International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7395
***Article Received**: 01 May 2023 **Revised:** 29 June 2023 **Accepted:** 21 July 2023*

Graph Neural Networks for E-Learning Recommendation Systems

Mr. Venkata Bhanu Prasad Tolety[1*], Prof. Evani Venkateswara Prasad[2]
[1*]Research Scholar, Department of Computer Science and Engineering,
Jawaharlal Nehru Technological University,
Kakinada, India
[1*]Email: tvbphyd@gmail.com
[2]Professor, Department of Computer Science and Engineering,
Gayatri Vidya Parishad College of Engineering for Women,
Visakhapatnam, India
[2]Email: profevprasad@yahoo.com

Abstract—This paper presents a novel recommendation system for e-learning platforms. Recent years have seen the emergence of graph neural networks (GNNs) for learning representations over graph-structured data. Due to their promising performance in semi-supervised learning over graphs and in recommendation systems, we employ them in e-learning platforms for user profiling and content profiling. Affinity graphs between users and learning resources are constructed in this study, and GNNs are employed to generate recommendations over these affinity graphs. In the context of e-learning, our proposed approach outperforms multiple different content-based and collaborative filtering baselines.

Keywords-Recommender System; GNN; collaborative filtering; content-based filtering.

I. INTRODUCTION

Today, recommender systems are ubiquitous [1]. Numerous applications, ranging from e-commerce to movie recommendations, to advertising, to social media, employ recommender systems in one form or the other. This evolution of recommender systems is a key enabler for several web applications. Today, several recommender systems [1, 2, 3, 4] can be tailored to a particular application without much hassle. However, just as the *no-free-lunch* theorem suggests, no "single" recommender system can be employed without modification to several applications. Each application domain warrants its design for recommender systems that takes into account domain-specific requirements/biases. It is quite evident by observing how tech giants like Amazon [3], Netflix [5], Google [6], etc., invest large amounts of resources in researching recommender systems tailored to their business needs.

In this paper, we tackle recommender systems' design/adaptation for generating recommendations for e-learning platforms in a similar vein. With the advent and subsequent commercialization of several e-learning platforms such as Coursera, edX, Udacity, etc., there has been significant interest in AI-assisted tools for e-learning, which would greatly reduce the amount of human intervention required to support such platforms and allow operations to scale while minimizing operational costs. However, these benefits come at a cost: operators of the e-learning platform now have lesser control over each step in this process. With larger volumes of students flocking to e-learning platforms (more so given the current COVID-19 situation), manually tracking the progress of an exorbitantly increasing number of online learners is a challenging task. We aim to mitigate this hurdle by developing a recommender system capable of "altering" a learner's curriculum by keeping track of the learner's performance and using it to determine the set of modules that the learner must revisit or proceed.

Over recent years, much of machine learning research has been dominated by the resurgence of deep neural networks [7]. Neural networks are a class of machine learning algorithms that are composed of several individual units (also called "neurons") that perform "linear"/" affine" computations, interspersed with nonlinear "activation" functions (such as sigmoid, relu, etc.). "Deep" neural networks refer to the process of stacking several such neural network "layers", which helps in the learning of arbitrarily complex functions due to the universal function approximation theorem [8]. Several variants of neural networks have been proposed, based on the type of data they operate. For example, Convolutional Neural Networks (CNNs) [9] have demonstrated state-of-the-art performance on image-based tasks (such as identifying what content an image has, etc.). Similarly, for structured/relational data, standard (feed forward) neural networks are not suitable

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7395
Article Received**: 01 May 2023* ***Revised: *29 June 2023* ***Accepted:*** *21 July 2023*

(neurons in such neural networks are incapable of *explicitly* modeling "relations" among variables).

For explicit relational modeling (e.g., in scenarios where a relation R relates a variable x and another variable y, usually written as xRy, and read as "x is related to y by relation R"), recently, graph neural networks have been proposed [10, 11, 12, 13]. Graph Neural Networks (GNNs) have been shown to explicitly model relations among variables and have been successfully employed in a few recommender systems in other domains (like social networks [14]). In a nutshell, graph neural networks employ a message passing scheme where input feature representations at each node in the graph are transmitted to the neighboring nodes in the graph (as "messages"), and these "messages" are "aggregated" to compute updated/new feature representations at each node. By repetitive applications of the message passing and aggregation routines, each node can learn sufficiently complex high-level feature representations suited for real-world tasks such as recommender systems.

To the best of our knowledge, this study presents the first GNN-based recommender systems for e-learning applications. We show that our GNN-based recommender system outperforms previous content-based and collaborative filtering-based recommender systems.

The rest of the paper is structured as follows. The second section examines related research in recommender systems and graph neural networks. Part III provides a quick overview of graph neural networks. The suggested GNN-based recommender system is presented in Section IV.

Part V reports on the results of this system's evaluation on a real-world dataset. Part VI brings the paper to a close by outlining some intriguing potential directions.

II. RELATED WORK

Sunil and Doja [15] proposed a recommendation system for e-learning environments to improve learners' ability to learn based on their learning style and knowledge level, taking into account learners' learning activities and Neil's VARK (visual, auditory, reading/writing, and kinaesthetic) learning questionnaires. The system then proposes the next course to the student based on the results of the profile and the students' review, as well as the area that the content developers should focus on based on the students' learning styles.

Bhaskaran and Marappan [16] has described some of the innovative tactics that are being tested to increase the efficacy of a hybrid recommender. The updated one-source denoising strategy is intended to pre-process the student data set. The Anarchic Society Optimization technique has been updated to increase performance metrics. To extract the learners' sequential patterns, an enhanced and broader sequential pattern technique is provided. The Improved Transductive Support Vector Machine (ETSVM) was created to rate extracted habits and interests. These new tactics analyse student confidentiality rates and provide the best recommendation to pupils. The proposed generalized model is simulated on public machine learning datasets, e.g. B. Movies, music, books, groceries, goods, health care, appointments, academic papers, and learning recommendations for open universities. According to the experimental results, the better clustering technique discovers clusters based on random size. In terms of predicted absolute error, precision, classification score, recall, and precision metrics, the proposed recommendation techniques outperform the methodologies.

Tahir et al. [17] proposed a system (Dynamic Recommendation of Filtered LOs (DRFLO)) for automatically extracting and classifying a collection of semantically relevant LOs from heterogeneous LORs. LOs are contextually categorised using the extra recommendation and classification models. Finally, when building a course, a rating of LOs is recommended with a simple consultation with the course designer based on learning preferences. In terms of precision, recall, and F-measurement, the proposed model performs admirably. Furthermore, the test results were compared to a common search engine, and the accuracy is likewise good.

In [18], Venkata Bhanu Prasad Tolety and Evani Venkateswara Prasad created a hybrid recommendation system that combines the desirable aspects of collaborative filtering and content-based filtering for the task of recommending course content/curriculum to students, users of an e-learning system. This recommendation combines easily changing user profiles (as learners move through course content) with generalisation across content sources (courses offered by several departments) and categories. The approach is applied to a real-world data collection with 111 students organised into multidisciplinary Groups. The results suggest that the proposed hybrid recommendation system has clear advantages, outperforming classic filtering strategies by more than 30 percentage points.

Murad and Yang [19] proposed a framework for the development of a personalised eLearning recommendation system that combines the techniques of student profiling, knowledge estimation, assessment, and feedback to improve student learning through the recommendation of online video-based learning materials to improve the student's profile and knowledge.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7395
***Article Received**: 01 May 2023 **Revised:** 29 June 2023 **Accepted:** 21 July 2023*

III. GNNS: RECAP

In this section, we review GNNs and present the mathematical formulation of underpinning GNNs. Formally, a GNN is a neural network architecture tailored to graph-structured data. As one can recall, a graph G is a set of vertices/nodes V, and a set of edges E. Assume that V = {v_1, v_2, …, v_n} is the set of n vertices of the graph G. Also, let E = {e_1, e_2, e_K} be the set of edges of the graph G, where each edge e_k connects a vertex v{i_k} to a vertex v_{j_k}, where i_k and j_k are both in the range [1, n].

A standard feed forward neural network (also called a multilayer perceptron (MLP)) consists of several "layers" of "neurons" interspersed with nonlinearities. At each layer, the input x is transformed into an output y by applying an affine transformation, followed by the nonlinearity and is mathematically represented as:

$$\psi = (\Omega_{\xi} + \beta) \tag{1}$$

where y is a nonlinearity (usually sigmoid, tanh, relu, etc. are used), x is the input to the layer, W is the (learnable) weight matrix of the neurons in the layer, and b is a (learnable) "bias" vector. For an MLP with N layers, N such operations are applied in sequence, i.e.,

$$\psi_{\nu} = \nu(\Omega_{\nu\xi\nu}{-1} + \beta_{\nu}), \ \omega\eta\epsilon\rho\epsilon\ \psi_{0} = \xi \tag{2}$$

An MLP necessitates the inputs and outputs' dimensions to be known beforehand, which is not true for graph-structured data. We want to handle a varying number of nodes at train/test time for graph data and across multiple runs of the GNN (e.g. courses/users are added/deleted). Further, MLPs cannot "explicitly" capture relations in data. Hence, we resort to using GNNs.

Formally speaking, a GNN layer comprises a "message passing" and an aggregation phase. During the message passing phase, each node vi transmits a "message"/" feature" to each of its neighboring nodes v_jNbd (v_i). Mathematically, the message passing phase can be represented as

$$\mu_{\iota}, \tau{+}1 = \mu_{\iota}, \tau + \varphi\ N\beta\delta(\varpi_{\iota})\mu_{\varphi}, \tau \tag{3}$$

here m_i, t denotes the message that originated at node i at time t, and mj, t denotes the messages originating at the neighbors of node i at time t. So, m_i, t+1, the message at node i at time t+1 is computed as the algebraic sum of the message at the node iiteself at time t, and all the messages originating from its neighboring nodes j Nbd(v_i) at time t.

The second phase of a GNN layer is "message aggregation". In this phase, the messages are aggregated in a permutation-invariant manner (i.e., the number and order of neighboring nodes do not influence the function's output). Mathematically, the message aggregation phase can be represented as

$$\mu_{\iota}, \tau{+}1 = (\Omega\ (\mu_{\iota}, \tau{+}1) + \beta) \tag{4}$$

here is a permutation invariant pooling operator (e.g., max, min, average, etc.), and (W, b) denote the weight matrix and bias vector of a standard MLP layer, respectively. Also, it is nonlinearity (typically sigmoid, tanh, relu, etc.). This procedure of combining a standard MLP with a message passing and aggregation scheme allows representation learning on graph-structured data while explicitly reasoning about relations among variables/attributes being modeled.

Typically, as with MLPs, several such GNN "layers" are stacked together to allow for representation learning for arbitrarily complex distributions of data. For a more in-depth explanation of GNNs, we refer the interested reader to [10, 11, 12, 13].

IV. GNN-BASED RECOMMENDER SYSTEM

A new recommender system for e-learning platforms is presented in this paper, incorporating recent research on graph-neural networks (GNNs). In order to understand what follows, we must first introduce the terminology and mathematical notation used throughout.

Let U = {u_1, u_2, ...,u_N}be the set of users in the e-learning platform, and let M = {m_1, m_2, ..., m_P} be the set of “modules”. Note that it is not necessarily equal to P (N, P). By "modules", we denote the set of e-learning modules that are open to recommendation. The platform administrator can choose for the set of modules to change dynamically (e.g., when newer course offerings are added to the platform or expired course offerings are removed).

We denote C as the user-module compatibility matrix. C is an n x p matrix (n rows, p columns), where each entry c_{ij} denotes the "compatibility of user u_i to module m_j". Due to cold-start problems, or simply because a module/user is new, most of the entries c_{ij} of the compatibility matrix C might not be available. All such "unknown" compatibilities are initialized to 0 by default (the task for the GNN will be to predict the compatibility scores for these edges, so initializing to zeros is a sensible assumption).

Let KNOWN = {(u_i, m_j) | c_{ij}0} be the set of "known compatibilities", i.e., the (user, module) tuples for which we know compatibility scores. Similarly let KNOWN = {(u_i, m_j) | c_{ij} = 0} be the set of "unknown compatibilities", i.e., the (user, module) tuples for which we do not know compatibility scores.

A. User-Module graph construction

The central entity in our GNN-based recommender system is what we call the "user-module compatibility graph". As one might recall, a graph G = (V, E) is a set of "vertices" / "nodes" and a set of "edges" where each edge connects two vertices/nodes. For our purposes, we assume an undirected graph (i.e., an edge (a, b) implies that a is connected to b, and that b is connected to a).

The user-module graph nodes are "users" U and "modules" M. This is a fully-connected bipartite graph where every user u_i is connected to every module m_j. Each edge e_{ij} connects u_i to m_j. The weight of the edge is given by looking up the corresponding index c_{ij} in the compatibility matrix C.

With the above terminology defined, we now proceed to outline our proposed method. It comprises three phases, namely,

1. User modeling
2. Content modeling, and
3. Compatibility prediction

In the subsections that follow, we present each of the phases in detail.

B. User modeling

The first step of our pipeline is user modeling. User modeling aims to learn features that help understand users and their preferences better. In this phase, we use a graph neural network over a subgraph of the compatibility graph, containing only one user. We consider the subgraph where only one user u_i is connected to all modules m_j (j {1...p}). We apply a GNN, referred to as the "UserGNN," to this graph to learn node representations that model a user's preferences. As stipulated in section III, message passing and aggregation are iteratively applied to learn node representations for the user.

C. Content modeling

The second step of our pipeline is content modeling. The aim of this phase is complementary to that of the user modeling phase. In this phase, we aim to learn good feature representations over the "content" (here "modules"). In this phase, we create a subgraph of the compatibility graph, such that each subgraph contains all users but only one unique module. We consider p subgraphs, each containing user nodes u_1, u_2, ..., u_n, and module nodes m_j. We apply another GNN, referred to as the "ContentGNN," to this graph to learn node representations that model each module's features. As described in Section III, learning proceeds by a sequence of message passing and message aggregation steps.

D. Compatibility prediction

In this final step of our pipeline, we use the UserGNN and the ContentGNN in conjunction with a third GNN, which we call the "RecommenderGNN", to predict compatibility scores over the entire graph and generate recommendations.

We use the entire user-module compatibility graph (matrix C defines the graph's adjacency matrix) in this phase.

We precomputed user node features, and modules node features using the UserGNN and the ContentGNN, respectively. The "RecommenderGNN" is applied over the entire compatibility graph to learn a compatibility function. The task for the RecommenderGNN is to predict all compatibilities c_{ij} in the compatibility matrix C.

E. Loss function

The loss function for training the RecommenderGNN is defined by comparing the predicted compatibility values c_{ij} to the true compatibility values c*ij. Since most compatibility values are known ahead-of-time during training, we generate training examples by manually masking out a few of the edges of the graph c_{ij}, and the loss function is the mean squared error between the predicted and true compatibilities. For N training examples,

$$\Lambda = 1N\kappa = 1N(\chi_{\iota}\kappa_{\varphi}\kappa - \chi*\iota\kappa\varphi\kappa)^2 \tag{5}$$

F. Training procedure

We train the RecommenderGNN using the loss function specified above and apply stochastic gradient descent for updating the GNN parameters (weight matrices and bias vectors).

V. EXPERIMENTS AND RESULTS

This section describes the experiments conducted to analyze the proposed GNN-based recommender system's efficacy and enlist our findings therein.

A. Dataset description

We use the "Open University Learning Analytics" dataset (OULAD) [20] for evaluating the proposed approach, as well as many baseline recommender systems. The OULAD is a tabular dataset comprising student analytics gathered for 1 year. The reason for using this dataset is that it contains a very large number of students (32,593) and a large number of modules (22 courses, with each course containing about 20 modules, making it around 880 modules overall).

For each student (whose records are anonymized), OULAD provides several attributes that can be used to inform user profiling or compatibility prediction. For example, some interest fields are students' gender, geographic region, the highest level of education, age band, number of previous attempts, disability (if any), and the student's final result, etc.

For each module, OULAD provides attributes that can be used for content profiling and compatibility prediction. Some fields of interest are: module domain, number of presentations in the module, number of students registered, types of assessment, number of successful students for the module, etc.

For a complete description of the tabular dataset structure, one might refer to [20]. We randomly mask out 20% of the user-module interactions in each experiment below and use the remaining 80% interactions for training. The held-out 20% of data is used for testing. We perform 5-fold cross-validation to ensure there is no hold-out bias in our experiments.

B. Approaches evaluated

In addition to the proposed approach, we evaluate the following recommender systems baselines on the OULAD dataset to demonstrate GNN-based recommender systems' advantages.

1. Content-based filtering: we implement the following content-based filtering approaches for evaluation (we use scikit-learn implementations, with OULAD features).
   a. Item-centered Bayesian classifier
   b. User-centered Linear regression
   c. User-centered decision tree

Collaborative filtering: we implement the following collaborative filtering approaches that regress to compatibility values (c_{ij}). We use the scikit-learn implementations for this as well.

a. Matrix factorization
b. Collaborative filtering using decision trees

GNN-based recommender system: This is the proposed method in this paper (Section IV).

C. Performance Comparision

In the Table 1 shown below, we summarize the above approaches' performance (namely, 1a, 1b, 1c, 2a, 2b, and 3). Note that (3) is the approach proposed in this paper (we denote it as "OURS" in the table). Figure 1 shows performance evaluation of RMSE. Figure 2 Shows performance evaluation of Precision.

TABLE 1 PERFORMANCE EVALUATION USING GNN

Approach	**RMSE (lower is better)**	**Precision (higher is better)**
1a (item-centered Bayesian)	1.51	21%
1b (user-centered regression)	1.48	21%
1c (user-centered DTree)	1.47	21%
2a (matrix factorization)	0.95	52%
2b (collaborative DTree)	0.72	59%
3 (GNN - OURS)	0.19	85%

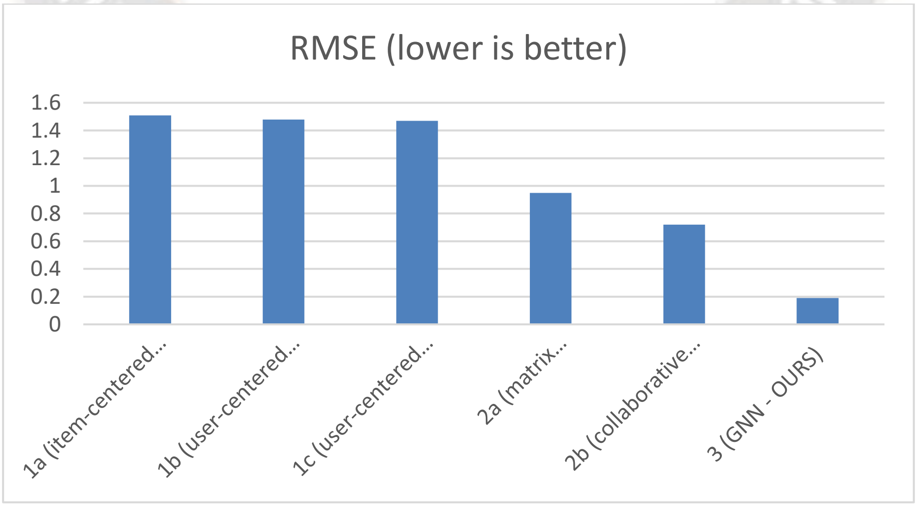

Figure 1. Performance evaluation of RMSE

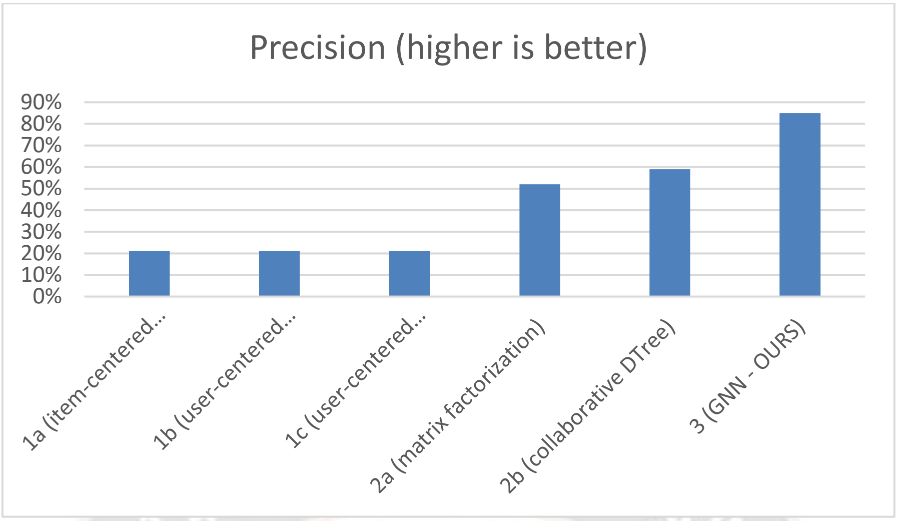

Figure 2. Performance evaluation of Precision

As we can see from the above table, in general, collaborative filtering approaches (2a, 2b) perform much better than the content filtering approaches (1a, 1b, 1c). However, their performance is still below par for the task at hand (see how 2a and 2b get precision values barely above 50%).

However, the GNN based approach (i.e., the UserGNN + ContentGNN + RecommenderGNN) achieves significantly better accuracy than the baselines evaluated. This demonstrates that GNNs are better suited for modeling user-module compatibilities in such high-dimensional spaces, where interactions are relatively sparse.

D. *Ablation analysis*

In this subsection, we consider the question "what is the contribution of the UserGNN and ContentGNN to the overall system?". We conduct an ablation analysis to experiment with 3 variants of our proposed approach to analyze this further.

1. Variant 1: Only RecommenderGNN (no UserGNN, no ContentGNN)
2. Variant 2: UserGNN + RecommenderGNN (no ContentGNN)
3. Variant 3: ContentGNN + RecommenderGNN (no UserGNN)
4. Variant 4: UserGNN + ContentGNN + RecommenderGNN (i.e., proposed method)

We analyze all combinations of the 3 GNNs that make sense. Variant 1 comprises only the RecommenderGNN, which uses raw attributes as input (i.e., does not have features computed by UserGNN or ContentGNN). Variant 2 comprises the UserGNN and the RecommenderGNN but does not have the ContentGNN. Complementarily, Variant 3 comprises the ContentGNN and the RecommenderGNN, but not the UserGNN.

The only other variant possible is Variant 4, which includes the UserGNN, ContentGNN, and the RecommenderGNN. Other variants are not possible, as we must in all cases have the RecommenderGNN (else, the point of having a recommender system is moot if we don't have the RecommenderGNN synthesizing the recommendations).

TABLE 2 ABLATION ANALYSIS

Approach	**RMSE (lower is better)**	**Precision (higher is better)**
Variant 1	0.33	78%
Variant 2	0.27	71%
Variant 3	0.31	76%
Variant 4	**0.19**	**85%**

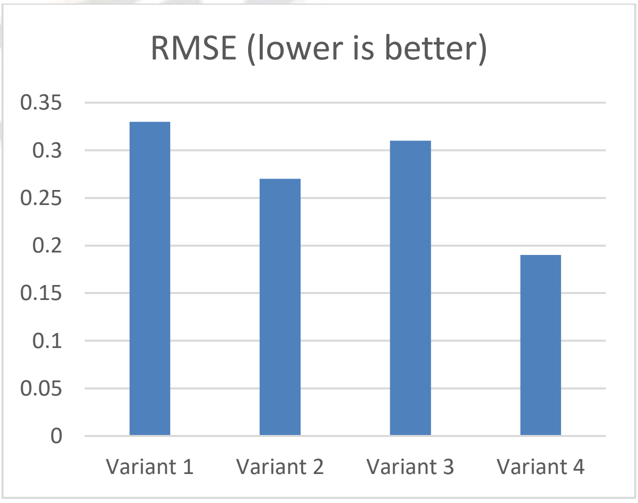

Figure 3. Ablation analysis for RMSE

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7395
***Article Received**: 01 May 2023 **Revised:** 29 June 2023 **Accepted:** 21 July 2023*

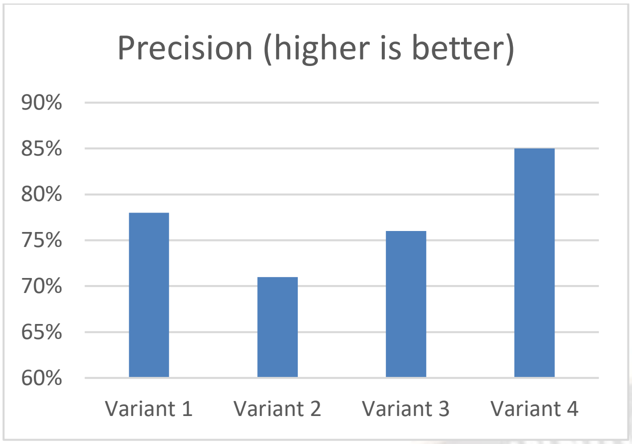

Figure 4. Ablation analysis for Precision

The above Table 2 presents a strong case for having both the UserGNN and the ContentGNN. As can be seen, Variant 2 and Variant 3 perform inferior to the full system (Variant 4). Also, Variant 1, which uses neither the UserGNN nor the ContentGNN, performs the worst. Figure 3 shows ablation analysis for RMSE. Figure 4 shows ablation analysis for Precision.

Hence, we empirically justify each design decision we took when proposing our GNN-based recommender system.

IV. CONCLUSION

In conclusion, this work describes an unique GNN-based recommender system for application in e-learning platforms. Using the OULAD dataset [20], we show that our proposed recommender system outperforms existing content-based and collaborative filtering-based recommender systems. Further, we justified each design decision underlying the proposed system through an ablation study that ensures that each component in the system contributes meaningfully towards higher precision.One of the limitations of the current approach is that it still suffers from cold-start issues (though not as much as existing approaches) because we still rely on some users using the e-learning platform. Future work could tackle this aspect of GNN-based recommender systems by using strategies employed in [3, 4] to minimize the impact of cold-start.

ACKNOWLEDGMENT: None

REFERENCES

[1] R. Burke, A. Felfernig, and M. H. Göker,“Recommender systems: An overview,” Ai Magazine, vol. 32(3), pp. 13-18, 2011

[2] M. J. Pazzani, and D. Billsus, “Content-based recommendation systems,” The adaptive web: methods and strategies of web personalization, pp. 325-341, 2007

[3] G. Linden, B. Smith, and J. York, “Amazon. com recommendations: Item-to-item collaborative filtering,” IEEE Internet computing, vol. 7, no. 1, pp. 76-80, 2003

[4] A. S. Das, M. Datar, A. Garg, andS. Rajaram,“Google news personalization: scalable online collaborative filtering,” In Proceedings of the 16th international conference on World Wide Web (pp. 271-280), 2007

[5] C. A. Gomez-Uribe, and N. Hunt, “The netflix recommender system: Algorithms, business value, and innovation. ACM Transactions on Management Information Systems (TMIS), vol. 6, no. 4, pp. 1-19, 2015

[6] Mosami Pulujkar, Sunil Kumar, Vivek Deshpande, Dattatray Waghole. (2023). Design and Implementation of the SIRC Protocol for Achieving QOS Parameters in Wireless Sensor Networks. International Journal of Intelligent Systems and Applications in Engineering, 11(2s), 311–315. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2698

[7] P. Chen, H. Xie, S. Maslov, andS. Redner, “Finding scientific gems with Google’s PageRank algorithm,” Journal of informetrics, vol. 1, no. 1, pp. 8-15, 2007

[8] S. Bengio, and Y. LeCun, “Scaling learning algorithms towards AI. Large-scale kernel machines,” vol. 34, no. 5, pp. 1-41, 2007

[9] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” Neural networks, vol. 2, no. 5, pp. 359-366, 1989

[10] A. Krizhevsky, “Advances in neural information processing systems” (No Title), 1097, 2012

[11] Johnson, M., Williams, P., González, M., Hernandez, M., & Muñoz, S. Applying Machine Learning in Engineering Management: Challenges and Opportunities. Kuwait Journal of Machine Learning, 1(1). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/90

[12] J. Zhou, G. Cui, S. Hu, Zhang, Z. Yang, C. Liu, Z.and Sun, M,“Graph neural networks: A review of methods and applications,” AI open, vol. 1, pp. 57-81, 2020

[13] F. Scarselli, M. Gori, A. C. Tsoi,Hagenbuchner, M.andMonfardini, G. “The graph neural network model,” IEEE transactions on neural networks, vol. 20, no. 1, pp. 61-80, 2008

[14] T. N. Kipf, and M. Welling, “Semi-supervised classification with graph convolutional networks,” arXiv preprint arXiv:1609.02907. 2016

[15] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I.Titov,and M.Welling,“Modeling relational data with graph convolutional networks,” In The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, pp. 3–7, 2018, Proceedings 15 (pp. 593-607). Springer International Publishing. 2018

[16] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, andD. Yin,“Graph neural networks for social recommendation,” In The world wide web conference (pp. 417-426), 2019

[17] Sunil and M. N. Doja, “An improved recommender system for e-learning environments to enhance learning capabilities of learners,” In Proceedings of ICETIT 2019: Emerging Trends in Information Technology (pp. 604-612). Springer International Publishing. 2020

[18] S. Bhaskaran, and R. Marappan, “Design and analysis of an efficient machine learning based hybrid recommendation system with enhanced density-based spatial clustering for digital e-learning applications,” Complex & Intelligent Systems, pp. 1-17, 2021

[19] S. Tahir, Y. Hafeez, M. A. Abbas, A. Nawaz, andB. Hamid, “Smart Learning Objects Retrieval for E-Learning with Contextual Recommendation based on Collaborative Filtering. Education and Information Technologies,” vol. 27, no. 6, pp. 8631-8668, 2022

[20] Dr. Bhushan Bandre. (2013). Design and Analysis of Low Power Energy Efficient Braun Multiplier. International Journal of New Practices in Management and Engineering, 2(01), 08 - 16. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/12

[21] V. B. P. Tolety, and E. V. Prasad, “Hybrid content and collaborative filtering based recommendation system for e-learning platforms,” Bulletin of Electrical Engineering and Informatics, vol. 11, no. 3, pp. 1543-1549, 2022

[22] H. Murad, and L. Yang, “Personalized e-learning recommender system using multimedia data,” International Journal of Advanced Computer Science and Applications, vol. 9, no. 9, pp. 565-567, 2018

[23] J. Kuzilek, M. Hlosta, and Z. Zdrahal, “Open university learning analytics dataset,” Scientific data, vol. 4, no. 1, pp. 1-8,2017