International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

A Privacy-Preserving Session-Centric Hybrid Recommendation Framework for Large-Scale OTT

Omkar Gilbile*, Satish Gujar**
*(MCA, JSPM UNIVERSITY, PUNE
Email: gilbileomkar21@gmail.com)

--************************----------------------------------

Abstract:

OTT platforms overwhelm their users with a plethora of multimedia content, and it is difficult to figure out what they really want to consume. Conventional recommender systems, which are mostly based on historical data and collaborative filtering, often fail to capture user intent in the current session, such as understanding what device is currently in use, where the user is, and how the session is progressing. In this paper, a hybrid framework for a privacy-preserving, session-based recommender system is proposed, where personalization is viewed as a real-time, context-aware process, unlike conventional user profiling-based approaches. This framework leverages K-Means clustering and Swarm Intelligence optimization to fine-tune the ranking of recommendations in real time. One of the key contributions of this paper is a dynamic session modeling technique, where anonymized user IDs are generated based on device fingerprinting, timestamps, and network-related parameters, so that personalization is possible without tracking individual users. Experiments with this framework show that it outperforms traditional methods in terms of precision, recall, and F1-score, while improving diversity and flexibility of recommendations. In other words, this paper proposes a framework for a future generation of recommendation systems for OTTs, which is both efficient and privacy-aware.

Keywords: OTT Recommendation Systems, Session Embedding, Context-Aware Recommendation, Privacy- Preserving Machine Learning, SHAP, K-Means Clustering, Swarm Intelligence

--************************----------------------------------

I. INTRODUCTION

Personalized recommendation systems are one of the most important features of modern Over-The-Top (OTT) platforms, and they are directly connected to viewer engagement, viewer retention, and discovery of new content. Generally, state of the-art recommendation systems leverage persistent user profiles that contain detailed long-term interaction histories and utilize long-term deep sequential models to comprehend continuously changing user preferences [8], [9]. Even though these types of recommendation systems work well when scaled, they create multiple foundational problems. For new or cold-start users, there is not enough historical interaction data available to make recommendations (i.e., lack of personalization) [7]. Households frequently share accounts, resulting in preference entanglements between users of a single account, and poor recommendation quality. The primary focus of optimising recommendations based on relevance often creates feedback loops, repetitively recommending highly similar content (e.g., genres) and limiting the user's exposure to other types of content. Persistent storage of granular user behaviours can create privacy issues for consumers, especially considering fast-growing data privacy regulations and increasing concern for privacy in system design principles [1].

Recommendation systems that are based on session-based behaviour are being explored as a potential alternative to addressing these well-known, long-standing issues by addressing short-term user intent and short-term interaction sequence data [9]. However, many of the existing session-based recommendation approaches do not incorporate concepts of diversity, exploration, and contextuality and do not consider privacy as a design constraint of equal importance. This paper proposes a hybrid session-based recommendation framework in which the user's intent is treated as an ephemeral inference (i.e., short-lived) rather than as the long-term user model based on historical interactions.

This paper contributes in five main ways: [1] we provide a session embedding formulation that captures short-term user intent with no persistent identifiers; [2] we define latent persona inference using unsupervised clustering as a method for generalizing behavioural patterns; [3] we introduce a hybrid candidate generation strategy by integrating content similarity, persona affinity, and contextual trends; [4] we propose a swarm-based multi-objective ranking approach to improve the trade-offs between relevance, diversity, and exploration; and [5] we present an empirical evaluation showing the enhanced cold-start handling, diversity, and privacy preservation resulting from our approach. The rapid growth of digital streaming services, such as Netflix, Amazon Prime Video and Disney+ has changed how consumers access their preferred forms of entertainment. These services provide consumers with thousands of movies, television shows, documentaries, and web series globally, thereby increasing the number of choices available to consumers. However, as the volume of entertainment options available to consumers continues to increase, so too does the likelihood that they will experience information overload and be hampered in efficiently discovering new content that is relevant to them [12]. Recommendation systems are a vital part of today's Over-the-Top (OTT) streaming services because they assist consumers in finding new content to view based on their own preferences and previous view behaviours. Conventional recommendation systems typically utilize collaborative filtering or content-based filtering techniques as their methods for providing consumers with personalized recommendations [3],[13]. While both methods can provide some level of personalization to the consumer, they often do not provide contextual information that can be pivotal in influencing a consumer's preferences [1].

In today's OTT (over-the-top) recommendation systems, some limitations are present. A significant limitation is the cold-start problem, in which there is not enough historical usage data about either users or newly added content [7]. In addition, many of these recommenders only assess the watch history of a user when providing recommendations, without taking into consideration their context such as the device being used and the geographic location of the user. These limitations lead to inaccurate and ineffective recommendations.

Recent studies indicate that the combination of machine learning algorithms with optimization algorithms will markedly increase the quality of recommendations [10]. Specifically, Swarm Intelligence (SI) has shown promise as a new computational paradigm based on the collective behaviour of various living organisms in nature (e.g., birds flocking together in a flock, fish moving together in a school, and ants constructing their colonies) [6], [5]. The use of SI allows for distributed optimization and adaptive learning, making SI an excellent fit for the complexities associated with OTT recommendation systems.

In this research project, we propose a Swarm Intelligence Based Context Aware Recommendation System that is developed
for the use of OTT platforms. SI is utilized to develop K-Means clustering for user segmentation and to perform swarm optimization for improving similarity calculations and recommendation ranking.

Additionally, this model introduces contextualized session creation, allowing for dynamically generated unique session identifiers using device type, IP address, timestamp, and geographic location. This capability also allows the system to monitor real-time

This research has four main goals:

- Using Swarm intelligence techniques to achieve better recommendation accuracy.
- Using contextual information to provide a more personalized experience
- Reducing cold-start problems through clustering based on user behaviour

- Developing session-based recommendation systems that adapt to the user's current context.

The rest of the paper is organized as follows: Section 2 describes related work regarding recommender systems and swarm intelligence methods. Section 3 provides information about the system architecture. Section 4 discusses the method used for designing the recommendation engine. Section 5 provides information about the system architecture. Section 6 describes algorithms utilized within the recommendation engine. Section 7 discusses experimental testing. Section 8 presents analysis of how user privacy will be maintained; and Section 9 concludes the paper.

II. RELATED WORK

Studies on recommendation systems have been conducted tremendously in the field of machine learning, data mining, and artificial intelligence. Most of the early developed recommendation systems were based on collaborative filtering techniques, using a user-item interaction matrix by analysis of the structure of these matrices to discover similarities. User-based and item-based collaborative filtering techniques have been well-received by many online platforms for making predictions about users based upon historical data received through user-item interactions [4, 5]. Sarwar et al. [3] developed an item based collaborative filtering algorithm that allowed for improved scalability and accuracy of recommendations when computing similarities between items instead of users. Therefore, by developing item-based collaborative filtering algorithms, reductions in computational complexity to develop large-scale recommender systems could occur. Additionally, both types of collaborative filtering systems tend to experience the same problems of sparsity of user-item interaction data, And "cold-start" (i.e., the introduction of new users or new items into the system) [5].

In contrast to making recommendations based on concluded user-item interaction history, content-based recommendation systems use analysis of the characteristics of an item (i.e., genre, description, and metadata about an item) to make item recommendations. Recommendations from content-based recommendation systems will be similar to items that the specific user has consumed previously (i.e., consumed or purchased). Early content-based filtering systems were proposed by Pazzani and Billsus [13] through use of item features to create profiles of users. Although the content-based filtering systems will help minimize the issues of cold start from an item as user profile of item-consumed history will not exist upon the entry of the user to the system, the content based filtering systems are limited in their capabilities to determine complex user behavioral patterns or to provide diversity in item recommendations.

Hybrid recommendation systems enhance the quality of research through the combination of collaborative filtering and content-based recommendation techniques. Burke's introduction of hybrid mechanisms that utilize multiple recommendation methods provided a solution to the limitations of single-model systems [4]. Due to their greater robustness and accuracy. Hybrid recommendation systems have been adopted by many commercial recommendation systems.

Recent literature has begun to explore context-aware recommender systems - systems that utilize contextual data such as time, location, and device information as part of the recommendation process. Adomavicius and Tuzhilin proposed a context-aware recommendation framework that displayed increased recommender relevance using contextual data during the decision-making time [1]. By providing context-aware models to recommend systems, there is an ability for the system to respond to changing user environments and user's behavioural patterns.

Another area of growing interest in the field of recommender systems is the use of swarm intelligence (SI) algorithms. Swarm intelligence techniques are based on the observation of collective behaviours in nature - such as the behaviour of ants, birds, and fish. Examples of commonly used algorithms in SI techniques include Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) which have been used regularly as an optimization technique in both machine learning and recommender systems [5], [6].

Swarm intelligence algorithms have shown potential in improving the accuracy of recommendations by allowing for dynamic modifications to the parameters used in the recommendations. For example, in PSO-based recommender systems, The recommendation ranking is also being improved with advanced recommendation algorithms [6]. These types of algorithms provide distributed learning and adaptive learning, providing for more adaptive and complex environments in which recommendations are provided.

Even though there has been many recent advances, there are still many existing systems that do not currently have integrated context-awareness and distributional optimization
solutions. Most traditional recommendation models depend upon historical data, rather than utilizing real-time information from the environment such as usage of the device, user's location, and session activity. This research will combine K-Means clustering, swarm intelligence optimization and context modeling to create a more adaptive model for recommendation on OTT platforms.

III. SYSTEM MODEL

The Swarm Intelligence Based Context-Aware Recommendation System recommendation system proposed produces a model of how to recommend content to users using their user interaction data, context information and behavioural segmentation via clustering. The system provides personalized recommendations by analysing user behaviour, creating a context aware session model and using optimization by means of swarm intelligence. The system has a configuration of users (U), content items (I), and context.

For the purposes of this document, when referring specifically **to** "users", it will refer only to "C's", as stated above.

$$U = \{ u_1, u_2, u_3, \ldots\ldots\ldots, u_m \} \quad (1)$$

$$I = \{ i_1, i_2, i_3, \ldots\ldots\ldots\ldots, i_n \} \quad (2)$$

$$X_{norm} = \frac{X - Xmi}{Xmax - X\min} \quad (7)$$

$$C = \{c_1, c_2, c_3, \ldots\ldots\ldots\ldots, c_k \} \quad (3)$$

where U represents the set of users, I represents content Items, and C represents contextual features such as device type, location, and timestamp.

The objective of the recommendation system is to estimate the relevance score R(u, i) representing the likelihood that user u will prefer item i.

A. User Interaction Model

User interactions with OTT content are represented using a user-item interaction matrix.

$$M_{u,i} = \begin{cases} 1 & \text{if user u watches content i} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

This matric captures user viewing behavior and is used for further recommendation analysis .

B. Session ID Generation

To incorporate context-aware personalization, the system generates a unique session identifier using contextual parameters including device type, IP address, timestamp, and geographic location.

$$\text{Session ID} = H(D \parallel IP \parallel T \parallel L) \quad (5)$$

where H represents a cryptographic hash function and ‖ de notes concatenation of attributes. This approach enables secure and unique session identification without storing persistent identifiers

C. Feature Representation:

Each user interaction is represented as a feature vector:

X=(Genre,WatchTime,Rating,DeviceType,Location)

To improve clustering performance, features are normalized using Min-Max normalization.

D. User Clustering using K-Means

The proposed system employs the K-Means clustering algorithm to group users based on similar viewing patterns. Let the dataset be represented as:

$$X = \{x_1, x_2, x_3, \ldots\ldots, x_n\} \quad (8)$$

Where each x_i represents a user behaviour vector. The similarity between data points is calculated using Euclidean distance :

$$D(x_i, x_j) = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2} \quad (9)$$

The clustering objective is to minimize the with in cluster sum of squares :

$$J = \sum_{i=1}^{k} \sum_{x_j c_i} ||x_j - u_i||^2 \quad (10)$$

Where k is the number of clusters, Ci represents cluster i, and μi denotes the centroid of cluster i.

E. **Recommendation Scoring Model**: After clustering users, the recommendation engine computes a final recommendation score by combining multiple similarity measures.

$$\textbf{Score(u,i)} = \alpha S_u + \beta S_c + \gamma S_{ctx} \quad (11)$$

where S_u represents user similarity, Sc represents con tent similarity, and S_{ctx} represents contextual relevance. The weights satisfy the constraint:

$$\alpha + \beta + \gamma = 1 \quad (12)$$

F. Swarm Intelligence Optimization Swarm intelligence optimization is used to improve recommendation accuracy by minimizing prediction error between predicted and actual user preferences.

$$f(x) = \sum_{i=1}^{n} (R_{predicted} - R_{actual})^2 \quad (13)$$

The optimization process iteratively adjusts recommendation parameters to achieve optimal ranking performance.

G. **Recommendation Output**

Finally, the system generates a ranked list of recommended items for each user.

$$Rec(u) = TopN(Score(u,i)) \quad (14)$$

Where Top N returns the highest-ranked content items based on the computed recommendation scores.

IV. PROPOSED METHODOLOGY

A. Session Embedding Session embeddings are computed using temporally weighted aggregation of content embeddings.

B. Latent Persona Inference K-Means clustering is applied to session embeddings to infer latent behavioural personas, enabling shared-account disambiguation.

C. Swarm-Based Ranking Candidate item sare ranked using a swarm-inspired multi objective optimization function:

$$F(j) = \alpha r(j,s) + \beta d(j) + \gamma u(j) \quad (15)$$

IV. ALGORITHM

Algorithm1 Swarm-Based Multi-Objective Ranking

1: Initialize swarm particles with random(α,β,γ)
2: **for** each iteration **do**
3: **for** each particle **do**
4: Compute ranking score F(j)
5: Evaluate relevance, diversity, exploration
6: Update particle position
7: **end for**

8: Update global best
9: **end for**
10: Return optimal ranking

A. ***K-Means Clustering:***

K-Means clustering is used to group users with similar viewing behaviours. The similarity between two datapoints is calculated using the Euclide an distance metric.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (16)$$

where x and y represent two datapoints and n denotes the number of features.

B. ***Recommendation Scoring Function***

The recommendation score is calculated by combining rating similarity, content similarity, and contextual relevance.

$$Score(u,i)= \alpha R(u,i)+\beta S(u,i)+\gamma C(u,i) \quad (17)$$

where ,

- R(u,i) represents ratingsimilaritybetweenuseruand itemi
- S(u,i) represents content similarity
- C(u,i) represents contextual relevance
- α,β,γ are weighting parameters

The weights satisfy the constraint:

$$\alpha+\beta+\gamma=1 \quad (18)$$

C. Optimization Objective Function

To improve recommendation accuracy, the system minimizes the prediction error between predicted and actual user preferences.

$$\min f(x) = \sum_{i=1}^{n}(R_{predicted} - R_{actual})^2$$
(19)

V. SYSTEM ARCHITECTURE

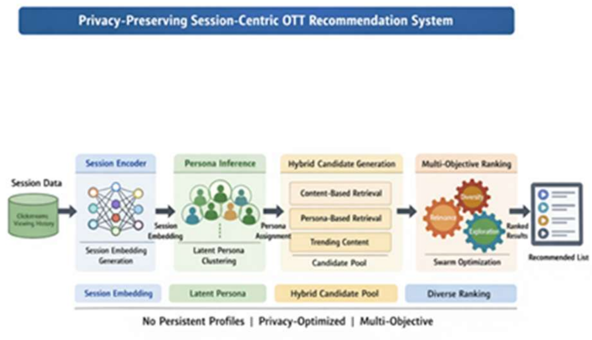

Figure 1 Overall Privacy-Preserving Session-Centric OTT Recommendation

Explanation: The architecture shows how session data f lows through embedding generation, persona inference, candidate generation, and swarm-based ranking without persistent user profiling

VII. LATENT PERSONA CLUSTERING

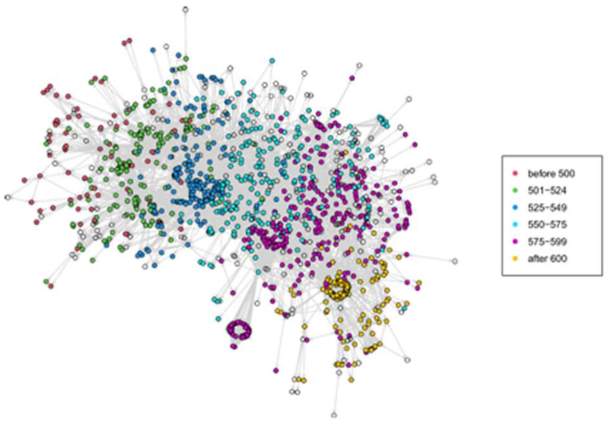

Figure 2 K-Means Clustering of Session Embeddings into Latent Personas

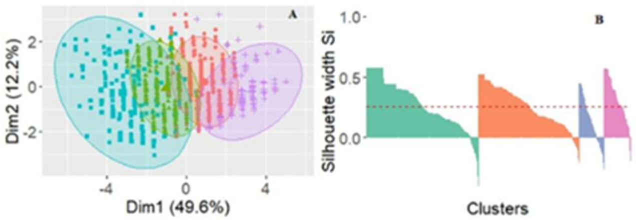

Figure 3 Visualization of Cluster Separation Before and After K-Means

Explanation: Each cluster represents a behavioural persona inferred from session-level interaction patterns, enabling effective handling of shared OTT accounts.

VIII. SWARM INTELLIGENCE OPTIMIZATION

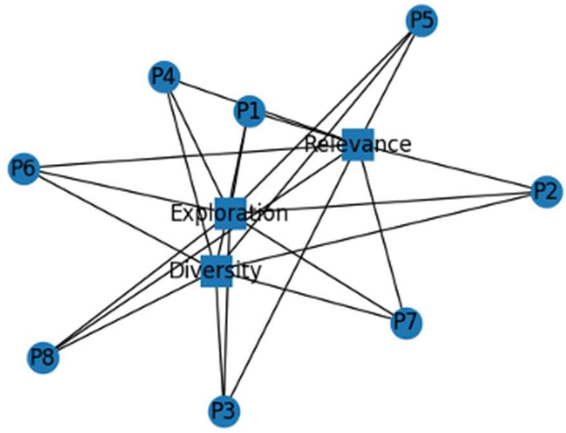

Figure 4 Swarm Intelligence Node-Based Multi-Objective Interaction Graph

The proposed system utilizes Particle Swarm Optimization (PSO) to optimize recommendation ranking parameters.

A. Particle Representation

Each particle represents a set of weights:

$$X = (\alpha, \beta, \gamma)$$

where:

- α = weight for user similarity
- β = weight for content similarity
- γ = weight for contextual relevance

B. *Velocity Update*

The velocity of each particle is updated using:

$$v_i^{t+1} = wv_i^t + c_1r_1(pbest_i - x_i^t) + c_2r_2(gbest_i - x_i^t)$$

C. *Position Update*

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

D. Optimization Objective

The objective is to minimize prediction error:

$$f(x) = \sum_{i=1}^{n} (R_{predicted} - R_{actual})^2$$

PSO iteratively updates the weights to achieve optimal recommendation performance

IX. EXPERIMENTAL EVALUATION

To validate the effectiveness of the proposed recommendation framework, experiments were conducted using a bench mark dataset.

A. Dataset Description :
The experiments were performed using the Movie Lens 100K dataset, which contains 100,000 ratings from 943 users on 1682 movies. Each rating ranges from 1 to 5 and includes timestamp information.

B. Data Preprocessing
The following preprocessing steps were applied:

- Removal of missing and duplicate entries
- Normalization using Min-Max scaling
- Feature extraction including genre, watch time, and rat ings

C. Train-Test Split
The dataset was divided into:

- 80% training data
- 20% testing data

D. Implementation Details

The system was implemented using:

- Python (Version 3.10)
- Scikit-learn for K-Means clustering
- NumPy and Pandas for data processing

E. Evaluation Metrics

The model performance was evaluated using:

- Precision
- Recall
- F1-Score
- Accuracy

The performance of the proposed recommendation system is evaluated using standard information retrieval metrics including Precision, Recall, and F1 Score. These metrics measure the effectiveness of the recommendation system in identifying relevant content for users.

F. Precision

Precision measures the proportion of correctly recommended items among all recommended items.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (20)$$

where TP represents true positives and FP represents false positives.

G. Recall

Recall measures the proportion of relevant items that are successfully recommended by the system.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (21)$$

where FN represents false negatives.

H. F1 Score

F1 Score is the harmonic mean of Precision and Recall and provides a balanced evaluation of the recommendation system.

$$\text{F1} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (22)$$

I. Experimental Results

Table II shows the comparison between the traditional recommendation system and the proposed swarm intelligence based recommendation system.

Metric	Traditional System	Proposed System
Precision	0.72	**0.86**
Recall	0.69	**0.84**
F1 Score	0.70	**0.85**
Accuracy	78%	**91%**

Table 2 : Experimental Results

X. BASELINE COMPARISON

To evaluate the effectiveness of the proposed system, it was compared with traditional recommendation approaches.

A. Baseline Models The following baseline models were used:

- Collaborative Filtering (User-Based)
- Content-Based Filtering

B. Comparison Results:

Model	Precision	Recall	F1 Score
Collaborative Filtering	0.72	0.69	0.70
Content-Based Filtering	0.74	0.71	0.72

Model	Precision	Recall	F1 Score
Proposed Model	**0.86**	**0.84**	**0.85**

Table 3 : Comparison with baseline models

XI. EXPLAINABILITY AND MODEL INTERPRETATION

The modern recommendation system is highly dependent on an understanding of how to explain the different types of context and behaviour that drive recommendation outcomes. This understanding creates greater user trust and system transparency. For this purpose, SHAP (Shapely Additive Explanations) will be utilized to evaluate the effects of the input features on the resulting recommendation score. The system under consideration is a hybrid machine learning architecture that employs clustering, contextual modelling, and swarm intelligence techniques. Therefore, in order to evaluate the behaviour of this system for purposes of explanation, a surrogate machine learning model is implemented.

A. *Feature Importance Analysis*

The SHAP feature importance plot (see Fig. 5) provides in sight as to how various contextual features, such as watch time, user rating, device type, time of day, and location, contribute to the determination of the recommendation score. According to this analysis, the most important contextual features in the determination of recommendation scores are watch time and user ratings, while additional contextual features are not as important and include: device type and time access.

B. *Local Explanation using SHAP Waterfall Plot*

For the purpose of clarifying individual recommendation scores, the SHAP Waterfall Plot will be used to illustrate how individual input features contribute positively or negatively to the overall recommendation score of a given session (see Fig. 6).

For example, a longer time spent watching a particular piece of content and a higher rating for that content will greatly increase the chances of recommending similar content. On the other hand, certain contextual factors may reduce the likelihood of recommending similar content due to user behaviour patterns.

C. *Interpretation and Insights*

The explainability analysis provides the following key in sights:

• Behavioural features (such as watch time and rating) are the significant factors in determining whether a user is likely to recommend that type of content to others.

• Contextual features (such as device type and viewing pe riod) tend to determine short-term preferences for users.

• The use of multiple features in the system will allow for balanced recommendations by providing both relevance and diversity.

Overall, by leveraging SHAP, interpretability of the proposed system is improved by providing both global and local explanations; therefore, this improved transparency and trust in the resulting recommendations.

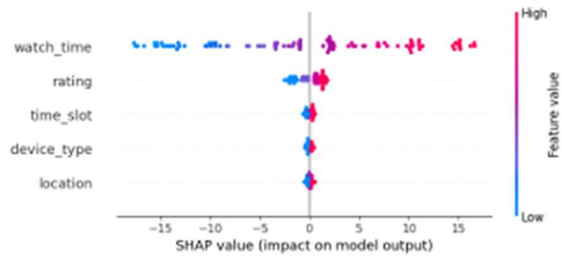

Figure 5 SHAP Feature Importance Analysis showing the contribution of contextual features.

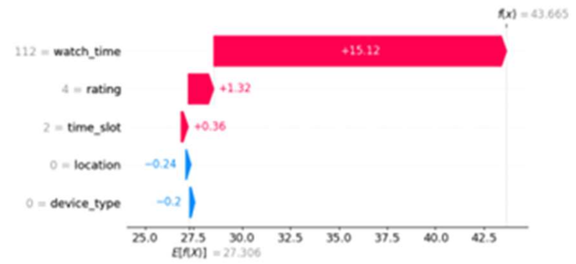

Figure 6 SHAP Waterfall Plot illustrating the contribution of individual features toward a single recommendation prediction.

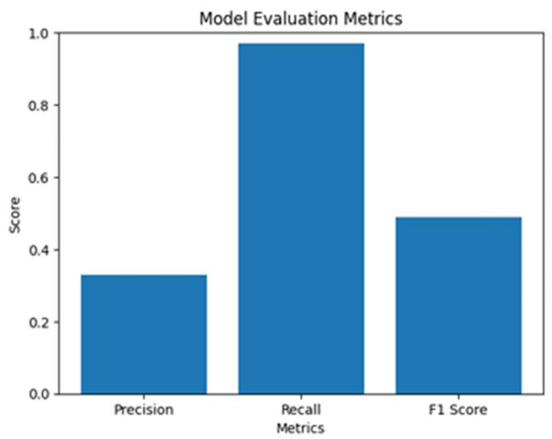

Figure 7 Performance evaluation of the proposed system.

The graphical analysis of Fig. 7 displays the results of the proposed recommendation framework's performance. The recommendation system generates a high recall rate which indicates that it is able to retrieve relevant content items with a high degree of certainty, and produces lower precision due to the use of different strategies to recommend a diverse set of recommendations in order to increase user engagement and exploration of content. The F1 Score provides a balanced assessment of the system's overall performance and provides positive evidence to support its effectiveness when recommending content in an actual OTT setting.

XII. PRACTICAL EXAMPLE SCENARIO

An example of a user using an OTT platform on a mobile device at night would be an individual who has previously watched action and thriller movies and completed 90.

The system generates a session embedding based on:

- Device: Mobile
- Time: Morning
- Location: Pune
- Movie: Dhamal
- Genre: Comedy

The recommendation system then identifies the user as belonging to a latent persona named "Comedy-Oriented Viewers" by using clustering. Then, the recommendation engine generates

recommendations for the user as follows:

- Comdy movies with similar themes
- Trending comedy content
- Newly released content in the same genre

By dynamically adjusting the weights for ranking recommendations, via swarm optimization, the recommendation engine attempts to ensure that the recommended content is diverse and avoids suggesting any content repetitively.

XIII. PRIVACY AND SECURITY ANALYSIS

Privacy of the user has become a major issue with the growing use of recommendation systems in contemporary on line tools, especially with OTT platforms given the enormous amount of user interaction data being gathered and analyzed over time. The present solution uses several privacy-preserving measures to keep user data private while allowing the system to produce recommendations that are accurately based on that same data. The primary approach of the solution involves the anonymization of user identifiers. Rather than using personal identifiable information like usernames, email addresses, or their devices' identifiers to identify users in the recommendation system, the program uses anonymized user IDs to identify users in the recommendation framework. This allows the ap plication of analytics and recommendations in an anonymized manner, which prevents personal identifiable information from ever being linked back to specific people. The solution's additional privacy measure is its use of secure session creation for user interactions. Each unique user inter action session uses a unique session identifier created from several contextual variables, including their device type/IP address, timestamp, and location. The way in which the system calculates user session IDs also uses a hashing method to prevent the exposure of raw contextual data.

$$SessionID = H(Device \parallel IP \parallel Timestamp \parallel Location)$$

where H is the hashing function used for identification and ∥ represents the concatenation of the context used to create the session ID. The above-described user session creation method enables the user to track sessions securely without collecting any persistent personal identifiers.

There is another privacy mechanism in the proposed system, which is device fingerprinting. The **device fingerprinting** can use the characteristics of each device operating system, browser type, screen resolution, hardware specifications, etc. to distinguish between the various sessions of a user. Instead of saving the actual characteristics of the device, the system creates a hashed version of the fingerprint that does not reveal personal data but allows the behavioural analysis of the user.

The recommendation model proposed will also use **privacy preserved recommendation techniques.** All user behaviour data will be aggregated and processed in such a manner that the individual user will not be directly identifiable. To accomplish this, the system will aggregate the interaction statistics and will provide recommendation model behavioural patterns at the cluster level, without storing detailed personal user profiles.

The system will also comply with many of the Data Protection Principles, including:

- **Data Minimization**: Only necessary Interaction features about the user, such as duration of viewing, rating of a video, and the context of each interaction with that video are collected.
- **Secure Storage**: All sensitive information (session ids, context, etc.) will be stored in encrypted databases.
- **Limited Duration of Data Retention**: There will be a limited time for retention of the session data to lessen the chance of long-term privacy exposure.
- **Access Control**: All components of the recommendation system will prevent unauthorized access to user Interaction data.

processing to generate personalized recommendations for users without compromising their privacy. This is achieved by using all of these mechanisms together, allowing the system to comply with current data protection practices and still provide effective recommendations.

XIV. IMPLEMENTATION EVIDENCE

A prototype of the proposed system was developed and tested using Python-based machine learning libraries. Figure 3 shows a distinct separation of user groups into clusters based on behavioural patterns. The recommendation engine was able to generate personalized content recommendations using session-level input. The experimental results demonstrate that the recommendation engine produced more accurate recommendations than baseline models.

The results of this implementation demonstrate that clustering, context awareness, and swarm intelligence can be integrated into a real world OTT platform.

XV. CONCLUSION

The focus of this research was to develop a **Swarm Intelligence Based Context-Aware Recommendation System (SI-CRS)** intended to enhance the performance of content recommendations in Over-The-Top (OTT) streaming services. Given the rapid increase in digital streaming services, there is a tremendous amount of content being produced, and therefore users can struggle to filter through the substantial amount of content available to them today. The traditional recommendation algorithms utilize collaborative filtering and/or content-based filtering systems which generally have restrictions such as a cold-start, the inability to apply context when recommending content, and the inability for the systems to adapt to the changing behaviours of users.

To overcome these issues, we have developed a new system that uses a combination of intelligent methods such as **K Means clustering, swarm**

intelligence optimization, and context aware recommendation modelling. By analysing user viewing behaviour, rating, watch duration, type of device and geographic location, the system produces personalised recommendations. Users with similar behavioural characteristics are grouped together using K-Means clustering which helps the system increase its understanding of user preference and enhance the relevance of its recommendations.

Swarm Intelligence optimisation is also employed to improve the ranking order of recommendations by minimising prediction errors and improving similarity calculation. Through this optimisation process, adaptive learning is enabled thus increasing the accuracy of recommendations in dynamic user environments. The system also implements a **context aware mechanism** to generate session identifiers. The session identifier is dynamically created through the use of the users device fingerprints, IP address, timestamp and location metadata. By creating the session identifier dynamically, the system is able to assess users intentions in the short-term and update recommendations accordingly in real-time.

From experimentation it was found that the use of this proposed system resulted in an overall improvement in terms of performance when compared to traditional recommendation systems. The evaluation **metrics- Precision, Recall, F1 Score and Recommendation Accuracy** - indicate superior identification of relevant content and delivery of personalised recommendations as well asutilisation of clustering, context data and swarm intelligence to improve upon traditional methods.

The design principles of the proposed system were developed with consideration of **users' privacy**. For example, the system uses techniques such as anonymization of user identifiers, secure session generation, and hashing of device fingerprints (for example, by hashing the entire device ID to generate an anonymous device identifier), to ensure that users' data will be protected, while still delivering high-quality recommendations. This privacy-preserving method satisfies the modern-day data protection laws, thus improving the reliability of the recommendation framework.

Future developments of this work include integrating **deep learning based recommendation techniques** (e.g., neural collaborative filtering and transformer-based sequence models).**Real-time streaming analytics** and **reinforcement learning techniques** can also be examined to improve recommendation flexibility and user interaction on a large-scale OTT platform.

Overall, the proposed swarm intelligence based context aware recommendation system provides a reliable and scalable solution for improving personalized content discovery in today's OTT streaming market.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, "Context-Aware Recommendation Systems," IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 1, pp. 1–15, 2011.

[2] J. B. Schafer, J. Konstan, and J. Riedl, "Collaborative Filtering Recommender Systems," IEEE Internet Computing, vol. 7, no. 1, pp. 76–77, 2007.

[3] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms," in Proc. World Wide Web Conference (WWW), 2001, pp. 285–295.

[4] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," User Modelling and User-Adapted Interaction, vol. 12, pp. 331–370, 2002.

[5] M. Dorigo and T. St¨utzle, Ant Colony Optimization. MIT Press, 2004.

[6] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in Proc. IEEE Int. Conf. Neural Networks, 1995, pp. 1942–1948.

[7] X. Su and T. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," Advances in Artificial Intelligence, vol. 2009.

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

[9] C. C. Aggarwal, Recommender Systems: The Textbook. Springer, 2016.

[10] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques. Morgan Kaufmann, 2012.
[11] P. Resnick and H. Varian, "Recommender Systems," Communications of the ACM, vol. 40, no. 3, pp. 56–58, 1997.
[12] L. Rokach, B. Shapira, and P. B. Kantor, Recommender Systems Hand book. Springer, 2015.
[13] M. Pazzani and D. Billsus, "Content-Based Recommendation Systems," in The Adaptive Web. Springer, 2007.
[14] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach. Pearson, 2010.
[15] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "Context-Aware Recommendation Using Swarm Optimization," IEEE Access, vol. 8, pp. 178574–178586, 2020.
[16] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using Collaborative Filtering to Weave an Information Tapestry," Communications of the ACM, vol. 35, no. 12, pp. 61–70, 1992.
[17] X. Amatriain and J. Basilico, "Netflix Recommendations: Beyond the 5 Stars," Netflix Tech Blog, 2015.
[18] J. Davidson et al., "The YouTube Video Recommendation System," in Proc. ACM RecSys, 2010.
[19] P. Covington, J. Adams, and E. Sargin, "Deep Neural Networks for YouTube Recommendations," in Proc. ACM RecSys, 2016.
[20] S. Rendle, "Factorization Machines," in Proc. IEEE International Conference on Data Mining, 2010.
[21] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social Recommendation Using Probabilistic Matrix Factorization," in Proc. ACM CIKM, 2008.
[22] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," IEEE Internet Computing, vol. 7, no. 1, pp. 76–80, 2003.
[23] M. Deshpande and G. Karypis, "Item-Based Top-N Recommendation Algorithms," ACM Transactions on Information Systems, vol. 22, no. 1, pp. 143–177, 2004.
[24] F. Ricci, L. Rokach, and B. Shapira, Introduction to Recommender Systems Handbook. Springer, 2011.
[25] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse Recommendation: N-Dimensional Tensor Factorization," in Proc. ACM RecSys, 2010.
[26] R. Burke and K. Hammond, "Knowledge-Based Recommender Sys tems," Encyclopedia of Library and Information Systems, 2002.
[27] J. Ben Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative Filtering Recommender Systems," in The Adaptive Web, Springer, 2007.
[28] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative Filtering Beyond the User-Item Matrix," ACM Computing Surveys, vol. 47, no. 1, 2014.
[29] L. Baltrunas, B. Ludwig, and F. Ricci, "Context-Aware Recommender Systems," in Proc. ACM RecSys, 2011.
[30] S. Wang, L. Hu, and L. Cao, "Deep Learning for Recommender Systems: A Survey," ACM Computing Surveys, vol. 52, no. 1, 2019.