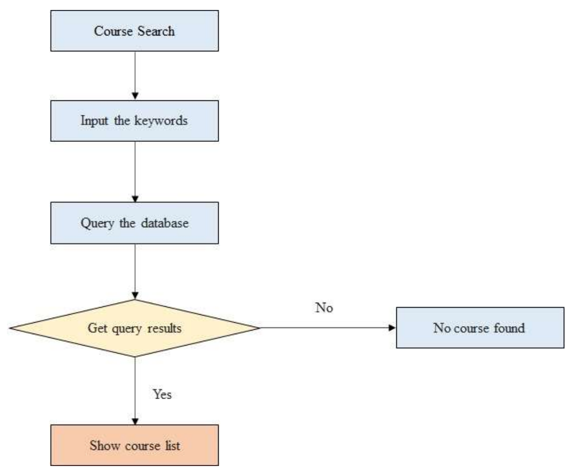

Figure 4: Course search flow chart

4.2.3 Course Recommendation Module

After a user search for courses, the system seamlessly shifts into course recommendation mode. To ensure highly accurate suggestions that align with evolving interests, it incorporates user interest modeling into its recommendation algorithms. The primary objective of this course recommendation module is to construct a course similarity matrix, which ultimately predicts a personalized Top-N list of courses tailored to each user's preferences. This matrix consists of two essential components: the course content similarity matrix and the course attribute similarity matrix.

To create the course content similarity matrix, the system first gathers comprehensive course content data, including course name, type, difficulty, duration, instructor, and ratings. It then meticulously extracts features from this data that accurately reflect similarities in course content. Finally, it leverages these extracted features to meticulously calculate the similarity between each possible course pairing, resulting in the course content similarity matrix.

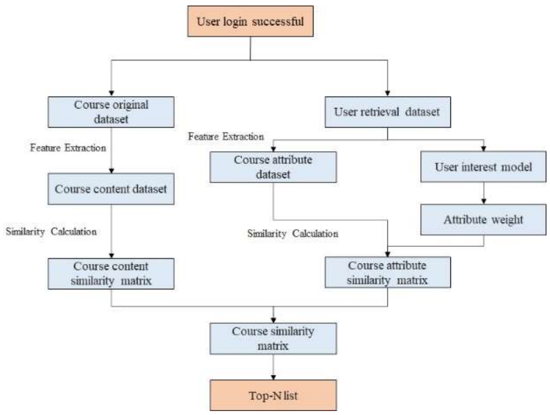

Figure 5: Course recommendation module flow chart

For the course attribute similarity matrix, the system directly inputs the user's search data. It then extracts features to obtain the underlying course attribute data. Simultaneously, it consults the user interest model to determine the user's specific course attribute weights. By expertly fusing these similarity calculations and attribute weights, the system constructs the course attribute similarity matrix. The entire course recommendation process, including the deployment of these algorithms, is visually depicted in Figure 5.

5. DEEP NEURAL NETWORK RECOMMENDATION MODEL

The recommendation system based on deep learning utilizes the raw data information of users and items as entries for the input layer [32]. In the hidden layer, a neural network model is employed to learn and extract implicit features of items and users [33]. Finally, the learned latent representations are used to make recommendations for both items and users. Figure 6 illustrates the deep neural network recommendation model, consisting mainly of two collaborative filtering methods: candidate generation network fusion and ranking network [34]. The framework describes a deep-learning approach for generating personalized recommendations. It consists of the following main stages. First, the raw data is input and personalized candidate DNN is generated through collaborative filtering methods. Second, hundreds of potentially relevant items are retrieved from a dataset based on user profiles and past interactions. Then, a deep neural network trained on user data refines this selection to a dozen top recommendations, taking into account additional item characteristics and predicted user preferences. Finally, this personalized list is ultimately seen by the user as a recommendation suggestion.

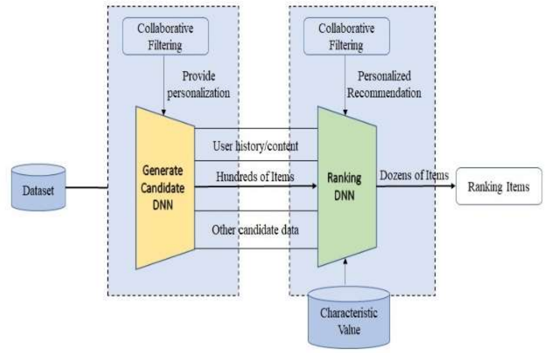

Figure 6: Deep neural network recommendation model

In the candidate generation phase, features extracted from the user's browsing history are used as input to generate a candidate set. This set is created based on multi-source databases and user-related datasets. Collaborative filtering [35] is then applied to achieve extensive personalization. Subsequently, various features of items and users are used to calculate similarity, enabling the minimum ranking level to be determined for recommendation based on collaborative filtering [36].

A neural network algorithm is employed to extract multiple keywords, storing them in a sequential manner where each course corresponds to multiple feature keywords [37]. User preferences for different feature keywords are obtained from their browsing records, and the system calculates weights based on the user's historical browsing of course keywords, aggregating the top 10 keywords [38]-[39]. Using this information, the system calculates the similarity of course descriptions. In Equation (1) assuming the course collection is denoted as C, with the kth course C_k as and its keywords as E_k, the ith keyword as E_i^k, and the keyword weight as W_i^k, the formula for calculating the similarity between course 1 and course 2 descriptions is given by $S_1\ (C_1, C_2)$.

$$S_1\ (C_1, C_2) = \cos\theta = \frac{\sum_{i=1}^{n} w_i^1 \cdot w_i^2}{\sqrt{\sum_{i=1}^{n}(w_i^1)^2 \cdot \sum_{i=1}^{n}(w_i^2)^2}} \qquad (1)$$

This study leverages user attribute retrieval records to capture their preferences for various course attributes. It employs the window control method to analyze course attribute frequencies, distinguishing between short-term and long-term user interests. Subsequently, the user interest model incorporates the basic attribute weight of course attribute similarity to quantify the resemblance between fundamental course attributes.

6. EXPERIMENTAL AND ANALYSIS

This section presents the empirical evaluation of our proposed personalized course recommendation system, OORSDNN. To assess its effectiveness, we conducted an analysis on a real-world dataset.

6.1 Dataset

To validate the accuracy of the algorithmic processing, we collected data within the context of online education networks. Specifically, course data was sourced from the academic management platform of Baoshan University in Baoshan City, Yunnan Province, China. After preprocessing, the dataset retained six main attributes: users, concepts, courses, course categories, course levels, and course evaluations. The dataset encompasses a total of 3,012 users, 11,037 concepts, 534 courses, 6 course categories, four course levels (excellent, good, medium, and poor), and 10,380 course evaluations. A detailed description is provided in Table 1.

Table 1: Online course's dataset for our research

Entities	Statistics	Entities	Statistics
Users	3,012	Course Categories	6
Concepts	11,037	Course Level	4
Courses	534	Course Evaluations	10,380

6.2 Evaluation Metrics

Motivated by the real-world scenario where learners [40]-[41] prefer ranked lists of courses, we follow the established practice in and evaluate all methods using the widely used metrics [42]-[43] of Precision (Prec@N) and Recall (Rec@N). These metrics assess the quality of Top-N recommendations, reflecting the realistic expectation of learners. In Equation (2) and (3), we denote the list of Top-N predicted items as $\hat{R}_{1:N}$ and the set of all possible items as R. Precision@N and Recall@N are then calculated as follows:

$$\text{Prec@N} = \frac{|R \cap \hat{R}_{1:N}|}{N} \tag{2}$$

$$\text{Rec@N} = \frac{|R \cap \hat{R}_{1:N}|}{|R|} \tag{3}$$

6.3 Baseline

To demonstrate the performance of the model, we selected several state-of-the-art models as baselines to compare with our model.

- **POP** [44]: A popularity-based recommendation method where items are ranked by their overall popularity among all users, determined by the number of interactions received.
- **SVD** [45]: An algorithm relying on the Singular Value Decomposition method, decomposing the rating matrix for subsequent predictions.
- **DsRec** [46]: A hybrid model integrating matrix factorization with the basis clustering model to enhance prediction accuracy.
- **BPR** [47]: A non-sequential recommendation method combining matrix factorization and Bayesian inference.
- **GRU4Rec** [48]: A session-based recommendation model utilizing recurrent neural networks to capture sequential dependencies for personalized predictions.

Result and analysis similar to previous studies [49], we assess the performance of our model using Recall@N and Precision@N. Table 2 presents performance metrics, specifically Recall (Rec) and Precision (Prec), for various recommendation methods at different cut-off points (5, 10, 15, and 20). The methods compared are POP, SVD, DsRec, BPR, GRU4Rec, and OORSDNN labeled as "ours." In Table 2, bold fonts show the best performance results, underline representation of the second-best results, and red fonts show the improvement from the best results to the second-best results.

Table 2: The recommendation accuracies of method measured by Recall and Precision

Methods	Rec@5	Rec@10	Rec@15	Rec@20	Prec@5	Prec@10	Prec@15	Prec@20
POP	2.11%	3.99%	5.15%	8.62%	11.98%	12.17%	10.42%	9.27%
SVD	3.04%	5.87%	7.23%	9.46%	14.32%	15.11%	13.26%	11.78%
DsRec	4.59%	5.99%	8.08%	9.82%	15.22%	16.25%	14.32%	12.77%
BPR	5.54%	8.65%	10.73%	14.58%	20.32%	17.99%	16.16%	15.55%
GRU4Rec	5.58%	8.56%	11.29%	16.31%	27.03%	24.18%	22.41%	20.53%
OORSDNN	**6.08%**	**9.57%**	**12.94%**	**18.23%**	**28.22%**	**26.16%**	**24.64%**	**22.36%**
Improve	**8.96%**	**11.80%**	**20.48%**	**10.53%**	**4.22%**	**7.57%**	**9.05%**	**8.18%**

Firstly, compare our method OORSDNN with three traditional methods POP, SVD, and DsRec. POP is based on item popularity, SVD is based on matrix factorization, and DsRec is a hybrid method that combines matrix factorization and clustering. POP focuses on item popularity, while SVD and DsRec focus on user interest. The experimental results in Table 2 show that POP, which focuses on item popularity, has the worst performance in terms of recall and precision (k=5,10,15,20), with values of 2.11%, 3.99%, 5.15%, and 8.62% for recall and 11.98%, 12.17%, 10.42%, and 9.27% for precision. DsRec performs best among the three traditional methods, with SVD in the middle. By comparing the three traditional methods, we find that matrix factorization-based recommendation methods outperform item popularity-based recommendation methods.

In a direct comparison with traditional baselines POP, SVD, and DsRec, our method consistently outperformed them in terms of Recall@k and Precision@k. This demonstrates the clear advantage of deep learning-based methods over both item popularity and matrix factorization approaches. Our model's ability to capture relevant items and deliver more personalized and accurate recommendations, as evidenced by the significant improvements in both recall and precision, highlights its superiority in enhancing recommendation quality compared to simply relying on popularity.

As shown in Table 2, BPR and GRU4Rec are currently popular benchmarks, and they serve as effective tools for assessing the performance of models. BPR is a collaborative filtering method that focuses on pairwise ranking. It aims to optimize the ranking of items based on users' preferences by considering their interactions with items. GRU4Rec is a recommendation model based on Gated Recurrent Units (GRUs), which are a type of recurrent neural network. This model is designed to capture sequential patterns in user-item interactions, considering the temporal aspects of user behavior. Compared to BPR and GRU4Rec, we observe that the GRU4Rec model performs slightly lower than BPR in REC@10 but outperforms BPR in other metrics. When contrasting our model with the second-best performing model, it is evident that our model improves the results significantly. Specifically, at Rec@5, our model enhances the performance from 5.58% to 6.08%, with an improvement rate of 8.96%. At Rec@10, Rec@15, and Rec@20, there are improvement rates of 11.80%, 20.4%, and 10.53%, respectively. Comparisons in Pre@k also reveal improvements ranging from 4.22% to 9.05%. Finally, as depicted in Figure 7, it is noteworthy that our model achieves the maximum improvement in both recall and precision when K equals 15, which are 20.48% and 9.05%, respectively. Subsequently, it becomes apparent that as the value of K increases, the results gradually deteriorate. As a result, we can choose k=15 for the top-k ranking number for prediction.

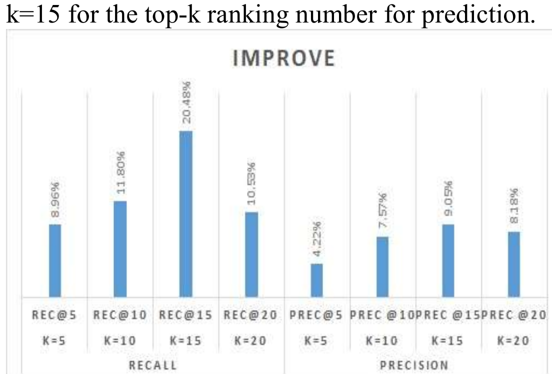

Figure 7: The improvement results of our model in recall and precision

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

7. FUTURE WORK

Building on the strong foundation of our object-oriented recommendation system, several exciting avenues beckon for future exploration:

One promising direction is focus on investigating the impact of diverse user feedback mechanisms on the recommendation system. Exploring how explicit user feedback, such as ratings and reviews, can be integrated into the algorithm would contribute to refining the system's understanding of individual preferences. Additionally, the incorporation of implicit feedback, such as user engagement patterns and completion rates, could further enhance the depth of the recommendation algorithm, providing more nuanced and context-aware suggestions. This exciting direction involves exploring context-aware recommendations. This approach would consider real-time factors like student goals, learning progress, and even external events to deliver dynamic and timely suggestions. For example, if a student is struggling with a specific concept, the system could recommend relevant supplemental materials or suggest alternative courses that address the same topic from a different perspective. This dynamic adaptation can significantly improve the relevance and effectiveness of recommendations.

Another avenue for future research lies in the continuous refinement of the deep neural network algorithm used for course recommendation. Specifically, focusing on enhancing the algorithm's interpretability and transparency could address concerns related to the "black-box" nature of deep learning models. Exploring methods to provide users with understandable explanations for the system's recommendations would contribute to building trust and user confidence, ultimately fostering a more transparent and user-friendly recommendation process.

Finally, ensuring system scalability is crucial for widespread adoption. Future work can explore distributed computing platforms and cloud infrastructure to efficiently handle large datasets and user bases. Additionally, developing real-time recommendation engines can significantly enhance user experience and responsiveness, especially for platforms with a large number of active users. By addressing these scalability concerns, we can ensure that our object-oriented online course recommendation system remains a viable and impactful solution for personalized learning in the ever-growing landscape of online education.

8. CONCLUSION

In conclusion, this paper pioneers a hybrid online course recommendation system that marries object-oriented design for modularity and deep learning for personalization. This novel approach delivers tailored learning experiences, adapting to user needs and ensuring long-term scalability for widespread adoption. The research emphasizes the importance of considering system maintainability and scalability, often overlooked in the literature, and successfully demonstrates the practical implementation of design principles for modularity, maintainability, and extensibility. The evaluation of the proposed system on real-world datasets showcases its effectiveness in providing accurate and personalized course recommendations, thereby contributing to an enriched learning experience for students. The comprehensive analysis and design principles laid out in this work offer valuable insights for future developments in the field of online learning platforms, highlighting the significance of both technological sophistication and thoughtful system design.

9. ACKNOWLEDGEMENT

This study was supported by Yunnan Province Local Universities Joint Special Youth Project(202101BA070001-270) and Teaching Quality and Teaching Reform Project of Baoshan University in 2022-2023(ZHP202344).

REFERENCES:

[1] I.-H. Jo, Y. Park, J. Kim, and J. Song, “Analysis of Online Behavior and Prediction of Learning Performance in Blended Learning Environments,” *Educ. Technol. Int.*, vol. 15, no. 2, pp. 71–88, 2014.
[2] A. Beutel *et al.*, “Latent cross: Making use of context in recurrent recommender systems,” *WSDM 2018 - Proc. 11th ACM Int. Conf. Web Search Data Min.*, vol. 2018-Febua, pp. 46–54, 2018, doi: 10.1145/3159652.3159727.
[3] R. Wirfs-Brock and B. Wilkerson, “Object-oriented design: A responsibility-driven approach,” *ACM SIGPLAN Not.*, vol. 24, no. 10, pp. 71–75, 1989, doi: 10.1145/74878.74885.
[4] I. Endres and D. Hoiem, “eccv2010--Category Independent Object Proposals.pdf.”
[5] A. Kaur and G. Dhiman, “A review on search-based tools and techniques to identify bad code smells in object-oriented systems,” in *Advances in Intelligent Systems and Computing*, 2019, vol. 741, doi: 10.1007/978-981-13-0761-4_86.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[6] T. B. Lalitha and P. S. Sreeja, "Personalised Self-Directed Learning Recommendation System," in *Procedia Computer Science*, 2020, vol. 171, doi: 10.1016/j.procs.2020.04.063.

[7] O. Malgonde, H. Zhang, B. Padmanabhan, and M. Limayem, "Taming complexity in search matching: Two-sided recommender systems on digital platforms," *MIS Q. Manag. Inf. Syst.*, vol. 44, no. 1, 2020, doi: 10.25300/MISQ/2020/14424.

[8] S. B. Aher and L. M. R. J. Lobo, "Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data," *Knowledge-Based Syst.*, vol. 51, 2013, doi: 10.1016/j.knosys.2013.04.015.

[9] M. Hikmatyar and Ruuhwan, "Book Recommendation System Development Using User-Based Collaborative Filtering," in *Journal of Physics: Conference Series*, 2020, vol. 1477, no. 3, doi: 10.1088/1742-6596/1477/3/032024.

[10] H. T. Cheng *et al.*, "Wide & deep learning for recommender systems," *ACM Int. Conf. Proceeding Ser.*, vol. 15-Septemb, no. March 2020, pp. 7–10, 2016, doi: 10.1145/2988450.2988454.

[11] J. Vatter, R. Mayer, and H. A. Jacobsen, "The Evolution of Distributed Systems for Graph Neural Networks and Their Origin in Graph Processing and Deep Learning: A Survey," *ACM Comput. Surv.*, vol. 56, no. 1, 2023, doi: 10.1145/3597428.

[12] C. Chen, M. Zhang, Y. Zhang, W. Ma, Y. Liu, and S. Ma, "Efficient Heterogeneous Collaborative Filtering without negative sampling for recommendation," 2020, doi: 10.1609/aaai.v34i01.5329.

[13] F. Ullah *et al.*, "Deep Edu: A Deep Neural Collaborative Filtering for Educational Services Recommendation," *IEEE Access*, vol. 8, pp. 110915–110928, 2020, doi: 10.1109/ACCESS.2020.3002544.

[14] Y. Gao, Z. W. Huang, Z. Y. Huang, L. Huang, Y. Kuang, and X. Yang, "Multi-scale broad collaborative filtering for personalized recommendation," *Knowledge-Based Syst.*, vol. 278, 2023, doi: 10.1016/j.knosys.2023.110853.

[15] C. D. Wang, W. D. Xi, L. Huang, Y. Y. Zheng, Z. Y. Hu, and J. H. Lai, "A BP Neural Network Based Recommender Framework With Attention Mechanism," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 7, 2022, doi: 10.1109/TKDE.2020.3023976.

[16] A. Shewalkar, D. nyavanandi, and S. A. Ludwig, "Performance Evaluation of Deep neural networks Applied to Speech Recognition: Rnn, LSTM and GRU," *J. Artif. Intell. Soft Comput. Res.*, vol. 9, no. 4, 2019, doi: 10.2478/jaiscr-2019-0006.

[17] L. Zhang, T. Luo, F. Zhang, and Y. Wu, "A Recommendation Model Based on Deep Neural Network," *IEEE Access*, vol. 6, 2018, doi: 10.1109/ACCESS.2018.2789866.

[18] H. Luo and Z. Li, "Research on Construction of Recommendation System on account of CNN and PMF model," in *Journal of Physics: Conference Series*, 2021, vol. 1992, no. 3, doi: 10.1088/1742-6596/1992/3/032078.

[19] J. A. H. López, J. L. Cánovas Izquierdo, and J. S. Cuadrado, "ModelSet: A labelled dataset of software models for machine learning," *Sci. Comput. Program.*, vol. 231, 2024, doi: 10.1016/j.scico.2023.103009.

[20] M. Laghari and A. Hassan, "A Software System for Smart Course Planning," in *Communications in Computer and Information Science*, 2023, vol. 1788 CCIS, doi: 10.1007/978-3-031-27609-5_32.

[21] K. Vlasenko, S. Volkov, I. Sitak, I. Lovianova, and D. Bobyliev, "Usability analysis of on-line educational courses on the platform 'higher school mathematics teacher,'" in *E3S Web of Conferences*, 2020, vol. 166, doi: 10.1051/e3sconf/202016610012.

[22] W. Jiang, Z. A. Pardos, and Q. Wei, "Goal-based course recommendation," 2019, doi: 10.1145/3303772.3303814.

[23] K. Julia, V. R. Peter, and K. Marco, "Educational scalability in MOOCs: Analysing instructional designs to find best practices," *Comput. Educ.*, vol. 161, 2021, doi: 10.1016/j.compedu.2020.104054.

[24] T. Guedes, L. A. Jesus, K. A. C. S. Ocaña, L. M. A. Drummond, and D. de Oliveira, "Provenance-based fault tolerance technique recommendation for cloud-based scientific workflows: a practical approach," *Cluster Comput.*, vol. 23, no. 1, 2020, doi: 10.1007/s10586-019-02920-6.

[25] X. Zhou *et al.*, "Latent error prediction and fault localization for microservice applications by learning from system trace logs," 2019, doi: 10.1145/3338906.3338961.

[26] A. Kumar, S. K. Pandey, S. Prakash, K. U. Singh, T. Singh, and G. Kumar, "Enhancing Web Application Efficiency: Exploring Modern Design Patterns within the MVC Framework," 2023, doi:

10.1109/CISES58720.2023.10183582.

[27] D. Mazinanian and N. Tsantalis, “An empirical study on the use of CSS preprocessors,” in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering, SANER 2016*, 2016, vol. 1, doi: 10.1109/SANER.2016.18.

[28] A. Amarulloh, “Analisis Perbandingan Performa Web Service Rest Menggunakan Framework Laravel, Django, dan Node JS Pada Aplikasi Berbasis Website,” *J. Tek. Inform. STMIK Antar Bangsa*, vol. 09, no. 01, 2023.

[29] L. Chazette, W. Brunotte, and T. Speith, “Explainable software systems: from requirements analysis to system evaluation,” *Requir. Eng.*, vol. 27, no. 4, 2022, doi: 10.1007/s00766-022-00393-5.

[30] N. K. Maina, G. M. Muketha, and G. M. Wambugu, “A New Complexity Metric for UML Sequence Diagrams,” *Int. J. Softw. Eng. Appl.*, vol. 14, no. 01, 2023, doi: 10.5121/ijsea.2023.14102.

[31] H. Luo, N. A. Husin, and T. N. M. Aris, “ROME: A Graph Contrastive Multi-View Framework From Hyperbolic Angular Space for MOOCs Recommendation,” *IEEE Access*, vol. 11, 2023, doi: 10.1109/ACCESS.2022.3232552.

[32] Z. Fu, X. Niu, and M. Lou Maher, “Deep Learning Models for Serendipity Recommendations: A Survey and New Perspectives,” *ACM Comput. Surv.*, vol. 56, no. 1, 2023, doi: 10.1145/3605145.

[33] H. Liu, Y. Zhang, P. Li, C. Qian, P. Zhao, and X. Wu, “DeepCPR: Deep Path Reasoning Using Sequence of User-Preferred Attributes for Conversational Recommendation,” *ACM Trans. Knowl. Discov. Data*, vol. 18, no. 1, 2023, doi: 10.1145/3610775.

[34] K. R, P. Kumar, and B. Bhasker, “DNNRec: A novel deep learning based hybrid recommender system,” *Expert Syst. Appl.*, vol. 144, 2020, doi: 10.1016/j.eswa.2019.113054.

[35] L. Zhu, J. Li, and W. Guan, “Multi-modal Discrete Collaborative Filtering,” in *Synthesis Lectures on Information Concepts, Retrieval, and Services*, vol. Part F1232, 2024.

[36] X. Luo *et al.*, “Criterion-based Heterogeneous Collaborative Filtering for Multi-behavior Implicit Recommendation,” *ACM Trans. Knowl. Discov. Data*, vol. 18, no. 1, 2023, doi: 10.1145/3611310.

[37] D. Alahmadi and F. Alruwaili, “Deep Learning for MOOCs Course Recommendation Systems: State of the Art Survey,” *Int. Trans. J. Eng.*, vol. 12, no. 11, 2021.

[38] P. Bhuvaneshwari, A. N. Rao, and Y. H. Robinson, “Correction to: Top-N Recommendation System Using Explicit Feedback and Outer Product Based Residual CNN,” *Wirel. Pers. Commun.*, vol. 128, no. 2, 2023, doi: 10.1007/s11277-022-10055-y.

[39] V. Coscrato and D. Bridge, “Estimating and Evaluating the Uncertainty of Rating Predictions and Top-n Recommendations in Recommender Systems,” *ACM Trans. Recomm. Syst.*, vol. 1, no. 2, 2023, doi: 10.1145/3584021.

[40] A. Li, B. Yang, H. Huo, H. Chen, G. Xu, and Z. Wang, “Hyperbolic Neural Collaborative Recommender,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, 2023, doi: 10.1109/TKDE.2022.3221386.

[41] X. Wang and S. Kadıoğlu, “Modeling uncertainty to improve personalized recommendations via Bayesian deep learning,” *Int. J. Data Sci. Anal.*, vol. 16, no. 2, 2023, doi: 10.1007/s41060-020-00241-1.

[42] M. Rostami, V. Farrahi, S. Ahmadian, S. Mohammad Jafar Jalali, and M. Oussalah, “A novel healthy and time-aware food recommender system using attributed community detection,” *Expert Syst. Appl.*, vol. 221, 2023, doi: 10.1016/j.eswa.2023.119719.

[43] Z. Shokrzadeh, M.-R. Feizi-Derakhshi, M.-A. Balafar, and J. Bagherzadeh Mohasefi, “Graph-Based Recommendation System Enhanced by Community Detection,” *Sci. Program.*, vol. 2023, 2023, doi: 10.1155/2023/5073769.

[44] D. Massimo and F. Ricci, “Popularity, novelty and relevance in point of interest recommendation: an experimental analysis,” *Inf. Technol. Tour.*, vol. 23, no. 4, 2021, doi: 10.1007/s40558-021-00214-5.

[45] F. Colace, D. Conte, M. De Santo, M. Lombardi, D. Santaniello, and C. Valentino, “A content-based recommendation approach based on singular value decomposition,” *Conn. Sci.*, vol. 34, no. 1, 2022, doi: 10.1080/09540091.2022.2106943.

[46] J. Liu, Y. Jiang, Z. Li, X. Zhang, and H. Lu, “Domain-Sensitive Recommendation with User-Item Subgroup Analysis,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 4. 2016, doi: 10.1109/TKDE.2015.2492540.

[47] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," 2009.

[48] Di. Jannach and M. Ludewig, "When recurrent neural networks meet the neighborhood for session-based recommendation," 2017, doi: 10.1145/3109859.3109872.

[49] J. Tang and K. Wang, "Personalized top-N sequential recommendation via convolutional sequence embedding," in *WSDM 2018 - Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, 2018, vol. 2018-Febuary, doi: 10.1145/3159652.3159656.