

Improving Accuracy and Interpretability of Educational App Recommendations via Hybrid Machine Learning and Deep Learning Techniques

Mohammed Najma Kulsoom.

Master's Student, Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India – 522502

Mobile : 9154063330

Email: najma8201@gmail.com

Dr. Rajendra Kumar Ganiya,

Professor,

Department of Computer Science and Engineering,
Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, A.P, 522302, India.

Mobile: +91 9393110375

E-mail: rajendragk@kluniversity.in

Abstract— As the usage of mobile apps in digital education is growing, it has become more critical to provide accurate and personalised recommendation systems that can enhance user engagement and learning effectiveness. The difficulty of collaborative filtering techniques is caused by the sparseness, dynamism and time dependence of crowdsourcing educational information. As a means of solving these problems, a mobile app recommendation framework is created with the help of a crowdsourced educational feedback dataset from Kaggle. It was methodologically preprocessed using timestamp creation, discard of irrelevant attributes, label encoding, and feature engineering (pivoting) and 80/20 train-test split. A number of machine and deep learning models are used including SVD, SVD++, NMF, SlopeOne, variations of KNN, CoClustering, BaselineOnly, NormalPredictor, GRU, LSTM, Stacked Autoencoder, Gated Autoencoder and hybrid LSTM-GRU. Explainable variance, MAE and RMSE are used to assess the model performance. The experimental results show that the hybrid LSTM-GRU consistently gives the minimum mean RMSE value of 0.267332 and mean MAE value of 0.118370 with the largest individual performance at $k = 16$ (RMSE = 0.005066, MAE = 0.003821). LIME and SHAP enhance the explainability, and user interface with Flask enables practical applicability. This technique increases the accuracy, robustness and interpretability of the mobile educational app recommendation systems.

“Keywords— Index Mobile App Recommendation, Collaborative Filtering, Deep Learning, Hybrid LSTM-GRU, Crowdsourced Educational Data, Explainable AI (XAI)”.

How to cite this article: Kulsoom MN, Ganiya RK., Improving Accuracy and Interpretability of Educational App Recommendations via Hybrid Machine Learning and Deep Learning Techniques. Int J Drug Deliv Technol. 2026;16(86s): 1095-1101; DOI: 10.25258/ijddt.16.86s.108

I. INTRODUCTION

The invention of digital content and services, and the vast amounts of data they carry, has led to the development of recommendation systems for personalised access. They have been widely used in such fields as e-commerce, entertainment and social networking, to match people to appropriate content based on their interests [1]. Data from these platforms is used to offer tailored suggestions and recommendations to users, such as in e-commerce and streaming services where data on user interactions, demographics, and behaviour are leveraged to deliver customized recommendations that boost user satisfaction and engagement [2]. Recommender systems have become important in aiding decision making processes with the rising digitisation of services, as they filter information overload and give useful suggestions [3].

In the commercial domain, recommender systems have been widely used and have been

successful, however, it is not without its challenges when applied in an educational context. Most of the educational data is inherently complex, scarce, context dependent and dynamic, coming from various sources like learning management systems, mobile applications, campus services, and social interactions online [4]. The traditional recommendation methods, however, are not good at adapting to these properties, resulting in low accuracy and limited personalisation [5]. Besides, existing solutions tend to poorly model developing learner interests and complicated user-item relationships in educational ecosystems [6]. This is a gap that highlights the need for a stronger and more flexible recommendations system in educational contexts.

To improve the mobile educational app suggestion by leveraging large-scale, crowd-sourced, educational data in order to gain a better understanding of student preferences in this work. The emphasis is on developing an overarching

recommendation framework that delivers accurate personalisation whilst being scalable and adaptive to changing user behaviour [7]. Furthermore, it centers on comparing performance and on interpretability to ensure that the results of the recommendations are valid and understandable for the stakeholders [8]. The main purpose of the work is to make recommendations more efficient on education platforms without using domain-specific heuristics by considering these elements.

What it does is to help enrich the learning experience by directing students to apps that are designed to address student needs and interests, thus contributing to education. The quality of the recommendations can increase the involvement of the students, utilize resources effectively and enhance the learning process in the digital learning environments [9]. Moreover, using explainable recommendation approaches helps promote trust and transparency, which are crucial for acceptance in academic contexts [10]. This is an initial step toward creating larger, intelligent, learner-centred systems that can enable personalised and data-driven decision making on scale.

II. LITERATURE REVIEW

There has been a lot of research on recommender systems to improve personalisation by considering many different user-centred aspects. Winoto and Tang [11] investigated how user mood impacts on movie recommendation and demonstrated the significant benefit that the contextual and affective information might bring to the relevancy of the recommendations. While they concluded that dynamic user state is important, their approach was studied only in a very narrow context and within limited context variables. Li and Tuzhilin [12] proposed a dynamical system approach for modelling developing consumer trajectories, which focuses on the temporal behaviour and long-term changes of preferences. The framework is good for recognizing dynamic patterns but it is not easy to scale it up and use it in real-world applications with large and complex data sets.

Su and Khoshgoftaar [13] gave a detailed survey of collaborative filtering approaches that established the basis for recommender systems. They had a strong emphasis on the benefits of neighbourhood based and model based approaches, but they also pointed to the consistent problems with sparsity, cold start and the inability to respond to changing user behaviour. Asad et al. [14] surveyed privacy preserving methods in federated recommendation systems and analyzed system-level problems. Their work, however, revealed practical challenges of guaranteeing user privacy, computational complexity, and recommendation accuracy as trade-offs, which are still open questions for practical educational and mobile platforms.

The scalability and sparsity are still the key challenges in the recommendation scenarios with

huge user-item space. These problems have been reviewed systematically by Singh [15] and he pointed out that sparse interaction matrices can adversely affect performance and generalisation. Mitigations are proposed, but solutions to manage dynamic and rapidly growing data sets do not seem to be common. In the educational setting, Moser [16] established a university recommendation system to help students in their choice making. The system was relevant to the domain, but limited to a specific academic setting, and did not leverage on crowdsourcing or temporal interaction data.

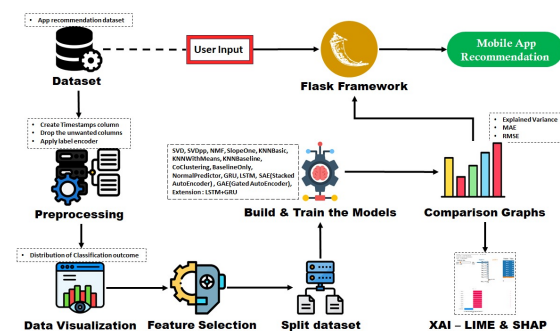
The recommender approaches in e-commerce applications have been discussed by Hossain et al. [17] and some general points at domain level are provided. Successful strategies for personalisation have been identified in the investigation and gaps identified in the transfer of such approaches into non-commercial contexts, like education. In the collaborative filtering field, Martins et al. [18] discussed the deep learning approach, which they believe to be more accurate and have a better representation learning capability. Their study, however, has pointed out the questions of interpretability, model complexity, and the requirement for large training data sets, which may make it less accessible in certain sensitive or resource-constrained environments.

Hybridisation and robustness to recommendations by structure optimisation have been explored recently. Camilli and Mézard [19] proposed a decimation approach for factoring a symmetric matrix, with a focus on increasing the stability of the factors, but mostly theoretical and little support in practice. Nabli et al. [20] came up with a hybrid recommendation architecture with clustering for customised cloud services and improved accuracy and flexibility. However their method was based on service recommendation and not educational or mobile app contexts. These studies put together, indicate that the lack of full adaptive coverage for sparse, dynamic, interpretable recommendations in educational environments still remains. In this work, we tackle these challenges by focusing on crowdsourced educational data, comparative evaluation and explainable outcomes, which can lead to more accurate, robust and practical mobile educational recommendation systems.

III. MATERIALS AND METHODS

The proposed method improves the accuracy and reliability of the mobile educational app suggestions by leveraging the user feedback data collected from popular mobile applications and crowdsourcing. The framework is designed to be followed along the end-to-end process, beginning with data intake and refinement, where the raw user-item interactions are converted into a representation suitable for learning; then data is segmented in a standard manner for an unbiased assessment. A set of collaborative filtering and deep learning based recommendation methods

are provided to model the latent preference structure and time evolution of user behaviour in a collaborative way. A hybrid LSTM-GRU is proposed to incorporate both longterm dependencies and short-term changes and achieve both prediction stability and good generalisation in both sparse and time-sensitive cases. Model performance is evaluated by means of explained variance, mean absolute error and root mean square error. Recommendation outputs are explained using explainable AI processes based on LIME and SHAP, ensuring transparency and enhancing user trust. Lastly, it has an improved usability via the Flask-based interface, enables practical deployment and provides real-time inference. System performance is better in terms of accuracy, robustness and interpretability in mobile education recommendation environments.



“Fig. 1. Proposed Architecture”

This is an architecture of recommender system, which utilizes machine learning and deep learning for personalised recommendations. Content-based filtering, collaborative filtering, association techniques, clustering, bayesian techniques, neural network, model-based filtering, memory-based filtering using user-based and item-based approaches are the ML algorithms. Algorithms for DL that use Graph Auto-Encoders, Stacked Auto-Encoders, Gated Recurrent Units and LSTM improve representation learning, provide scalable recommendations, accurate and context-aware across various data sources, and enable real-time decision making in real world applications.

A) Dataset Collection

This system's dataset is a crowdsourced dataset of feedback on educational mobile apps collected from the Kaggle repository. It has 15,000 entries of user-app interaction. For each record, there is a unique user identification, an application identifier and an explicit rating label and a timestamp for temporal behaviour. There are numeric and categorical variables in the dataset that represent the various types of user preferences and interaction patterns that have occurred over time. It is a moderately sized system, providing explicit feedback and temporal information, which are ideal for the evaluation of personalised and time-aware recommendation systems.

B) Pre-Processing

The preprocessing stage involves distribution analysis, categorical attribute encoding, construction of user-item representations, and organization of the raw crowdsourced educational data, which is essential for the successful learning of reliable, unbiased and fast recommendation models.

Exploratory Data Analysis and Distribution Assessment: We perform an exploratory study to understand the data features and we analyse the distribution of ratings over users and applications. The individual frequency of every rating level is analysed and visualised to discover the imbalance patterns and user preference preferences. This stage is crucial to identify significant feedback behaviours and to understand any skewed distributions which may have an impact on recommendation performance, bias and evaluation results in customised learning environments.

Application Usage Analysis: This will show the distribution of application usage on the data set, In order to determine the relative popularity and frequencies of interaction of the mobile applications. The system can learn the extent of user interactivity with the items (how often each one is used) and also the dominant level of each item (how many times it is used). The stage is critical for understanding the imbalance of exposure of items, which is directly connected with collaborative learning and the fairness of recommendations, and the strength of the preference modelling of popular and unpopular applications.

Feature Encoding: Encoding categorical identifiers that identify mobile applications as numbers for mathematical modelling and statistical analysis. This change allows it to be used with similarity based and learning based recommendation methods which require numeric input. Categorical attributes can be encoded while preserving individual item identities and enabling rapid computation, thus supporting correlation analysis and rapid generation of utility matrices that facilitate smooth integration with classic recommendation algorithms and with deep learning systems.

Correlation Analysis: A correlation analysis is conducted on the linear correlation of the numerical attributes in the data set. The process provides the insight about the dependency relationship between user identities, application identities, ratings and the temporal features after the transformation. Being familiar with the correlations can help identify redundant or redundant features and can help inform feature engineering decisions. It also checks to see whether there are relationships that might affect prediction accuracy, or bias the prediction.

Feature Engineering: The data of user-item interaction are converted to a structured utility

matrix with rows as users and columns as applications and the values in the cell are ratings. Neutral values are placed in the matrix for missing interactions to preserve the matrix coherence. This representation is crucial in the context of collaborative filtering and neural recommendation systems, as it provides a concise representation of preference patterns. The matrix created can be used quickly to learn the latent linkage between users and applications.

C) Training and Testing:

Processed data are split into a training set and a testing set, with a fixed proportion split, in order to have an unbiased evaluation of performance. This separation guarantees that the model assessment is due to generalisation to unseen data and not memorisation. Also the data are reshaped in suitable formats to allow both conventional recommendation approaches and deep learning models. A good split and format is essential to enable a fair comparison, stability and reliable performance assessment.

D) Algorithms

Singular Value Decomposition (SVD): SVD decomposes the interactions to low dimensional representations, representing the latent characteristics associated with users and items. It can capture the underlying preference structure and minimize the impact of noise and sparsity and consequently enhance the accuracy of any recommendation and the generalisation capability.

Singular Value Decomposition Plus Plus (SVD++): SVD++ is an extension of the latent factor model which incorporates the implicit user interactions as explicit feedback. This inclusion provides for better preference learning, sparsity robustness, and better and more customized recommendations.

Non-negative Matrix Factorization (NMF): Non-negative Matrix Factorisation learns additive latent representations which preserve the interpretability of user/item preferences. It preserves the non-negativity, thus improves stability, decreases overfitting and enhances generalisation over sparse interaction matrices.

Slope One: Slope One ratings are based on the average difference between each pair of items (based on known interactions). It is simple to compute, robust to dynamic updates and reasonably accurate to be a viable lightweight collaborative baseline.

KNNBasic: Neighborhood-based reasons: Combining preferences of related users or things. It has good learning of local interaction patterns and gives interpretable advice and is still performing well on the areas for which it has the right interaction density.

KNNWithMeans: KNNWithMeans adds an extra step to neighbourhood based prediction by normalising ratings by means adjustments. This removes user bias and item bias and enhances the accuracy and robustness by focusing on relative preference patterns instead of absolute ratings.

KNNBaseline: KNNBaseline is a method which is a hybrid of neighbourhood similarity and baseline bias correction, designed to separate the trends at global level from the trends at the local level. This pairing reduces prediction instability, noise and provides good generalisation under sparse data regimes.

CoClustering: CoClustering clusters users and items together to represent the common interactive behaviour at the cluster level. This method takes advantage of collective preference patterns to reduce dimensionality, deal with sparsity and boost robustness.

BaselineOnly: BaselineOnly predicts evaluations based on global, user and item bias components. Easy to implement and provides consistent and robust estimates as a solid reference, because it does not require interaction modelling.

NormalPredictor: NormalPredictor is a predictor which predicts from a global rating distribution. It offers a stochastic baseline to compare with, useful for robustness analysis and to identify the benefits of informed recommendation algorithms.

Gated Recurrent Unit (GRU): GRU models sequential interaction pattern using gated memory processes with memory of relevant temporal information. It makes time dependent recommendation more accurate and more robust, it has fast learning and strong generalization ability.

Long Short-Term Memory (LSTM): LSTM is used to capture lengthy temporal dependencies with the use of memory cells and gating mechanisms. It enables the precise modelling of user preferences' evolution and offers robustness and generalisation in sequential recommendation scenarios.

Stacked Autoencoder (SAE): This is a type of Autoencoder that learns the hierarchical latent representations of interaction by reconstructing interaction data for multiple layers of Autoencoders. This deep representation learning reduces noise, overcomes sparsity and enhances predictive accuracy and generalisation.

Gated Autoencoder (GAE): Learning expressive latent representations with interaction-aware gating in Gated Autoencoders. This mechanism can selectively enrich the corresponding patterns, thus boosting the robustness, adaptability and the accuracy of the recommendation on complex interaction data.

Hybrid LSTM-GRU: Hybrid LSTM-GRU design which retains the ability to capture long-term memory and models short-term dynamics. This approach improves the quality of temporal representation, accuracy and offers good and well generalized suggestions in time sensitive contexts.

IV. EXPERIMENTAL RESULTS

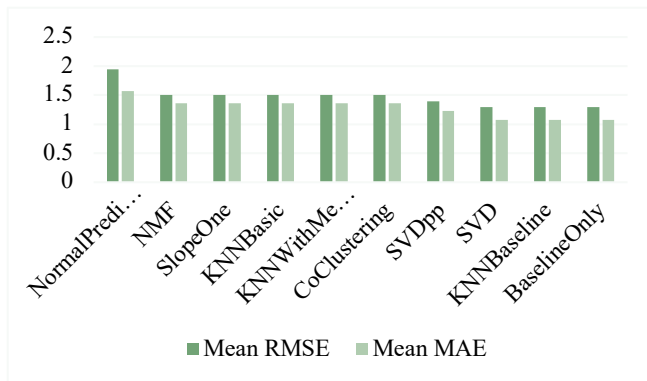
RMSE: RMSE is the mean of the differences between the observed values and the model values of a statistical model, at a specific point in time. In terms of the calculation, it is the standard deviation of the residuals. Residuals are the distances from the data points to the regression line.

“Table. 1. Performance Evaluation Table”

Model	RMSE	MAE
NormalPredictor	1.947676	1.570670
NMF	1.508077	1.359505
SlopeOne	1.508077	1.359505
KNNBasic	1.508077	1.359505
KNNWithMeans	1.508077	1.359505
CoClustering	1.508077	1.359505
SVDpp	1.396566	1.232219
SVD	1.291230	1.078124
KNNBaseline	1.289471	1.075780
BaselineOnly	1.290447	1.076168

The comparison results of recommender models based on RMSE and MAE are presented in the table. The error is the biggest for NormalPredictor, and the error for both SVD, KNNBaseline and BaselineOnly is the smallest, which means that they both are the most accurate predictors.

“Fig. 2. Comparison Graph “



The graph RMSE and MAE for the recommender models demonstrates that the NormalPredictor has the highest errors while SVD, KNNBaseline and BaselineOnly have lower RMSE and MAE, which indicates that these models are more accurate at predicting.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n ||y(i) - \hat{y}(i)||^2}{N}} \quad (1)$$

MAE: MAE – the size of the error of your measurements. This is the difference between the actual measured value and the "true value". For example, if the scale says you weigh 90 pounds but you know you really weigh 89 pounds, then the absolute inaccuracy of the scale is 90 lbs - 89 lbs = 1 lbs.

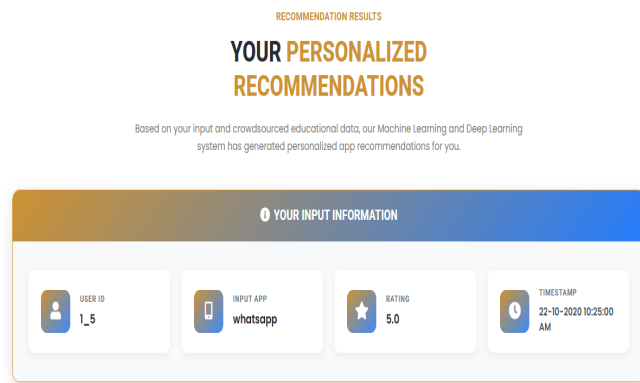
$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2)$$

“Fig.3 Enter the input”

The interface features a customised application recommendation workflow, which includes a unique user ID submission, an application progress step-by-step and a machine-learning recommendation process via a user-friendly web form.

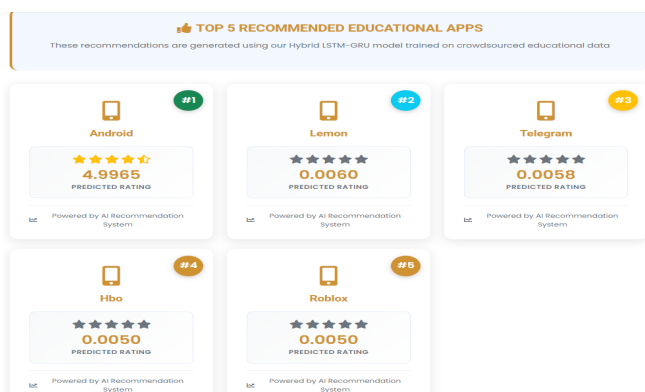
“Fig.4 Predicted Result”

This is the completed feedback stage of the rating screen (5 is the maximum rating and satisfaction with the product/process) and the next step in the recommendation workflow process.



“Fig.5 Enter the input”

The screen's recommendation results with the personalised suggestions by the machine learning system is displayed in Fig. 5. The selected app, the user ID, and the rating score and the time stamp information are displayed to the user.



“Fig.6 Predicted Result”

Fig. 6. The top five educational applications suggested by the Hybrid LSTM-GRU model. The figure shows the ranking applications and forecasted ratings along with the details of the AI-based recommendation system in a structured fashion.

V. CONCLUSION

To summarise, the main goal of this system is to increase the accuracy and reliability of mobile educational app suggestions, by making efficient use of crowdsourcing user feedback data and advanced learning algorithms. For this purpose, an education App Feedback dataset from kaggle was used and carefully pre-processed by creating the right level of time (timestamp), dropping irrelevant attributes, encoding the labels, feature engineering using pivot and splitting it into train and test sets at 80/20. We have implemented a wide variety of traditional collaborative filtering techniques such as SVD, SVD++, NMF, SlopeOne, KNNBasic, KNNWithMeans, KNNWithZScore, CoClustering, BaselineOnly, NormalPredictor as well as deep learning based architectures such as GRU, LSTM, Stacked Autoencoder and Gated Autoencoder for recommendation models. Moreover, we develop an enhanced hybrid LSTM-GRU model to learn complex temporal user-item relationships. The model was studied in terms of the explained variance, MAE and RMSE. The hybrid LSTM-GRU model gave the best results with the lowest mean RMSE (0.267332) and the lowest mean MAE (0.118370), and the

highest individual performance ($k = 16$, RMSE = 0.005066, MAE = 0.003821). LIME + SHAP was used to improve the explainability of the model and a Flask interface was used to ensure the deployment feasibility. In general, it provides a strong recommendation system which is both interpretable and realistically deployable, enabling automated decision making and user-centric personalisation in mobile educational situations.

Future research on the same theme could be focused on enhancing the mobility application recommendation systems in terms of scalability, adaptability and contextual awareness. The approach can be extended to utilize additional multimodal data sources like textual reviews, usage logs and context signals like location and time to improve the user's preference modelling. Temporal dependency learning and personalisation accuracy can be further enhanced by using advanced attention-based architectures and transformer models. Online learning methods can also be incorporated to enable the model to be updated continuously as the user gives feedback online. Federated learning is another privacy-preserving technique that can be used for secure data use. These enhancements can increase the strength of the system, the level of personalisation, and its use in different mobile learning contexts.

REFERENCES

- [1] Kord, A., Aboelfetouh, A., & Shohieb, S. M. (2025). Academic course planning recommendation and students' performance prediction multimodal based on educational data mining techniques. *Journal of Computing in Higher Education*, 1-39.
- [2] San, K. K., Win, H. H., & Chaw, K. E. E. (2025). Enhancing Hybrid Course Recommendation with Weighted Voting Ensemble Learning. *Journal of Future Artificial Intelligence and Technologies*, 1(4), 337-347.
- [3] Zhu, Y., Dai, W., & Kang, Q. (2025). The analysis of artificial intelligence-based mobile learning in students' open teaching recommendation system based on deep learning. *Scientific Reports*, 15(1), 21927.
- [4] Mulyana, H. L., & Rumaisa, F. (2024). Course learning recommendation system using neural collaborative filtering. *Brilliance: Research of Artificial Intelligence*, 4(2), 517-524.
- [5] Zhang, C., Hu, X., & Wei, W. (2024, November). A Review of AI-based Techniques in Online Course Recommendation: Metrics, Factors, and Research Methods. In *2024 International Conference on Intelligent Education and Intelligent Research (IEIR)* (pp. 1-8). IEEE.
- [6] M. Zhai, X. Wang, and X. Zhao, "The importance of online customer reviews characteristics on remanufactured product sales: Evidence from the mobile phonemarket on Amazon.com," *J.RetailingConsum.Services*, vol. 77, Mar. 2024, Art. no. 103677.
- [7] C. C. Aggarwal, "An introduction to recommender systems," in *Recommender Systems: The Textbook*. Cham, Switzerland: Springer, 2015, pp. 1-28.
- [8] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76-80, Jan. 2003.
- [9] N.TintarevandJ.Masthoff, "Beyondexplainingingleitemrecommendations," in *Recommender Systems Handbook*. Cham,Switzerland: Springer, 2022, pp. 711-756.
- [10] C. Musto, M. de Gemmis, P. Lops, F. Narducci, and G. Semeraro, "Semantics and content-based recommendations," in *Recommender Systems Handbook*. Cham, Switzerland: Springer, 2022, pp. 251-298.
- [11] P. Winoto and T. Y. Tang, "The role of user mood in movie recommendations," *Expert Syst. Appl.*, vol. 37, no. 8, pp. 6086-6092, Aug. 2010.

- [12] P. Li and A. Tuzhilin, "A dynamical system framework for exploring consumer trajectories in recommender system," SSRN, May 2024, pp. 1–55.
- [13] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, no. 1, Oct. 2009, Art. no. 421425.
- [14] M. Asad, S. Shaikat, E. Javanmardi, J. Nakazato, and M. Tsukada, "A comprehensive survey on privacy-preserving techniques in federated recommendation systems," *Appl. Sci.*, vol. 13, no. 10, p. 6201, May 2023.
- [15] M. Singh, "Scalability and sparsity issues in recommender datasets: A survey," *Knowl. Inf. Syst.*, vol. 62, no. 1, pp. 1–43, Jan. 2020.
- [16] M. Moser, "College Tennis University recommendation system (CTURS)," M.S. thesis, Dept. Psychol., Univ. Salzburg, Salzburg, Austria, 2022.
- [17] I. Hossain, M. A. H. Palash, A. T. Sejuty, N. A. Tanjim, M. A. A. L. Nasim, S. Saif, A. B. Suraj, M. M. A. Haque, and N. Karim, "A survey of recommender system techniques and the e-commerce domain," 2022, arXiv:2208.07399.
- [18] G. B. Martins, J. P. Papa, and H. Adeli, "Deep learning techniques for recommender systems based on collaborative filtering," *Expert Syst.*, vol. 37, no. 6, p. 12647, Dec. 2020.
- [19] F. Camilli and M. Mézard, "The decimation scheme for symmetric matrix factorization," *J. Phys. A, Math. Theor.*, vol. 57, no. 8, Feb. 2024, Art. no. 085002.
- [20] H. Nabli, R. B. Djemaa, and I. A. B. Amor, "Improved clustering-based hybrid recommendation system to offer personalized cloud services," *Cluster Comput.*, vol. 27, no. 3, pp. 2845–2874, Jun. 2024.
- [21] A. A. Ka'bi, "Proposed artificial intelligence algorithm and deep learning techniques for development of higher education," *Int. J. Intell. Netw.*, vol. 4, pp. 68–73, Jun. 2023.
- [22] W. Wang, W. Chen, Y. Luo, Y. Long, Z. Lin, L. Zhang, B. Lin, D. Cai, and X. He, "Model compression and efficient inference for large language models: A survey," 2024, arXiv:2402.09748.
- [23] R. Nahta, G. S. Chauhan, Y. K. Meena, and D. Gopalani, "Deep learning with the generative models for recommender systems: A survey," *Comput. Sci. Rev.*, vol. 53, Aug. 2024, Art. no. 100646.
- [24] J. H. Yoon and B. Jang, "Evolution of deep learning-based sequential recommender systems: From current trends to new perspectives," *IEEE Access*, vol. 11, pp. 54265–54279, 2023.
- [25] A. K. Yengikand, M. Meghdadi, S. Ahmadian, S. M. J. Jalali, A. Khosravi, and S. Nahavandi, "Deep representation learning using multilayer perceptron and stacked autoencoder for recommendation systems," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2021, pp. 2485–2491.
- [26] J. Choi, J.-K. Rhee, and H. Chae, "Cell subtype classification via representation learning based on a denoising autoencoder for single-cell RNA sequencing," *IEEE Access*, vol. 9, pp. 14540–14548, 2021.
- [27] Y. Geng, X. Xiao, X. Sun, and Y. Zhu, "Representation learning: Recommendation with knowledge graph via triple-autoencoder," *Frontiers Genet.*, vol. 13, Jun. 2022, Art. no. 891265.
- [28] S. K. Pandey and R. R. Janghel, "Recent deep learning techniques, challenges and its applications for medical healthcare system: A review," *Neural Process. Lett.*, vol. 50, no. 2, pp. 1907–1935, 2019.
- [29] S. Nosouhian, F. Nosouhian, and A. K. Khoshouei, "A review of current neural network architecture for sequence learning: Comparison between LSTM and GRU," Preprints, Jul. 2021. [Online]. Available: <https://www.preprints.org/manuscript/202107.0252/v1>
- [30] S. Zargar, "Introduction to sequence learning models: RNN, LSTM, GRU," Dept. Mechanical and Aerospace Engineering, North Carolina State Univ., Raleigh, NC, USA, Tech. Rep., 2021.