International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

Enhancing Mobile App Recommendations with Crowd source Educational Data Using Machine Learning and Deep Learning

Chityala Dharma Teja
PG , Department of Computer science and engineering,
Sree Dattha institute of Engineering & Science,
Hyderabad, Telangana, India - 501510
Email: dharmatejachityala@gmail.com

Dr. GNV Vibhav Reddy
Associate
Professor, Department of Computer science and engineering, Sree Dattha institute of Engineering & Science, Hyderabad, Telangana, India - 501510
Email: vicechairman@sreedattha.ac.in

***Abstract*— The increasing adoption of mobile applications in digital education has intensified the need for accurate and personalized recommendation systems to improve user engagement and learning effectiveness. Traditional collaborative filtering techniques face limitations when handling sparse, evolving, and time-dependent crowdsourced educational data. To overcome these challenges, a mobile app recommendation framework is developed using a crowdsourced educational feedback dataset obtained from Kaggle. The dataset is systematically preprocessed through timestamp generation, removal of irrelevant attributes, label encoding, pivot-based feature engineering, and an 80/20 train–test split. Multiple machine learning and deep learning models are implemented, including SVD, SVD++, NMF, SlopeOne, KNN variants, CoClustering, BaselineOnly, NormalPredictor, GRU, LSTM, Stacked Autoencoder, Gated Autoencoder, and a hybrid LSTM–GRU architecture. Model performance is evaluated using explained variance, mean absolute error (MAE), and root mean square error (RMSE). Experimental results demonstrate that the hybrid LSTM–GRU model consistently outperforms all baseline and standalone deep learning models, achieving the lowest mean RMSE of 0.267332 and mean MAE of 0.118370, with the best individual performance at k = 16 (RMSE = 0.005066, MAE = 0.003821). Explainability is further enhanced using LIME and SHAP, and practical applicability is supported through a Flask-based user interface. This approach significantly improves accuracy, robustness, and interpretability in mobile educational app recommendation systems.**

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

I. INTRODUCTION

Recommender systems have become an essential component of modern information systems, enabling personalized access to vast collections of digital content and services. They are widely adopted across domains such as e-commerce, entertainment, and social networking to assist users in discovering relevant items aligned with their preferences [1]. Platforms such as online retail stores and streaming services leverage user interaction data, demographic information, and behavioral patterns to deliver tailored recommendations that improve user satisfaction and engagement [2]. With the increasing digitization of services, recommender systems now play a crucial role in supporting decision-making processes by filtering information overload and presenting meaningful suggestions [3].

Despite their success in commercial domains, the application of recommender systems in educational environments presents unique challenges. Educational data are often heterogeneous, sparse, dynamic, and context-dependent, arising from diverse sources such as learning management systems, mobile applications, campus services, and online social interactions [4]. Traditional recommendation approaches frequently struggle to adapt to these characteristics, leading to reduced accuracy and limited personalization [5]. Furthermore, existing solutions often fail to adequately capture evolving learner interests and complex user–item relationships inherent in educational ecosystems [6]. This gap highlights the need for more robust and adaptive recommendation frameworks tailored to educational contexts.

The primary objective of this contribution is to enhance mobile educational app recommendation by leveraging large-scale, crowdsourced educational data to model learner preferences more effectively. The focus is on developing a comprehensive recommendation framework that supports accurate personalization while remaining scalable and adaptable to dynamic user behavior [7]. In addition, the contribution emphasizes comparative performance assessment and interpretability to ensure that recommendation outcomes are both reliable and understandable to stakeholders [8]. By addressing these aspects, the work aims to advance recommendation effectiveness in educational

platforms without relying on domain-specific heuristics.

The significance of this contribution lies in its potential to improve learning experiences by guiding students toward relevant educational applications that align with their interests and needs. Enhanced recommendation quality can lead to increased engagement, better resource utilization, and improved learning outcomes in digital education environments [9]. Moreover, the integration of explainable recommendation mechanisms fosters trust and transparency, which are critical for adoption in academic settings [10]. Overall, the presented approach contributes to the broader goal of intelligent, learner-centered educational systems capable of supporting personalized and data-driven decision-making at scale.

II. LITERATURE REVIEW

Recommender systems have been extensively explored to improve personalization by incorporating diverse user-centric factors. Winoto and Tang [11] investigated the influence of user mood on movie recommendations, demonstrating that contextual and affective information can significantly enhance recommendation relevance. While their findings highlighted the importance of dynamic user states, the approach relied on limited contextual variables and was evaluated in a relatively constrained domain. Li and Tuzhilin [12] introduced a dynamical system framework to model evolving consumer trajectories, emphasizing temporal behavior and long-term preference shifts. Although effective in capturing dynamic patterns, the framework faced challenges related to scalability and practical deployment in large, heterogeneous datasets.

Foundational insights into recommender systems are provided by Su and Khoshgoftaar [13], who presented a comprehensive survey of collaborative filtering techniques. Their analysis outlined the strengths of neighborhood-based and model-based approaches but also underscored persistent issues such as sparsity, cold-start problems, and limited adaptability to changing user behavior. Addressing system-level concerns, Asad et al. [14] surveyed privacy-preserving techniques in federated recommendation environments. While their work highlighted methods to protect user data across distributed systems, it revealed trade-offs between privacy guarantees, computational overhead, and recommendation accuracy, leaving open challenges for real-world educational and mobile platforms.

Scalability and sparsity remain central obstacles in recommendation scenarios involving large user–item spaces. Singh [15] systematically reviewed these issues, emphasizing that sparse interaction matrices often degrade performance and generalization. Despite proposing mitigation strategies, the survey noted the lack of unified solutions capable of handling dynamic and rapidly growing datasets. In an educational context, Moser [16] developed a university recommendation system to support student decision-making. Although the system demonstrated domain relevance, its scope was limited to a specific academic setting and did not fully exploit crowdsourced or temporal interaction data.

Broader domain-level perspectives are offered by Hossain et al. [17], who surveyed recommender techniques in e-commerce applications. Their work highlighted successful personalization strategies but also identified gaps in transferring these methods to non-commercial domains such as education. Martins et al. [18] examined deep learning approaches for collaborative filtering, reporting improved accuracy and representation learning capabilities. Nevertheless, their analysis pointed out challenges related to interpretability, model complexity, and the need for extensive training data, which can restrict adoption in sensitive or resource-constrained environments.

Recent advances have explored hybridization and structural optimization to enhance recommendation robustness. Camilli and Mézard [19] proposed a decimation scheme for symmetric matrix factorization, improving factor stability but remaining primarily theoretical with limited application validation. Nabli et al. [20] introduced a clustering-based hybrid recommendation framework for personalized cloud services, demonstrating gains in accuracy and flexibility. However, their solution emphasized service recommendation rather than educational or mobile app contexts. Collectively, these studies reveal persistent gaps in handling sparse, dynamic, and interpretable recommendations for educational environments. The current contribution addresses these challenges by focusing on crowdsourced educational data, comparative evaluation, and explainable outcomes, thereby advancing accuracy, robustness, and practical applicability in mobile educational recommendation systems.

III. MATERIALS AND METHODS

The proposed system focuses on improving the accuracy and reliability of mobile educational app recommendations by exploiting crowdsourced user feedback data collected from widely used mobile applications. The framework follows a structured end-to-end workflow, beginning with data ingestion and refinement to convert raw user–item interactions into a learning-ready representation, followed by standardized data partitioning to ensure unbiased evaluation. A broad spectrum of collaborative filtering and deep learning–based recommendation techniques is employed to capture both latent preference patterns and temporal dynamics in user behavior. A hybrid LSTM–GRU strategy is introduced to jointly model long-term dependencies and short-term variations, thereby enhancing prediction stability and generalization under sparse

and time-sensitive conditions. Model effectiveness is assessed using explained variance, mean absolute error, and root mean square error. To promote transparency and user trust, explainable artificial intelligence mechanisms based on LIME and SHAP are integrated to interpret recommendation outcomes. Finally, practical deployment is supported through a Flask-based interface, enabling real-time inference and improved usability. Overall, the system delivers enhanced accuracy, robustness, and interpretability for mobile educational recommendation environments.

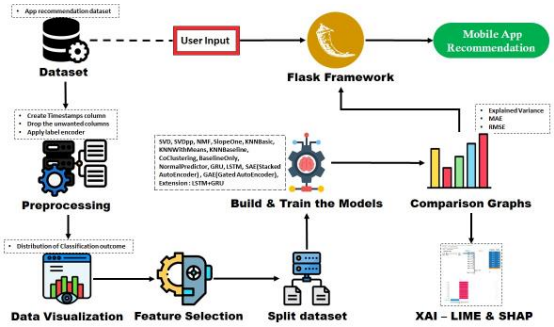


Fig. 1. Proposed Architecture

This architecture illustrates a recommender system integrating machine learning and deep learning approaches to generate personalized recommendations. Machine learning algorithms include content-based filtering and collaborative filtering, using association techniques, clustering, Bayesian methods, neural networks, model-based filtering, and memory-based filtering with user-based and item-based approaches. Deep learning algorithms enhance representation learning through Graph Auto-Encoders, Stacked Auto-Encoders, Gated Recurrent Units, and Long Short-Term Memory networks, enabling scalable, accurate, and context-aware recommendations across diverse data sources and supporting real-time decision-making in practical applications.

A) Dataset Collection

The dataset employed in this system is a crowdsourced educational mobile app feedback dataset obtained from the Kaggle repository. It consists of 15,000 user–app interaction records, each comprising a unique user identifier, application identifier, explicit rating label, and a timestamp reflecting temporal behavior. The dataset contains both numerical and categorical attributes, capturing diverse user preferences and interaction patterns over time. Its moderate size, temporal information, and explicit feedback structure make it well suited for evaluating personalized and time-aware recommendation systems.

B) Pre-Processing

The preprocessing phase prepares raw crowdsourced educational data by analyzing distributions, encoding categorical attributes, constructing user–item representations, and organizing inputs to ensure reliable, unbiased, and efficient recommendation model learning.

Exploratory Data Analysis and Distribution Assessment: An initial exploratory analysis is conducted to understand the underlying characteristics of the dataset and assess rating distributions across users and applications. The frequency of each rating level is analyzed and visualized to identify imbalance patterns and user preference tendencies. This step is essential for revealing dominant feedback behaviors and ensuring awareness of skewed distributions, which can significantly influence recommendation performance, bias, and evaluation outcomes in personalized learning environments.

Application Usage Analysis: The distribution of application usage across the dataset is examined to identify the relative popularity and interaction frequency of each mobile application. By analyzing how often each application appears in user interactions, the system gains insight into user engagement diversity and item-level dominance. This step is important for understanding item exposure imbalance, which directly affects collaborative learning, recommendation fairness, and the robustness of preference modeling across frequently and infrequently used applications.

Feature Encoding: Categorical identifiers representing mobile applications are transformed into numerical representations to enable mathematical modeling and statistical analysis. This transformation ensures compatibility with similarity-based and learning-based recommendation techniques that require numeric inputs. Encoding categorical attributes preserves distinct item identities while allowing efficient computation, facilitating correlation analysis, utility matrix construction, and seamless integration with both traditional recommendation methods and deep learning architectures.

Correlation Analysis: A correlation analysis is performed to examine linear relationships among numerical attributes within the dataset. This step provides insights into the dependency structure between user identifiers, application identifiers, ratings, and temporal attributes after transformation. Understanding correlations helps identify redundant or weakly informative features and supports informed feature engineering decisions. It also aids in assessing whether relationships exist that may influence prediction accuracy or introduce unintended bias.

Feature Engineering: User–item interaction data are reorganized into a structured utility matrix, where rows represent users, columns represent applications, and cell values denote ratings. Missing interactions are filled with neutral values to maintain matrix consistency. This representation is

fundamental for collaborative filtering and neural recommendation models, as it captures preference patterns in a compact form. The resulting matrix enables efficient learning of latent relationships between users and applications.

C) Training and Testing:

The processed dataset is divided into training and testing subsets using a fixed proportional split to support unbiased performance evaluation. This separation ensures that model assessment reflects generalization to unseen data rather than memorization. Additionally, the data are reshaped into appropriate formats to support both conventional recommendation techniques and deep learning models. Proper splitting and formatting are critical for fair comparison, stability, and reliable performance estimation.

D) Algorithms

Singular Value Decomposition (SVD): Singular Value Decomposition models latent user and item factors by decomposing interaction data into low-dimensional representations. It improves recommendation accuracy and generalization by capturing hidden preference structures while reducing noise and sparsity effects.

Singular Value Decomposition Plus Plus (SVD++): SVD++ enhances latent factor modeling by incorporating implicit user interactions alongside explicit feedback. This inclusion enables richer preference learning, improves robustness under sparse conditions, and delivers more accurate and personalized recommendations.

Non-negative Matrix Factorization (NMF): Non-negative Matrix Factorization learns additive latent representations that preserve interpretability of user and item preferences. By enforcing non-negativity, it improves stability, reduces overfitting, and enhances generalization across sparse interaction matrices.

Slope One: Slope One predicts ratings using average deviations between item pairs derived from known interactions. Its simplicity enables fast computation, robustness to dynamic updates, and reasonable accuracy, making it suitable as a lightweight collaborative baseline.

KNNBasic: KNNBasic applies neighborhood-based reasoning by aggregating preferences from similar users or items. It captures local interaction patterns effectively, offering interpretable recommendations and stable performance in regions with sufficient interaction density.

KNNWithMeans: KNNWithMeans improves neighborhood-based prediction by normalizing ratings using mean adjustments. This reduces user and item bias, enhances accuracy, and improves robustness by focusing on relative preference patterns rather than absolute scores.

KNNBaseline: KNNBaseline combines neighborhood similarity with baseline bias correction to separate global tendencies from local interactions. This integration improves prediction stability, reduces noise, and enhances generalization under sparse data conditions.

CoClustering: CoClustering groups users and items simultaneously to model shared interaction behavior at a cluster level. This strategy reduces dimensionality, mitigates sparsity, and improves robustness by leveraging collective preference patterns.

BaselineOnly: BaselineOnly predicts ratings using global, user, and item bias components. Although simple, it provides stable and robust estimates, serving as a strong reference by capturing systematic rating behavior without interaction modeling.

NormalPredictor: NormalPredictor generates predictions by sampling from a global rating distribution. It offers a stochastic baseline for comparison, supporting robustness analysis and highlighting performance gains achieved by informed recommendation methods.

Gated Recurrent Unit (GRU): GRU models sequential interaction patterns through gated memory mechanisms that retain relevant temporal information. It improves accuracy and robustness for time-dependent recommendations while maintaining efficient learning and stable generalization.

Long Short-Term Memory (LSTM): LSTM captures long-term temporal dependencies using memory cells and gating structures. This enables accurate modeling of evolving user preferences, improving robustness and generalization in sequential recommendation scenarios.

Stacked Autoencoder (SAE): Stacked Autoencoders learn hierarchical latent representations by reconstructing interaction data through multiple layers. This deep representation learning reduces noise, addresses sparsity, and enhances predictive accuracy and generalization.

Gated Autoencoder (GAE): Gated Autoencoders incorporate interaction-aware gating to learn expressive latent features. This mechanism selectively emphasizes relevant patterns, improving robustness, adaptability, and recommendation accuracy across complex interaction data.

Hybrid LSTM–GRU: The hybrid LSTM–GRU architecture integrates long-term memory retention with efficient short-term dynamics modeling. This fusion enhances temporal representation quality,

improves accuracy, and delivers robust, well-generalized recommendations in time-sensitive environments.

IV. Experimental Results

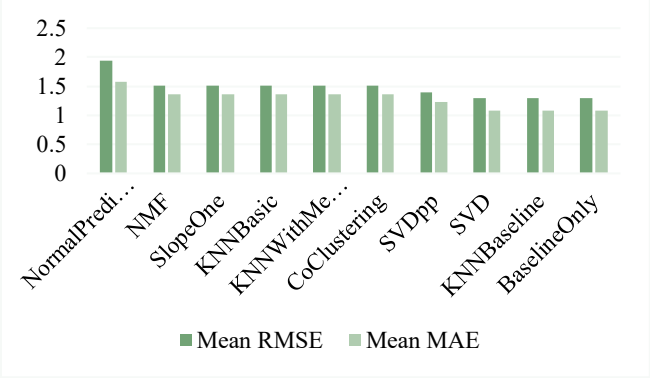
RMSE: The root mean square error (RMSE) measures the average difference between a statistical model's predicted values and the actual values. Mathematically, it is the standard deviation of the residuals. Residuals represent the distance between the regression line and the data points.

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 table compares recommender models using RMSE and MAE, showing NormalPredictor with highest error, while SVD, KNNBaseline, and BaselineOnly achieve the lowest errors, indicating superior prediction accuracy.

Fig. 2. Comparison Graph



The comparison graph illustrates RMSE and MAE across recommender models, highlighting NormalPredictor with highest errors, while SVD, KNNBaseline, and BaselineOnly show lower error values, indicating improved prediction accuracy.

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

MAE: Absolute Error is the amount of error in your measurements. It is the difference between the measured value and "true" value. For example, if a scale states 90 pounds but you know your true weight is 89 pounds, then the scale has an absolute error of 90 lbs – 89 lbs = 1 lbs.

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

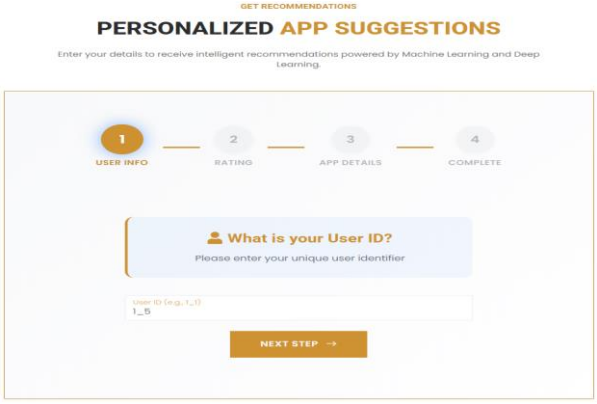


Fig.3 Enter the input

The interface displays a personalized app suggestion workflow where users enter a unique user ID, follow step-based navigation, and proceed to generate machine learning–driven recommendations through an intuitive web form.

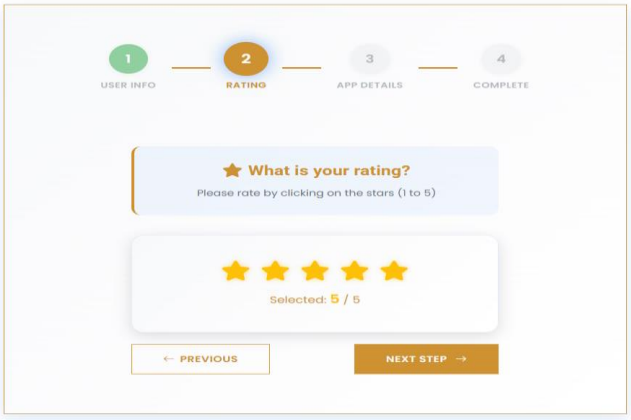


Fig.4 Predicted Result

The rating screen shows a completed feedback step where the user selects a full five-star score, confirming maximum satisfaction, before proceeding to the next stage of the recommendation workflow process.

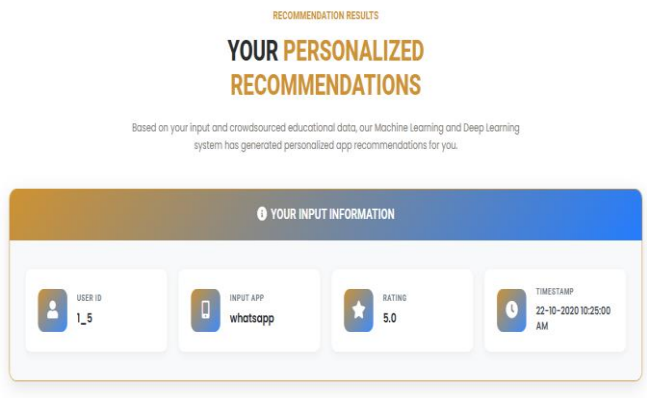


Fig.5 Enter the input

Fig. 5 presents the recommendation results screen displaying personalized suggestions generated by the machine learning system, along with user input details including user ID, selected app, rating score, and timestamp information.

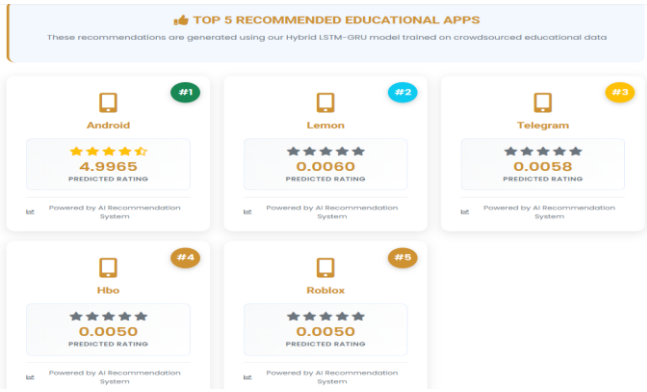


Fig.6 Predicted Result

Fig. 6 displays the top five recommended educational apps generated by the Hybrid LSTM-GRU model, presenting ranked applications with predicted ratings and AI-powered recommendation system details in a structured layout.

V. Conclusion

In conclusion, the primary objective of this system is to improve the accuracy and reliability of mobile educational app recommendations by effectively leveraging crowdsourced user feedback data and advanced learning techniques. To achieve this goal, a Kaggle-sourced educational app feedback dataset was utilized and systematically processed through timestamp generation, removal of irrelevant attributes, label encoding, pivot-based feature engineering, and an 80/20 train–test split. A broad spectrum of recommendation techniques was implemented, including traditional collaborative filtering models such as SVD, SVD++, NMF, SlopeOne, KNN variants, CoClustering, BaselineOnly, and NormalPredictor, alongside deep learning architectures comprising GRU, LSTM, Stacked Autoencoder, and Gated Autoencoder. An enhanced hybrid LSTM–GRU model was further employed to capture complex temporal user–item interaction patterns. Model evaluation was conducted using explained variance, mean absolute error (MAE), and root mean square error (RMSE). The hybrid LSTM–GRU model demonstrated superior performance, achieving the lowest mean RMSE of 0.267332, the lowest mean MAE of 0.118370, and the best individual performance at k = 16 with RMSE of 0.005066 and MAE of 0.003821. Explainability was strengthened through the integration of LIME and SHAP, while deployment feasibility was ensured using a Flask-based interface. Overall, the system delivers a robust, interpretable, and practically deployable recommendation framework that enhances automated decision-making and user-centric personalization in mobile educational environments.

Future advancements of this system may focus on improving scalability, adaptability, and contextual awareness of mobile educational app recommendations. The framework can be extended to incorporate multimodal data sources such as textual reviews, usage logs, and contextual signals including location and time to enrich user preference modeling. Advanced attention-based architectures and transformer models may further enhance temporal dependency learning and personalization accuracy. Online learning mechanisms can be integrated to support continuous model updates with real-time user feedback. Additionally, privacy-preserving techniques such as federated learning can be adopted to ensure secure data utilization. These enhancements would strengthen system robustness, personalization depth, and practical applicability across diverse mobile learning environments.

References

[1] Kord, A., Aboelfetouh, A., & Shohieb, S. M. (2025). Academic course planning recommendation and students' performance prediction multi-modal 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 phonemarketonAmazon.com,''J.RetailingConsum.Services, vol. 77, Mar. 2024, Art. no. 103677.

[7] C. C. Aggarwal, ''An introduction to recommender systems,'' in Rec ommender 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,''Beyondexplainingsingleitemrecommenda tions,'' 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 Sys tems Handbook. Cham, Switzerland: Springer, 2022, pp. 251–298.

[11] P. Winoto and T. Y. Tang, ''The role of user mood in movie recommenda tions,'' Expert Syst. Appl., vol. 37, no. 8, pp. 6086–6092, Aug. 2010.

[12] P. Li and A. Tuzhilin, ‘‘A dynamical system framework for explor ing consumer trajectories in recommender system,’’ SSRN, May 2024, pp. 1–55.

[13] X. Su and T. M. Khoshgoftaar, ‘‘A survey of collaborative filtering tech niques,’’ Adv. Artif. Intell., vol. 2009, no. 1, Oct. 2009, Art. no. 421425.

[14] M. Asad, S. Shaukat, E. Javanmardi, J. Nakazato, and M. Tsukada, ‘‘A comprehensive survey on privacy-preserving techniques in feder ated recommendation systems,’’ Appl. Sci., vol. 13, no. 10, p. 6201, May 2023.

[15] M. Singh, ‘‘Scalability and sparsity issues in recommender datasets: Asurvey,’’ 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 ecommerce domain,’’ 2022, arXiv:2208.07399.

[18] G. B.Martins, J. P. Papa, and H. Adeli, ‘‘Deep learning techniques for rec ommender 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 mul tilayer perceptron and stacked autoencoder for recommendation sys tems,’’ 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: Rec ommendation 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, chal lenges 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 recur rent 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.