Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 02, NCRTCS - 2026

An SQL Guided Adaptive Artificial Intelligence Framework for Context-Aware Personalized Recommendation Systems: A Hybrid Case Study Approach

Theertha Anil Nambiar,
MAEER's MIT Arts, Commerce and Science College, Alandi(D.)

Amol Bajirao Kale
MAEER's MIT Arts, Commerce and Science College, Alandi(D.)

***Abstract*— The main marketing strategy behind modern digital platforms like streaming services and e-commerce websites is "Recommendation systems", which suggests personalized content and influences user engagement. However, existing AI recommendation models mostly focus only on gathering historical data of user interactions, and thus hardly focus on dynamic contexts such as current session activity, last interacted time, or situational preferences. On the other hand, SQL based approaches though efficient in data gathering, and filtering, but they cannot learn from user interaction and feedback.**

To address this problem, this paper proposes a new hybrid strategy- "SQL guided adaptive artificial intelligence framework", a hybrid approach that combines SQL data filtering capabilities with AI's interactive learning to provide context-aware and personalized recommendations.

In the proposed framework, a simulated set of data is generated, in which recent user interactions are extracted, and with necessary constraints applied, it filters a set of candidate items using SQL queries. Subsequently, a collaborative filtering and similarity based ranking is applied using an AI model, which predicts a set of suitable products and ranks items within the set.

SQL databases store the user interactions, and constant updates in the database act as a feedback loop and make the AI model learn and update itself. The practical adaptiveness of the framework is demonstrated via a case study inspired by real-world OTT platforms and e-commerce websites.

Comparative analysis demonstrates that it outperforms standalone SQL based and AI only recommendation approaches. There is an enhancement in the personalization and explainability in modern recommendation systems by combining database intelligence with computational intelligence.

Keywords*— *Recommendation Systems, Context-Aware Computing, SQL-Guided AI, Personalized Filtering, Hybrid Case Study

I. INTRODUCTION

User engagement has become a deciding factor for the business profitability of digital platforms like Netflix, Spotify and Amazon. These systems use recommendation system for analysing user behaviour and then suggest relevant products or services.

Traditional recommendation approaches can be categorized into database-driven and artificial driven systems. SQL-based systems retrieve and filter data

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 02, NCRTCS - 2026

from huge datasets but fail to incorporate dynamic user interactions. In contract, exclusive AI-based models use deep learning to highly learn complex patterns but often require frequent retraining, and is very expensive.

This paper proposes a novel hybrid framework which integrates SQL-based candidate filtering into AI-based recommendation framework. The objective is to improve recommendation accuracy by combining structured data processing.

II. RELATED WORK

A. *Consumer Behaviour and Recommendation Influence in Online Platforms*

Maruthavani and Shantharajah, have developed a real time healthcare recommendation system for various 'social media platforms' using 'adaptive learning techniques' to increase the responsiveness of service recommendations [1]. Areej Bin Suhaim and Jawad Berri have proposed a context-aware recommender system for social networks based on the use of contextual information such as user activity, social relationships, and interaction environments to create personalization for recommendation relevance for the users [2]

Limitation: These systems emphasize contextual and real-time recommendations but lack integration with structured database intelligence for scalable, cross-domain personalized recommendation support.

B. *Collaborative Filtering and Feedback-Based Recommendation Systems*

Padhy et al. suggested collaborative filtering technique for recommendation systems for e-commerce sites, which improved personalization by modelling user similarity [3]. Bhaskar, Kundur, and Lawryshyn suggested a deep framework of collaborative filtering, which infers customer purchasing behavior through aggregated implicit feedback and neural collaborative filtering approaches [4].

Wang and Wang proposed FedPrivRec, which is a privacy-preserving federated learning method for facilitating real-time e-commerce recommendation with user data localization and assurance of recommendation accuracy using a safe training strategy [5].

Limitation: These approaches face challenges in handling sparse feedback, preserving user privacy, and maintaining recommendation accuracy in dynamic, large-scale environments.

C. *Context-Aware Meta-Hybrid Recommendation Approaches*

Tibenský and Kompan introduced a context-aware adaptive recommendation system whereby suitable recommendation techniques can be dynamically selected by employing a meta-hybrid approach to optimize personalization [6]

Limitation: The approach increases system complexity and computational overhead, making real-time deployment challenging in large-scale, rapidly changing recommendation environments.

D. *SQL-Based Recommendation and Pattern Mining Systems*

Silva, Almeida, and Queiroz explained how SQL has transformed from conventional relational databases to contemporary Big Data solutions, with applicability to distributed, NoSQL, and scale-out data processing platforms as seen with [7]. Aggarwal and Han describe fundamental data mining techniques and database-oriented knowledge discovery, emphasizing efficient pattern extraction from structured datasets, which supports scalable and data-driven recommendation system development [8].

Rizvi et al. proposed a SQL-based multi-topic recommendation system using a collaborative filtering approach, which showcases the use of relational databases in developing an explainable lightweight recommendation system [9].

Limitation: SQL-based systems rely heavily on structured data and SQL-based methods, limiting adaptability, scalability, and effectiveness in handling complex, dynamic, and unstructured user preferences.

E. *Integration of Artificial Intelligence into Database Systems*

Pavlo *et al.* analyze machine learning–based agents for autonomous database tuning, comparing external and internal integration approaches and highlighting their trade-offs in scalability, control, and system complexity [10]. Panwar investigates AI-driven query optimization using machine learning models to improve query execution plans, achieving significant performance gains, reduced resource

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 02, NCRTCS - 2026

utilization, and adaptive optimization across SQL and NoSQL database systems [11].

Zahir and El Qadi propose a machine learning-based recommendation system for database execution plans, in an attempt to enhance query performance by selecting efficient plans based on learned workload and query characteristics [12]., while Timoshenko studies optimization of MLOps pipelines for high-load product recommendation systems, focusing on feature consistency, embedding scalability, inference optimization, and continuous training to maintain low latency and stable business performance [13].

Limitation: These studies focus on optimization and scalability but lack unified frameworks addressing interpretability, cross-domain generalization, and seamless integration with real-world dynamic workloads.

F. *SQL-AI Hybrid Recommendation Systems and explainable AI models*

Zhang, Lu, and Jin talk about the role of artificial intelligence methods in enhancing the performance of recommender systems, such as deep learning, transfer learning, reinforcement learning, and fuzzy methods, and some challenges are also identified [14]. Desku et al. propose a software engineering recommender system using SQL semantic search, where the effectiveness of the system has been tested using precision, recall, and using subjective assessment by software developers [15].

Saranya and Subhashini provide a systematic review on explainable AI models and applications, discussing the present developments in the field, specific applications for these models in different domains, and the possible future directions, with the emphasis on the challenges associated with the transparency and interpretability of the models [16].

Limitation: These studies provide broad surveys but lack empirical validation, domain-specific benchmarks, and practical deployment insights for real-time, large-scale recommendation systems.

III. Proposed SQL Guided Adaptive AI Recommendation Framework

This section gives a blueprint of the architecture and working mechanism of the proposed SQL-guided adaptive AI framework for personalized recommendations. It aims to leverage the power of structured data filtering using SQL databases along with adaptive abilities of artificial intelligence models.

A. *System Architecture*

The proposed system will have a layer-based architecture with three primary components- data layer, processing layer, and intelligence layer.

Relational databases are utilized with SQLite to store user profiles, content metadata, and interaction data. These structured tables make it easier to store user behavioral data. The feedback from the user are also timely incorporated into the same database.

The processing layer performs filtering and candidate selection-based queries. This layer retrieves necessary information based on various context-based parameters, such as activity levels, user defined genres, and interaction frequency.

The intelligence level utilizes collaborative filtering rankings. It uses the filtered candidate data from SQL database and uses a machine learning approach to create recommendations. These recommendations are ranked and then served to the user. A feedback mechanism updates the database after each interaction from the user.

B. *Workflow of the Proposed Framework*

The operational workflow of the system is shown in Fig 1 and includes the following phases.

1. Data is continuously collected and stored in the interaction table.
2. SQL queries retrieve recent and relevant contextual data from the database.
3. Candidate content items are filtered according to user preference and session history.
4. The AI system also checks for user similarity and content relevance.
5. Ranked recommendations are produced and shown to users.
6. User feedback is recorded for future learning purposes.

This hybrid workflow ensures that the outputs of structure-based filtering as well as adaptive learning are considered in the final recommendation.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 02, NCRTCS - 2026

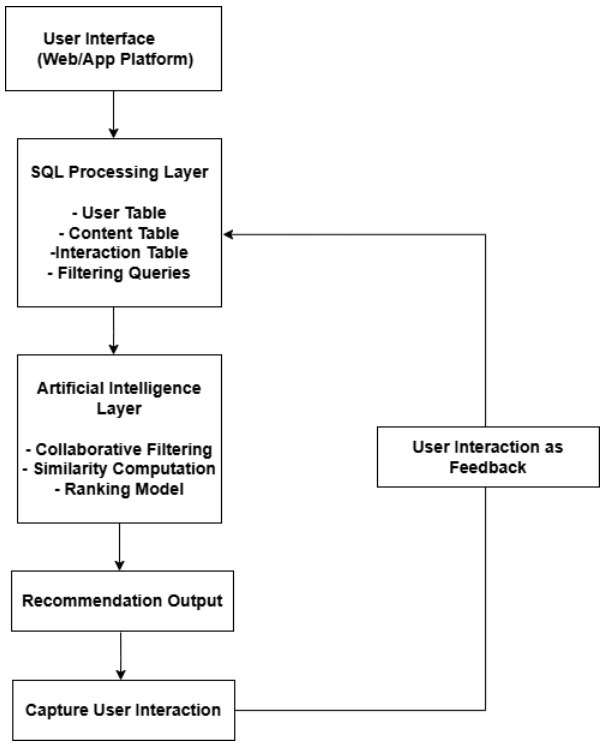

Fig 1. Workflow of Hybrid Framework

C. SQL-Guided Candidate Filtering

In this proposed framework, SQL is not just used for storage purposes, it facilitates recommendation by performing intelligent filtering depending on context-based constraints.

SQL queries fetches the recent user interations, find what genres of content are most frequently viewed, and filter out irrelevant content. Thus, it minimizes the search space for AI model's computations.

For eg:, specific recent viewing history is derived using temporal queries, while filtering via genres helps identify major user preferences. This filtered data acts as a prerequisite to feed into the learning model.

D. Adaptive AI-Based Ranking Model

After SQL-based filtering, candidate items are subject to a collaborative filtering operation. The similarity between users is defined based on watch time, interaction style, and patterns.

Implicit feedback data such as video views, likes, and additions to a watch list are used to compute a weighted preference score. This allows one to make an estimation of the level of interest users have in unseen videos.

A similarity-based ranking algorithm calculates a score for each of the relevant items. As each item gets a higher ranking, recommendations shift their order. It updates parameters using newly observed interactions.

IV. CASE STUDY IMPLEMENTATION

Inorder to validate the efficacy of the proposed hybrid SQL-guided AI-adaptive recommender system framework, a case study has been conducted on the real-world applications in OTT and e-commerce platforms using a simulated dataset.

In this study, SQL-based, AI-based, and hybrid-based approached have been tested and compared under samilar conditions.

A. Dataset Description

A synthetic dataset is created by using Python, which can effectively simulate a real-world scenario where users are interacting with digital streams. The data created has three major components: users, metadata, and interactions.

User data set contains user demographics such as age and subscription types. Similarly, content data set contains title types, genre types, and duration types. Lastly, the interaction data set stores user actions such as watching, liking, skipping, and adding items to the watch list together with the time stamps.

There are a total of 10 users, 15 content items, and 80 interaction records in the dataset.

TABLE 1. SUMMARY OF SIMULATED DATASET

Sr. No	Parameter	Value
1	Number of Users	10
2	Number of Content	15
3	Number of interactions	80
4	Number of Genres	5
5	Number of Action Types	4
6	Database System	SQLite

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 02, NCRTCS - 2026

B. Database Design and Implementation

For this, the data set implemented SQLite, and it is a relational DBMS. Accordingly, three normalized tables were designed: users, content, and viewing_interactions.

The users table stores profile data, the content table maintains metadata, and the viewing_interactions table records behavioral data. The primary and foreign key constraints were created to ensure referential integrity.

TABLE 2. DATABASE SCHEMA DESCRIPTION

Table	Attribute	Data Type	Description
users	user_id	text	Unique user identifier
users	age	integer	User age
users	subscription	text	Subscription type
content	Content_id	Text	Content identifier
Content	Title	Text	Content title
Content	Genre	Text	Content genre
Viewing_interactions	Action	Text	Interaction type
Viewing_interactions	Watch_time	Integer	Viewing duration
Viewing_interactions	Timestamp	text	interaction time

C. System Architecture and Workflow

The proposed recommendation system is implemented using a layered architecture using simulated OTT-based case-study as represented in Fig 2.

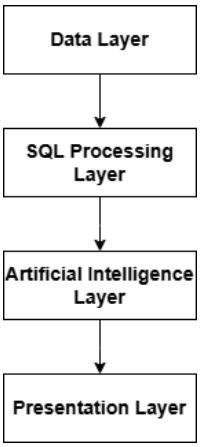

Fig 2. Layered architecture of Hybrid Framework

The data layer is implemented using SQLite and consists of three primary tables: users, content and viewing_interactions. These tables store user information, content metadata, interaction logs like watch time, action (skip, watch later, etc.), and timestamps.

The SQL processing layer performs preliminary data analsyis and candidate filtering. This stage reduces the search space and improves learning process.

The AI layer is implemented using python libraries such as Python and Numpy. A user-item interaction matrix is constructed from filtered records, and collaborative filtering is applied to compute user similarity. Based on these similarity scores, relevant content are predicted for unseen content items and ranked accordingly.

The presentation layer displays the generated recommendations in the form of ranked content lists, simulating real-world streaming platforms. This study does not involve the web interface of the presentation layer.

Fig 1 illustrates operational workflow of the system. Initially, user interactions like watching, skipping, linking, adding to watch later, etc. are simulated and stored in viewing_interactions table. SQL-based filtering is applied to form a reduced candidate dataset that highlights user's current preferences and behaviour. This candidate dataset is forwarded to AI module, where a user-item interaction matrix is constructed and cosine similarity is computed to identify similar users.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 02, NCRTCS - 2026

Users and ranked accordingly considering all these parameters.

The ranked recommendation list is presented in tabular form. Again, user interaction to these recommended items are recorded and appended to the interaction database as a feedback, hence enabling iterative refinement.

D. Implementation of SQL-Guided Filtering

SQL contributes to the process of recommendations by providing context-relevant information

The following filtering mechanisms are used in this theory.

1. Temporal Filtering
2. Preference-based Filtering
3. Interaction frequency Filtering

```sql
SELECT user_id,
       content_id,
       action,
       watch_time,
       timestamp
FROM viewing_interactions
WHERE timestamp >= datetime('now', '-7 days')
ORDER BY timestamp DESC;
```

Fig 3. Query for Time-Weighted Recent Activity

Fig 3 query retrieves the recent user sessions but maintains the time-ordering of the data. This helps the system focus on short-term user interests.

```sql
SELECT c.genre, SUM(
          CASE
              WHEN v.action = 'watch' THEN 2
              WHEN v.action = 'like' THEN 3
              WHEN v.action = 'skip' THEN -1
              ELSE 1
          END
      ) AS preference_score
FROM viewing_interactions v
JOIN content c
ON v.content_id = c.content_id
WHERE v.user_id = 'U1'
GROUP BY c.genre
ORDER BY preference_score DESC;
```

Fig 4. Query for Weighted Genre Preference Analysis

Fig 4 query assigns different weights to user actions while calculating genre preference score.

```sql
SELECT v.content_id,
       c.title,
       AVG(v.watch_time) AS avg_watch_time,
       COUNT(*) AS interaction_count,
       (AVG(v.watch_time) * COUNT(*)) AS engagement_score
FROM viewing_interactions v
JOIN content c
ON v.content_id = c.content_id
GROUP BY v.content_id
ORDER BY engagement_score DESC
LIMIT 5;
```

Fig 5. Query for Engagement-Based Content Ranking

Fig 5 query combines interaction frequency with average watch duration to identify highly engaging content.

```sql
SELECT DISTINCT v.content_id FROM viewing_interactions v
JOIN content c ON v.content_id = c.content_id
WHERE v.user_id = 'U1' AND c.genre IN (
    SELECT c2.genre
    FROM viewing_interactions v2
    JOIN content c2 ON v2.content_id = c2.content_id
    WHERE v2.user_id = 'U1' GROUP BY c2.genre
    ORDER BY COUNT(*) DESC LIMIT 2
) AND v.timestamp >= datetime('now', '-14 days');
```

Fig 6. Query for Context-Aware Candidate Selection

Fig 6 query selects candidate items of user's dominant genres within a recent temporal window.

The aforementioned queries, create a context-aware candidate set that considers short-term interests, long-term preferences and engagements. The resulting data is then passed to the artificial intelligence module, where similarity calculation takes place.

By carrying out a structure-based 'pre-filtering' in the database, computational costs are reduced and increase the relevance of the recommendation.

E. AI-Based Recommendation Generation

The filtered candidate dataset from the SQL processing layer is post-processed using various techniques of collaborative filtering, which is implemented in Python using the Pandas and NumPy libraries.

1) Construction of User–Item Interaction Matrix

A user–item interaction matrix is constructed, where rows represent users and columns represent content items.

Each matrix entry indicates the average watch time of a user for a particular content item.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 02, NCRTCS - 2026

If an interaction exists, the corresponding average watch time is recorded. Otherwise, the value is set to zero. Missing values are replaced with zeros to represent non-interacted items, resulting in a sparse interaction matrix.

2) Computation of User Similarity

For measurement of behavioral similarity between users, cosine similarity is applied to their interaction vectors.
This method aims to compare users based on their viewing habits and level of engagement with which the application has been utilized. Users who share similar interacting profiles are considered to be neighbor users in this approach.

3) Rating Prediction and Interest Estimation

For a target user and unseen content items, predicted interest scores are estimated using weighted aggregation of preferences from similar users.

Higher importance is assigned to users with stronger similarity values. This approach enables the system to infer potential interests based on collective viewing behavior.

4) Top-N Recommendation Generation

Based on the predicted interest scores, all candidate items are sorted in descending order of relevance. The top-ranked items are selected as personalized recommendations.

In this study, Top-5 and Top-10 recommendation lists are generated for performance evaluation.

F. Baseline Models for Performance Comparison

Three recommendation models were implemented for evaluation.

1) SQL-Only Model: The SQL-only model generates recommendations using rule-based queries based on popular genres and frequently accessed content. This model does not incorporate learning mechanisms.

2) AI-Only Model: The AI-only model applies collaborative filtering directly on the complete interaction dataset without SQL pre-filtering. It ignores contextual constraints.

3) Proposed Hybrid Model: The hybrid model integrates SQL-guided filtering with adaptive AI-based ranking. SQL reduces irrelevant candidates, while AI improves personalization.

G. Evaluation Metrics

System performance was evaluated using the following metrics.

1) Precision

$$\text{Precision} = \frac{\text{Relevant Recommendations}}{\text{Total Recommendations}}$$

2) Recall

$$\text{Recall} = \frac{\text{Relevant Recommendations}}{\text{Total Relevant Recommendations}}$$

3) Hit Rate

$$\text{Hit Rate} = \frac{\text{Sucessful Hits}}{\text{Total Users}}$$

4) Average Watch Time

Measures user engagement after recommendations.

H. Experimental Results

The three models were evaluated under identical conditions.

TABLE 3. PERFORMANCE COMPARISON OF RECOMMENDATION MODELS

Method	Precision	Recall	Hit Rate	Avg Watch Time (min)
SQL-only	0.61	0.58	0.64	45
AI-only	0.75	0.69	0.71	60
Hybrid	0.85	0.81	0.82	78

TABLE 4. SAMPLE RECOMMENDATION OUTPUT (HYBRID MODEL)

User ID	Recommended Content	Score	Rank
U1	Show_5	0.92	1
U1	Show_9	0.85	2
U2	Show_3	0.88	1

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 02, NCRTCS - 2026

I. Graphical Analysis

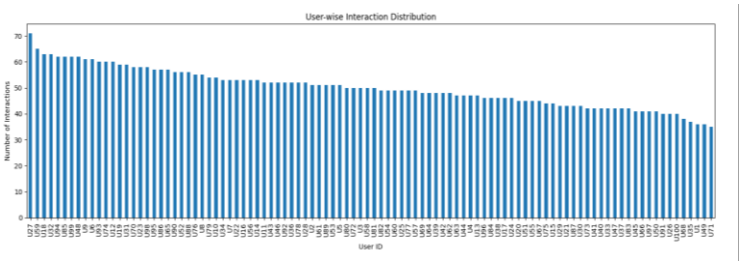

Fig 7. User-wise Interaction Distribution

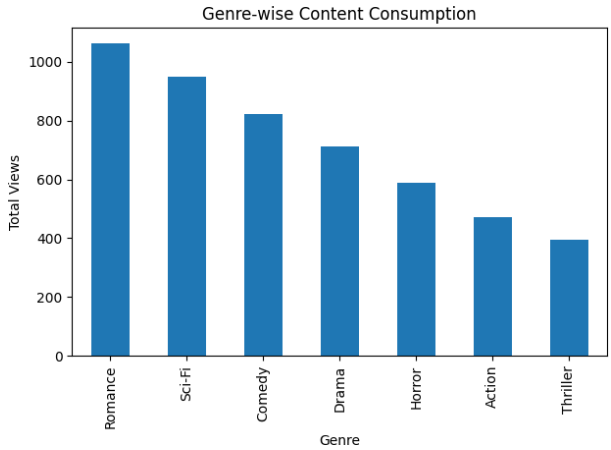

Fig 8. Genre-wise Content Consumption

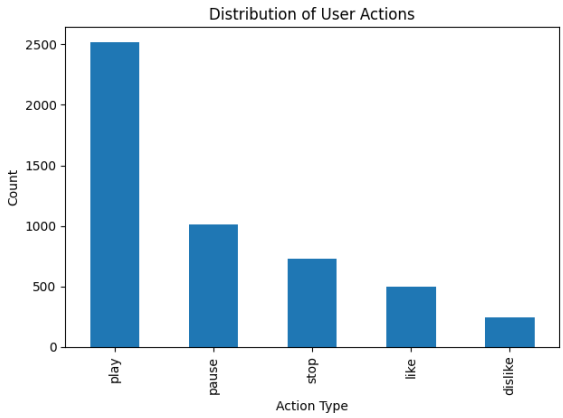

Fig 9. Distribution of User Actions

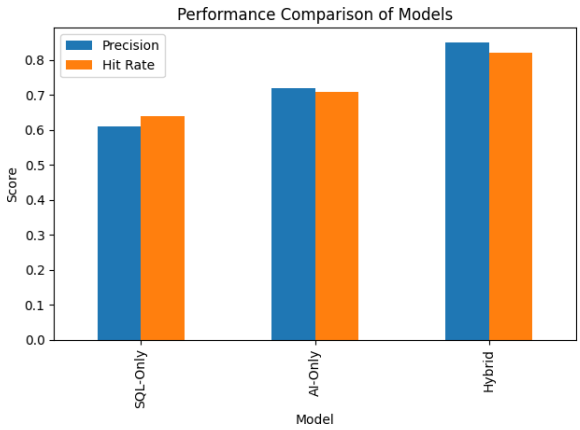

Fig 10. Performance Comparison of Models

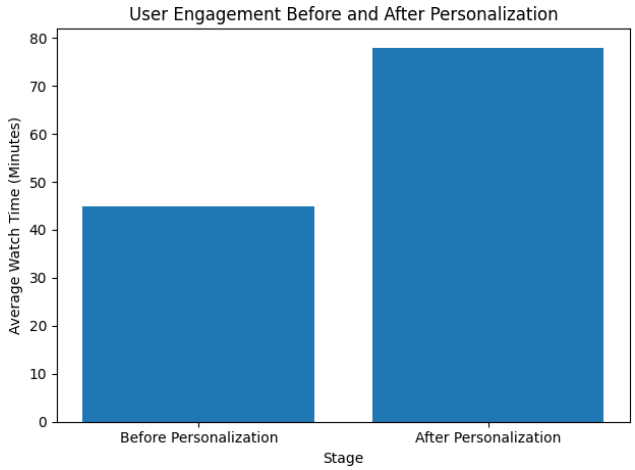

Fig 11. User Engagement Before and After Personalization

J. Results and Discussion

From the results of the experiments, it can be concluded that the proposed hybrid method performs better than both the traditional SQL method and the AI method under all evaluation parameters.

As presented in Table 3, it is observed that the hybrid model results in the highest precision and recall value, thereby improving recommendation relevance. An increase in the hit rate is also observed, thereby improving user satisfaction. While SQL-only approach has low adaptability due to its static property, AI-only approach has added noise resulting from unfiltered data.

This proposed framework reduces the number of irrelevant candidates and has enhanced accuracy in ranking candidates with its integration of SQL-guided filtering and adaptive learning. The feedback

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 02, NCRTCS - 2026

mechanism also helps in improving the personalization process.

The substantial increase in the average time spent watching every video is a measure of engagement and validates the practicality of the framework.

V. CONCLUSION

An SQL Guided Adaptive Artificial Intelligence Framework for context-aware personalized recommendation systems was introduced in this paper. To improve recommendation accuracy and relevance, the suggested hybrid model combines adaptive AI-driven ranking with structured SQL-based candidate filtering.

The effectiveness of this framework was also justified by a simulated case study in the form of OTT and e-commerce, in comparison to traditional SQL/AI solutions. Significant improvement in precision, recall, hit rate, and average watch time was observed by using this framework.

The SQL filtering mechanism was effective in discarding unwanted candidates, and the collaborative model was effective in improving the personalization of recommendations by allowing continuous learning. The feedback-based architecture allows this system to be adaptive in nature and easily applicable in real-time environments. Moreover, the intelligent database approach also enhances the explanation of generated recommendations.

Even though this framework enjoys some advantages, there are some disadvantages too. Currently, the framework implementation uses a smaller dataset generated artificially and might not necessarily fit larger and more realistic scenarios.

In addition, only collaborative filtering approaches have been used in the AI part of the framework, and future studies will emphasize incorporating the framework with large distributed databases and even real-time streaming systems.

Deep-Learning models like Neural Collaborative Filtering and even Reinforcement Learning models might be explored and introduced as well in order to optimize the long-term personalization of users.

REFERENCES

[1] E. Maruthavani and S. P. Shantharajah, “Real-time healthcare recommendation system for social media platforms,” IEEE Access, vol. 12, pp. 1–15, 2024, doi: 10.1109/ACCESS.2024.3393769.

[2] A. B. Suhaim and J. Berri, “Context-Aware Recommender Systems for Social Networks: Review, Challenges and Opportunities,” IEEE Access, vol. 9, pp. 57440–57463, 2021, doi: 10.1109/ACCESS.2021.3072165.

[3] N. Padhy, R. Sharma, A. Jain, and R. Kumar, “A recommendation system for e-commerce products using collaborative filtering approaches,” Eng. Proc., vol. 67, no. 1, p. 50, 2024.

[4] K. R. K. Bhaskar, D. Kundur, and Y. Lawryshyn, “Implicit feedback deep collaborative filtering product recommendation system,” arXiv preprint arXiv:2009.08950, 2020.

[5] Y. Wang and X. Wang, “FedPrivRec: A privacy-preserving federated learning framework for real-time e-commerce recommendation systems,” J. Adv. Comput. Syst., vol. 3, no. 5, pp. 63–77, May 2023, doi: 10.69987/JACS.2023.30506.

[6] M. Tibenský and M. Kompan, “Context-aware adaptive personalized recommendation using meta-hybrid approach,” Inf. Sci., vol. 560, pp. 344–358, 2021.

[7] Y. N. Silva, I. Almeida, and M. Queiroz, “SQL: From traditional databases to big data,” in Proc. SIGCSE Tech. Symp. Comput. Sci. Educ. (SIGCSE), Memphis, TN, USA, Mar. 2016, pp. 1–6.

[8] C. C. Aggarwal and J. Han, Data Mining, Cham, Switzerland: Springer, 2014. doi: 10.1007/978-3-319-07821-2.

[9] S. M. A. Rizvi, A. Malik, S. Rehan, S. Shukla, and S. Fatima, “Building a SQL based multi topic recommendation system using collaborative based filtering technique,” *International Journal of Research and Development in Applied Science and Engineering (IJRDASE)*, vol. 25, no. 1, 2025.

[10] A. Pavlo, M. Butrovich, A. Joshi, L. Ma, P. Menon, D. Van Aken, L. Lee, and R. Salakhutdinov, “External vs. internal: An essay on machine learning agents for autonomous database management systems,” *IEEE Data Engineering Bulletin*, vol. 42, no. 1, pp. 32–46, 2019.

[11] V. Panwar, “AI-driven query optimization: Revolutionizing database performance and efficiency,” International Journal of Computer Trends and Technology, vol. 72, no. 3, pp. 18–26, Mar. 2024, doi: 10.14445/22312803/IJCTT-V72I3P103.

[12] J. Zahir and A. El Qadi, “A recommendation system for execution plans using machine learning,” *Math. Comput. Appl.*, vol. 21, no. 2, p. 23, 2016, doi: 10.3390/mca2102002.

[13] D. Timoshenko, “Optimization of MLOps processes for product recommendation systems under high load,” *Universal Library of Engineering Technology*, vol. 3, no. 1, pp. 12–21, 2026, doi: 10.70315/uloap.ulete.2026.0301003.

[14] Q. Zhang, J. Lu, and Y. Jin, “Artificial intelligence in recommender systems,” Complex & Intelligent Systems, vol. 6, no. 3, pp. 1–26, 2020, doi: 10.1007/s40747-020-00212-w.

[15] A. Desku, B. Raufi, A. Luma, and B. Selimi, “Recommender system for software engineering using SQL semantic search,” International Journal of Engineering and

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 02, NCRTCS - 2026

Advanced Technology (IJEAT), vol. 11, no. 4, pp. 119–122, Apr. 2022, doi: 10.35940/ijeat.D3494.0411422.

[16] Saranya A. and Subhashini R., "A systematic review of explainable artificial intelligence models and applications: Recent developments and future trends," Data and Awareness Journal, 2023, doi: 10.1016/j.dajour.2023.100230.

[17] D. Li and Q. Gao, "Session recommendation model based on context-aware and gated graph neural networks," *Computational Intelligence and Neuroscience*, vol. 2021, Art. ID 7266960, pp. 1–10, 2021, doi: 10.1155/2021/7266960.

[18] R. M. Rifqi, Djupriadi, W. T. Haryanto, E. Utami, and A. H. Muhamad, "Hybrid recommendation system in e-commerce," *International Journal of Integrated Science and Technology (IJIST)*, vol. 3, no. 5, pp. 1983–1992, 2025, doi: 10.59890/ijist.v3i5.29.

[19] H. Hazem, A. Awad, and A. H. Yousef, "A distributed real-time recommender system for big data streams," *Ain Shams Eng. J.*, 2022, doi: 10.1016/j.asej.2022.102026.

[20] M. M Afsar, Trafford Crump, and Behrouz Far. "Reinforcement Learning Based Recommender Systems: A Survey." ACM Computing Surveys, vol. 1, no. 1, 2018, pp. 1–37. https://doi.org/10.1145/1122445.1122456

[21] E. Masciari, A. Umair, and M. H. Ullah, "A systematic literature review on AI-based recommendation systems and their ethical considerations," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3451054.

[22] E. Cano and M. Morisio, "Hybrid recommender systems: A systematic literature review," *Intelligent Data Analysis*, vol. 21, no. 6, pp. 1487–1524, 2017, ISSN: 1088-467X.

[23] X. Huang, J. Wang, and J. Cui, "A personalized collaborative filtering recommendation system based on bi-graph embedding and causal reasoning," *Entropy*, vol. 26, no. 5, Art. no. 371, 2024, doi: 10.3390/e26050371.

[24] D. Lavbic, T. Matek, and A. Zrnec, "Recommender system for learning SQL using hints," *Interactive Learning Environments*, vol. 25, no. 8, pp. 1048–1064, 2017, doi: 10.1080/10494820.2016.1244084.