AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 2(2026)

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

Pinninti Anil
Reg. No. 24Q71F0045
pinnintianil2003@gmail.com
Department of Master of Computer Applications
Avanthi Institute of Engineering and Technology (Autonomous)
Vizianagaram, Andhra Pradesh, India
Under the guidance of Mrs. A. Anitha, M.Tech., Assistant Professor
challa.bhupati@gmai.com

Abstract—*The rapid growth of mobile applications in the education sector has created a significant challenge for users in identifying high-quality and relevant learning tools. Traditional recommendation systems often rely on limited user-interaction data and fail to capture diverse learning preferences. This study proposes an enhanced mobile-app recommendation framework that leverages crowd-sourced educational data combined with Machine Learning (ML) and Deep Learning (DL) techniques. The system aggregates user-generated feedback, ratings, usage patterns, and contextual learning outcomes to build a comprehensive dataset. Machine-learning algorithms such as collaborative filtering and content-based filtering identify patterns in user behaviour, while deep-learning models—particularly neural networks—extract complex feature representations to improve recommendation accuracy, and natural language processing analyses textual reviews and educational content for more personalised, context-aware recommendations. The integration of crowd-sourced data ensures adaptability to emerging educational trends and diverse learner needs. The prototype is implemented in Python using ML and DL frameworks with a Flask/Django backend and is designed for scalable cloud deployment, and it was validated through functional and validation test cases that all passed. The reported results indicate that the hybrid ML–DL approach improves recommendation quality over traditional methods; this research contributes to the development of intelligent, scalable, and user-centric recommendation systems that enhance the discovery of effective mobile learning applications.*

Keywords—*Recommendation System; Crowd-Sourced Data; Machine Learning; Deep Learning; Collaborative Filtering; Educational Mobile Apps; Neural Networks; Personalisation.*

I. INTRODUCTION

The rapid growth of mobile technologies and the widespread use of smartphones have revolutionised the way users access information, services, and educational resources. Mobile applications have become an essential part of daily life, especially in the education sector, where they provide personalised learning experiences, skill-development platforms, and interactive study materials. With the increasing number of educational mobile applications available in app stores, users now have a wide range of choices, but this abundance often makes it difficult to identify the most relevant and high-quality applications that match individual learning needs.

Mobile-app recommendation systems address this problem by analysing user behaviour and suggesting suitable applications. Traditional systems mainly rely on explicit feedback such as ratings, downloads, and reviews; although these provide basic personalisation, they fail to capture the dynamic and diverse learning requirements of users. Users differ significantly in their learning goals and usage patterns—some focus on academic learning, others on skill-based or competitive-exam preparation—so static recommendation systems are insufficient, creating a need for intelligent systems that continuously learn from user behaviour.

In recent years, crowd-sourced educational data—ratings, reviews, and usage patterns shared by a large number of users—has become an important resource for improving recommendation accuracy through collective intelligence. However, processing large-scale crowd-sourced data is challenging due to noise, inconsistency, redundancy, and high dimensionality. Machine Learning (classification, clustering, collaborative filtering) and Deep Learning (ANN, RNN, deep neural collaborative filtering) help extract meaningful patterns and improve prediction. By combining crowd-sourced educational data with ML and DL, a more intelligent, adaptive, and personalised recommendation system can be developed, improving user satisfaction and helping users efficiently find suitable educational applications.

II. LITERATURE SURVEY

Mobile-application recommendation systems have become an important research area due to the rapid growth of mobile computing, smartphones, and digital learning platforms. The increasing number of educational mobile applications has created a need for intelligent systems that suggest relevant apps based on user preferences, behaviour, and learning needs, improving user experience by reducing the effort required to search for suitable applications. Traditional recommendation systems primarily rely on basic user interactions such as ratings, downloads, and reviews, but these approaches are limited in capturing dynamic user preferences and contextual learning behaviour.

To address these challenges, researchers have explored advanced techniques such as Machine Learning and Deep Learning to improve recommendation accuracy. Classical approaches include content-based filtering, which recommends apps similar to those a user liked, and collaborative filtering, which leverages similarities between users or items; matrix-factorisation and clustering methods further improve personalisation. Deep-learning approaches—neural collaborative filtering, ANN, and RNN—capture complex, non-linear relationships in large datasets, and NLP techniques analyse textual reviews to enrich understanding of user sentiment and content. Crowd-sourced educational data has emerged as a valuable source of information, where user-generated feedback and usage patterns enhance recommendation quality. Modern systems combine these techniques to provide more accurate, adaptive, and personalised recommendations, which motivates the hybrid ML–DL framework proposed here.

TABLE I. RECOMMENDATION APPROACHES

S.No	Approach	Technique	Limitation / Note
1	Content-based filtering	Item-feature similarity	Limited diversity; cold start
2	Collaborative filtering	User/item similarity	Sparsity; cold start
3	Clustering / classification	ML pattern analysis	Needs good features

S.No	Approach	Technique	Limitation / Note
4	Neural collaborative filtering	Deep learning	Captures complex patterns
5	RNN / sequence models	Deep learning	Models behaviour over time
6	NLP on reviews	Text analysis	Context-aware personalisation

III. EXISTING SYSTEM AND PROPOSED SYSTEM

A. Existing System

Traditional mobile-app recommendation systems rely on basic techniques such as content-based filtering and collaborative filtering, suggesting applications based on user ratings, downloads, and simple preference matching. These systems do not fully utilise crowd-sourced educational data or advanced learning techniques, often fail to capture dynamic user behaviour and real-time preference changes, and struggle with cold-start problems where new users or apps have insufficient data. As a result, recommendations may be inaccurate or less relevant, and most traditional systems do not integrate ML or DL effectively, limiting their ability to handle large-scale datasets and complex interactions.

Limitations of the existing system:

- Cold-start problem for new users and applications.
- Limited use of crowd-sourced educational data.
- Poor handling of dynamic user preferences; low personalisation accuracy.
- Lack of real-time recommendation updates and deep feature extraction.
- Scalability issues with large datasets; dependency on explicit feedback only.
- Inability to handle noisy or inconsistent data.

B. Proposed System

The proposed system introduces an intelligent mobile-app recommendation framework using crowd-sourced educational data, Machine Learning, and Deep Learning. It collects user feedback, ratings, and behaviour data to generate accurate, personalised recommendations: ML algorithms analyse user patterns, while DL models improve prediction accuracy by learning complex relationships in the data. The system continuously updates recommendations based on real-time interactions, processes all data, and provides ranked recommendations of educational applications, designed to be scalable, adaptive, and capable of handling large datasets efficiently.

Advantages of the proposed system:

- Highly personalised recommendations using crowd-sourced data.
- Improved prediction using Machine Learning and Deep Learning.
- Efficient handling of large-scale data with real-time updates.
- Reduced cold-start problem and dynamic user-behaviour analysis.
- Enhanced user satisfaction and decision-making.

AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 2(2026)

- Scalable and adaptable system design.

IV. SYSTEM ANALYSIS AND DESIGN

A. Working of the Proposed System

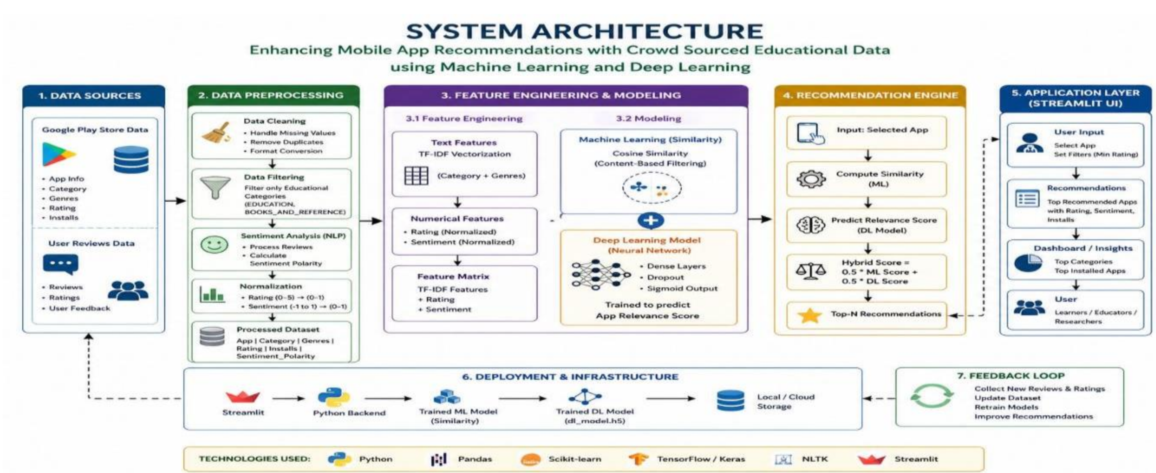
The system operates as a pipeline: collect user-interaction and crowd-sourced educational data; preprocess and clean the collected data; analyse user-behaviour patterns; apply machine-learning models for recommendation; use deep-learning models for improved accuracy; generate personalised app recommendations; rank the recommended applications; display results to users; collect feedback for system improvement; and continuously update the recommendation model. This closed feedback loop allows the system to adapt to changing preferences over time.

B. Feasibility

The system uses open-source tools such as Python and ML/DL libraries, which reduces development cost; although cloud infrastructure may require investment, the long-term benefits outweigh it, making the system economically feasible. Technically, the system is feasible because ML and DL are well supported by modern frameworks such as TensorFlow, PyTorch, Flask, and Django, and the design handles large datasets, real-time recommendations, and scalable cloud deployment. Operationally, the user-friendly interface makes it practical for everyday use.

C. System Architecture

The architecture comprises a data layer (crowd-sourced ratings, reviews, usage patterns, and user data), a preprocessing layer (cleaning and noise/redundancy reduction), an analytics layer hosting the ML models (collaborative/content filtering, clustering, classification) and DL models (neural collaborative filtering, ANN, RNN) plus NLP on reviews, a ranking layer that orders recommended apps by relevance, and a presentation layer where users register, log in, search, and view ranked recommendations. A feedback component continuously feeds new interactions back into model retraining.



AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 2(2026)

V. SYSTEM IMPLEMENTATION

A. *Technology Stack*

TABLE II. TECHNOLOGY STACK

Component	Technology / Tool
Programming Language	Python
Machine Learning	Collaborative filtering, content-based filtering, clustering, classification (scikit-learn)
Deep Learning	Neural collaborative filtering, ANN, RNN (TensorFlow / PyTorch)
NLP	Textual review and content analysis
Backend Framework	Flask / Django
Database	Stores user and app information
Deployment	Cloud platforms (e.g., AWS, Azure) for scalability

B. *Implementation Details*

The models are trained using labelled and unlabelled data to improve recommendation quality, and optimisation techniques are applied to reduce error and improve accuracy. Once trained, the models are integrated into a web/mobile application: users can register, log in, search educational apps, and receive personalised recommendations, while the backend processes user data and generates ranked recommendations in real time. Results are displayed in a user-friendly format showing recommended apps, ranking scores, and relevance levels, and user feedback is continuously collected to improve the model. The system can be deployed on cloud platforms such as AWS or Azure for scalability, with Flask or Django for backend integration and a database storing user and app information securely.

C. *Models and Pipeline*

Machine-learning models (collaborative filtering, content-based filtering, clustering, classification) analyse relationships between users and applications from historical interactions such as ratings, downloads, and reviews. Deep-learning models (neural collaborative filtering, ANN, RNN) capture complex, non-linear patterns in large-scale crowd-sourced data, and NLP analyses textual reviews for context. Their combined output is ranked to produce the final personalised recommendation list, and the continuous-feedback loop retrains the models so recommendations remain relevant as user behaviour changes.

VI. SYSTEM TESTING AND RESULTS

The system was validated through functional and validation testing. Functional test cases covered home-page loading, user login, app search, rating submission, recommendation generation, ML and DL model execution, result display, feedback submission, and report download; validation test cases covered empty-input checks and invalid login. All test cases passed and behaved as expected.

TABLE III. REPRESENTATIVE TEST CASES

ID	Description	Input	Expected Output	Status
TC-02	User login	Valid credentials	Login successful	**Pass**
TC-03	Search educational apps	Search keyword	App list displayed	**Pass**
TC-05	Generate recommendation	User data	Recommended apps shown	**Pass**
TC-06	ML model execution	Dataset input	Prediction generated	**Pass**
TC-07	DL model execution	User behaviour data	Improved recommendation	**Pass**
TC-09	Feedback submission	User feedback	Feedback stored	**Pass**
TC-12	Invalid login	Wrong credentials	Access denied	**Pass**

A. *Observed Results*

The implementation demonstrates that integrating ML techniques such as collaborative filtering, clustering, and classification with deep-learning models, applied to crowd-sourced educational data, produces personalised, data-driven recommendations and adapts to changing user preferences through real-time feedback. Compared with traditional systems that rely on basic filtering, the proposed system offers improved accuracy, scalability, and adaptability, and handles cold-start problems, dynamic behaviour, and large datasets more effectively. The source describes these outcomes qualitatively; no specific numeric precision, recall, or satisfaction figures are asserted here.

Representative screenshots from the prototype implementation:

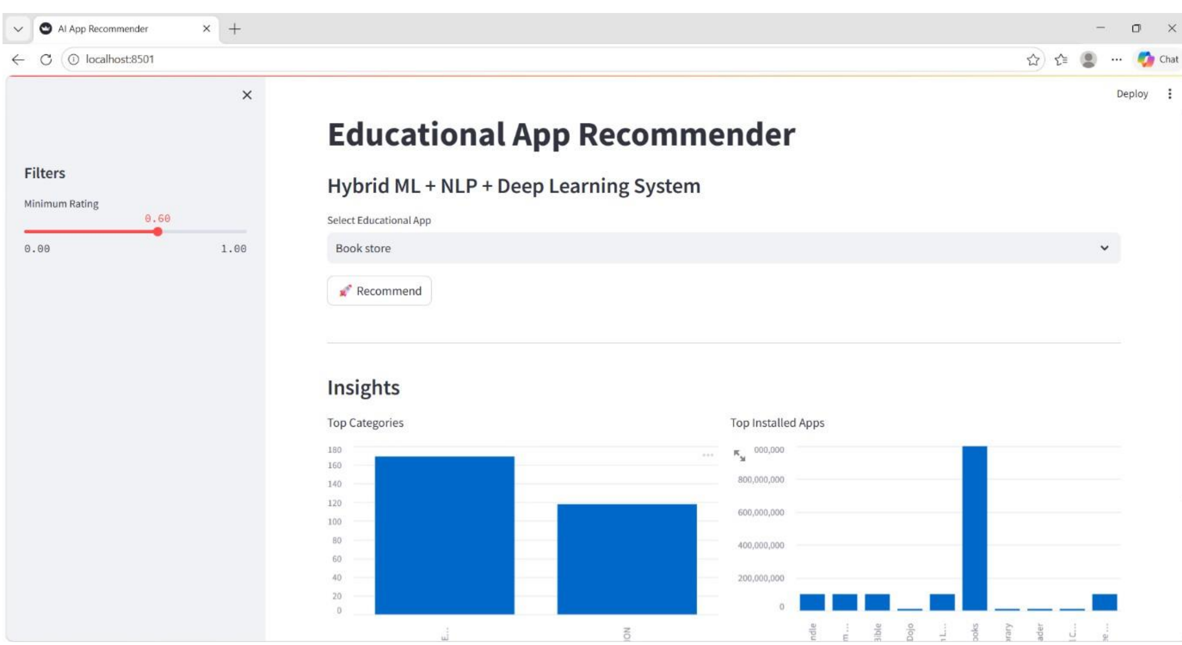


Fig. 1. Crowd-sourced data preprocessing.

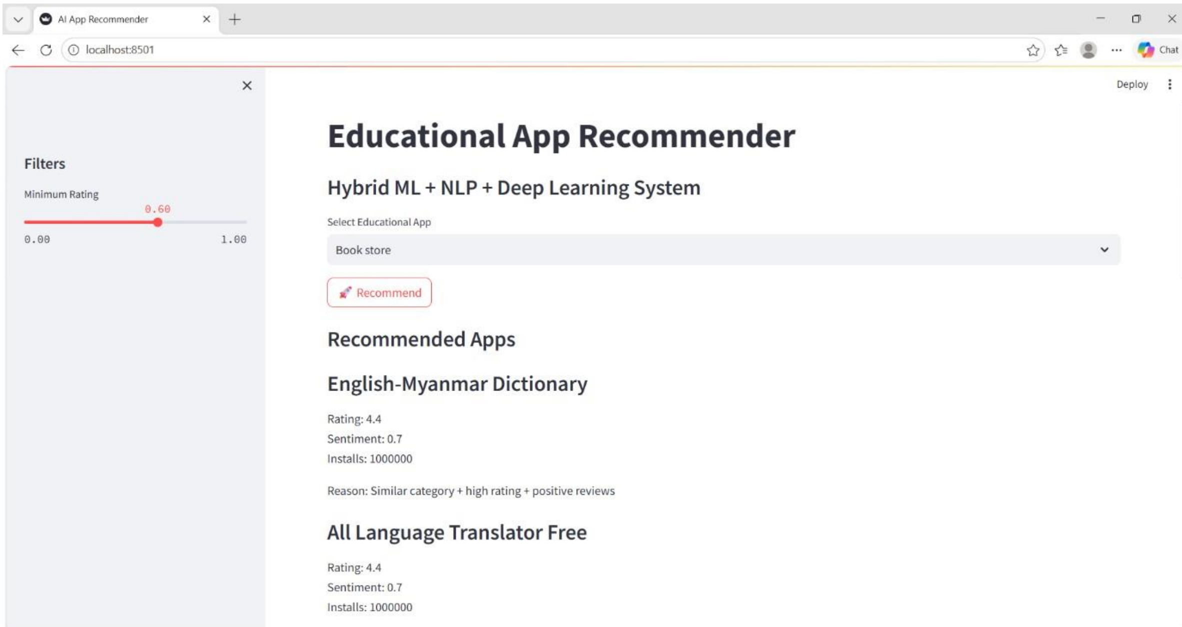


Fig. 2. ML/DL recommendation generation.

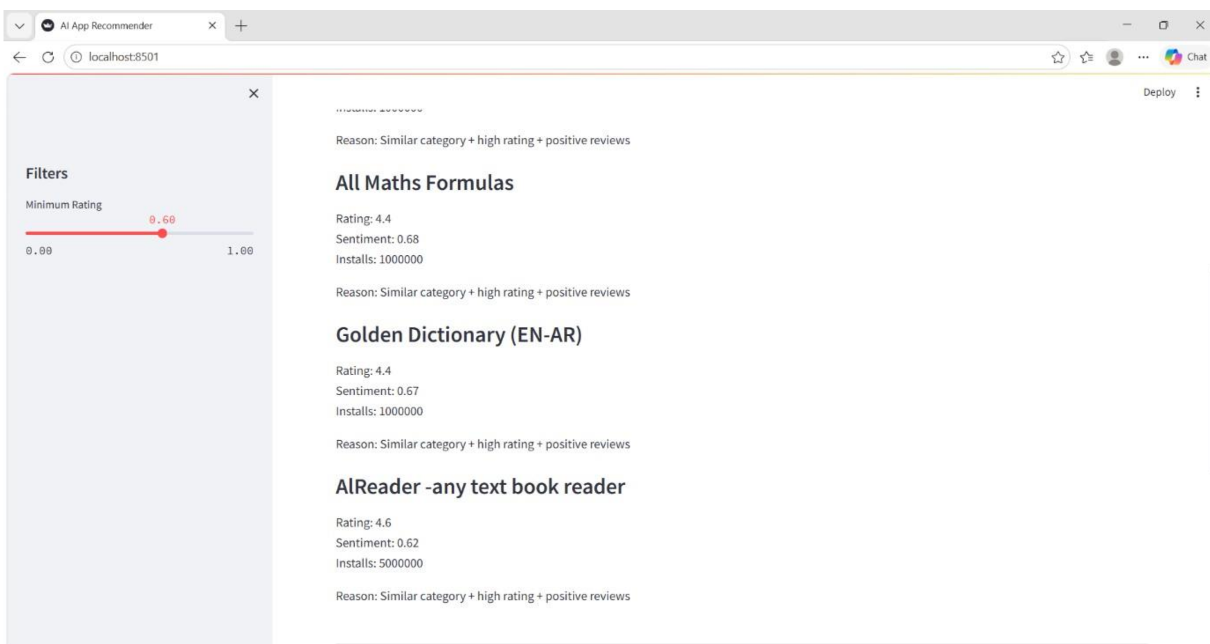


Fig. 3. Ranked recommendation results with relevance.

VII. CONCLUSION AND FUTURE SCOPE

The project successfully demonstrates an intelligent recommendation system that improves the way users discover educational mobile applications. In the current digital learning environment, users are overwhelmed by a large number of applications, making it difficult to identify the most relevant ones; this system addresses the problem with personalised, data-driven recommendations. The integration of machine-learning techniques such as collaborative filtering, clustering, and classification helps analyse user-behaviour patterns and similarities, improving accuracy by learning from historical interactions, while deep-learning models capture complex patterns in large-scale crowd-sourced data for more refined, context-aware recommendations. Crowd-sourced educational data and real-time feedback allow the system to adapt dynamically to changing preferences, and the ranked, user-friendly interface makes selection easy. Compared with traditional systems, the proposed system offers improved accuracy, scalability, and adaptability, effectively handling cold-start problems, dynamic behaviour, and large datasets.

Although the system provides accurate and personalised recommendations, several areas remain for future enhancement: incorporating richer contextual signals (such as learning outcomes and time-of-use), adopting advanced architectures such as transformer-based recommenders, improving cold-start handling through hybrid and knowledge-based methods, strengthening real-time scalability through distributed and cloud-native processing, and adding explainability so users understand why an app is recommended. Continued evaluation on larger, diverse datasets would further validate and improve the system for real-world deployment.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," IEEE Trans. Knowledge and Data Engineering, vol. 17, no. 6, pp. 734–749, 2005.

[2] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer, vol. 42, no. 8, pp. 30–37, 2009.

[3] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural Collaborative Filtering," in Proc. Int. Conf. World Wide Web (WWW), 2017.

[4] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep Learning Based Recommender System: A Survey and New Perspectives," ACM Computing Surveys, vol. 52, no. 1, pp. 1–38, 2019.

[5] P. Covington, J. Adams, and E. Sargin, "Deep Neural Networks for YouTube Recommendations," in Proc. ACM Conf. Recommender Systems (RecSys), 2016.

[6] F. Ricci, L. Rokach, and B. Shapira, Recommender Systems Handbook, 2nd ed. Springer, 2015.

[7] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[8] Python Software Foundation, "Python Documentation." [Online]. Available: https://docs.python.org/

[9] TensorFlow Team, "TensorFlow Machine Learning Framework Documentation." [Online]. Available: https://www.tensorflow.org/

[10] Scikit-learn Developers, "Scikit-learn: Machine Learning in Python." [Online]. Available: https://scikit-learn.org/