Model Engineering College, Ernakulam
Department of Computer Engineering
B. Tech. Computer Science & Engineering
CSD334 MINI PROJECT
WatchWise
Literature Survey

January 9, 2025

22CSB20, MDL22CS086, GAYATHRI K BINOY
22CSB34, MDL22CS124, MEGHA B
22CSB43, MDL22CS142, NIHAR NIRANJAN S
22CSB46, MDL22CS148, NIRANJAY AJAYAN

1 Introduction

In today's fast-paced world, entertainment plays a crucial role in helping individuals relax, unwind, and cope with daily challenges. However, with the sheer volume of movies and TV shows available on streaming platforms, selecting content that aligns with a user's current emotional state can often feel overwhelming. This has given rise to the need for intelligent systems that personalize recommendations based on individual preferences and contexts.

Our project addresses this need by developing a mood-based recommendation system. Unlike conventional systems that rely solely on user watch history or ratings, our system takes a more dynamic and empathetic approach by first gauging the user's current mood. By prompting users to share how they are feeling, the system uses sentiment analysis techniques to interpret the user's emotional state.

The extracted mood data is then fed into a hybrid recommendation system that combines content-based filtering and collaborative filtering methods. The content-based approach analyzes the features of movies or TV shows, such as genre, cast, or storyline, to find matches, while the collaborative filtering method leverages the preferences and behaviors of similar users to suggest relevant content. This dual approach ensures a rich, personalized experience that adapts to both the user's immediate mood and broader preferences.

By incorporating mood analysis and hybrid recommendation strategies, this system aims to not only enhance user satisfaction but also create a more intuitive and meaningful interaction with entertainment platforms.

2 Articles

2.1 ”Intelligent Movie Recommendation System Based on Hybrid Recommendation Algorithms”(2023) by Qingna Pu

The Intelligent Movie Recommendation System Based on Hybrid Recommendation Algorithms combines content-based filtering (CBF), item-based collaborative filtering (Item-Based CF), and user-based collaborative filtering (User-Based CF) to enhance recommendation accuracy and efficiency. The system integrates Spark for big data processing, TensorFlow for deep learning, and Redis for fast data retrieval. Achieving an 81% accuracy rate and 70% movie coverage, it surpasses traditional recommendation methods by reducing search time and providing more personalized suggestions.

Implementation Details
The system architecture consists of four main layers: the front-end display layer for user interaction, the recommendation business layer for filtering and ranking movies, the model training layer utilizing deep learning techniques, and the data processing layer for managing historical and real-time user data. Users can engage with features such as preference selection, movie reviews, and a knowledge-based recommendation system. The Pearson correlation coefficient is used for similarity calculations, refining predictions based on user interactions. The system efficiently retrieves data using a tiered storage strategy, where Redis handles active movie and user features, while HDFS stores the complete dataset.

Advantages:
The hybrid system significantly improves recommendation accuracy, offering diverse movie suggestions while reducing user search time. The integration of Spark and TensorFlow ensures scalability, allowing the system to process large datasets efficiently. By using a knowledge-based recommendation approach, it enables users to explore personalized content while filtering out spam and malicious reviews. The combination of different filtering techniques ensures that recommendations are well-rounded, capturing both user preferences and item similarities.

Disadvantages:
Despite its advantages, the system faces high computational costs due to the complexity of hybrid algorithms. It relies heavily on accurate metadata and user ratings, which can impact performance if data quality is poor. The cold-start problem for new users remains a challenge, as recommendations rely on historical interactions. Additionally, the implementation complexity requires expertise in machine learning, deep learning, and big data technologies, making deployment more resource-intensive.

2.2 ”Movie Recommendation System Using Cosine Similarity (2024)” by Nilesh P. Sable

The paper focuses on developing a movie recommendation system that employs a content-based filtering approach combined with sentiment analysis. Unlike collaborative filtering, which often struggles with cold start and scalability issues, this system uses metadata such as cast, genre, and plot details, along with user reviews, to suggest movies. Cosine similarity is used as the primary technique to calculate the similarity between movies, and the recommendations are personalized to match user preferences. Supplementary information, such as movie ratings and reviews, is also provided to enhance user decision-making. The system integrates various data sources, including Kaggle datasets and the TMDB API, to ensure accurate and up-to-date information.

Implementation Details:
The implementation consists of six main steps: data collection, cleaning, API setup, sentiment analysis, vectorization, and recommendation generation. The datasets are collected from Kaggle and Wikipedia, then cleaned using Python tools like Jupyter Notebooks. The TMDB API is utilized to fetch additional metadata such as movie posters and cast information. Sentiment analysis is performed using the Natural Language Toolkit (NLTK) and machine learning methods like Naive Bayes, with reviews split into training and testing sets. The cosine similarity algorithm calculates the similarity between the search input and other movies in the database, producing a list of the top 10 recommendations. The system is hosted on a dynamic website with an intuitive GUI, ensuring user-friendly interaction.

Advantages:
The content-based approach avoids common collaborative filtering issues like data sparsity and cold start. Personalized recommendations are based on user preferences, making the system highly relevant. Sentiment analysis adds an extra layer of decision support by providing insights into user reviews. The integration of APIs ensures that the data is current and rich in details. The simple cosine similarity metric is computationally efficient and easy to implement.

Disadvantages:
The system may not perform well in cases where metadata is sparse or inconsistent across movies. Content-based filtering can lead to over-specialization, suggesting only similar movies and failing to explore diverse options. The reliance on textual metadata and sentiment analysis might miss nuanced user preferences not captured in text. Dynamic updates require continuous database maintenance, which could increase operational costs.

2.3 ”Movie Recommendation System Based on User Ratings and Critique” (2023) by K. Maheshan

This research delves into a novel hybrid approach that combines user ratings with critique-based feedback to construct a robust movie recommendation system. It focuses on improving user satisfaction by iteratively refining recommendations based on explicit user feedback, addressing limitations in traditional collaborative filtering and content-based filtering methods. By integrating critiques, the system creates an interactive recommendation loop, enabling dynamic updates tailored to evolving user preferences.

Implementation Details:
The system is built around the following components: a collaborative filtering algorithm that generates an initial set of recommendations by identifying patterns in user rating data, establishing a baseline preference profile for each user. A critique module facilitates user interaction by allowing explicit feedback on movie attributes, such as genre, director, cast, or themes. This module uses Natural Language Processing (NLP) techniques to process textual critiques. Additionally, an adaptive mechanism dynamically recalculates similarity scores by incorporating user critiques alongside historical ratings, ensuring real-time refinement of recommendations. Techniques like Term Frequency-Inverse Document Frequency (TF-IDF) and Latent Dirichlet Allocation (LDA) are used for thematic analysis of critiques.

Advantages:
This system actively incorporates user feedback to refine results, ensuring alignment with individual preferences. It enables real-time updates to recommendations, enhancing the user experience. The system handles diverse user preferences effectively, even in scenarios with sparse rating data, by combining qualitative and quantitative inputs. By leveraging NLP and topic modeling, the system provides deeper insights into user critiques, enriching the recommendation process..

Disadvantages:
The iterative refinement process, combined with NLP-based critique analysis, requires significant computational resources. The system relies heavily on active user engagement for critiques, limiting its efficacy for passive users. Over-reliance on specific critique attributes could skew recommendations, reducing diversity.

2.4 ”Movie Recommendation System Using Euclidean Distance(2024)” by Muneeb Sami Khan

This paper explores an alternative approach to movie recommendation systems by utilizing the Euclidean distance metric to measure similarity between users' preferences and movie genres. Data is preprocessed and analyzed to build two core matrices: one for top-rated movies in each genre and another for users' most preferred genres. The system computes Euclidean distances to generate personalized recommendations, offering insights into the relationship between user behaviors and genre-specific trends. By evaluating the accuracy of recommendations across varying thresholds and user counts, the study demonstrates the practicality of using Euclidean distance as a reliable metric for content personalization in movie recommendation systems.

Implementation Details:
The recommendation system is implemented using a systematic approach that begins with data collection, data preparation, and feature extraction. The process flow involves several key steps: counting the number of movies in each genre, normalizing user ratings, and identifying the user's top genre by analyzing normalized ratings. Concurrently, the system evaluates the highest-rated movies belonging to each genre by identifying those that contain the highest normalized ratings and ranking them in descending order. The viewing history of users is inspected to determine the genres they often watch, with those above average count being prioritized.
Two matrices are constructed: one capturing the top-rated movies by genre and the other representing users' top genres based on their viewing patterns. Euclidean distance is then applied between these matrices to measure the similarity between user preferences and movie genres, enabling the system to recommend the top 10 movies with the smallest distances. This methodology is implemented in Python using libraries such as Pandas, NumPy, and Scikit-learn, and its effectiveness is evaluated using metrics like Percentage Accurate Recommendation (PAR) across various thresholds and datasets.

Advantages:
This approach is straightforward and computationally efficient, leveraging the Euclidean distance for effective similarity measurement. It offers personalized recommendations tailored to users' genre preferences and is implemented using robust tools and libraries, making it accessible and adaptable. The use of normalized data and structured matrices ensures consistency and improves the reliability of recommendations. Additionally, the visualization of user and genre trends aids in better system analysis and fine-tuning.

Disadvantages:
The system faces challenges such as the cold-start problem, where new users or movies lack sufficient data for effective recommendations. Sparse datasets can also limit its performance, reducing the system's ability to detect meaningful patterns. The reliance on predefined attributes like genres might constrain the diversity and novelty of recommendations. Moreover, the accuracy of Euclidean distance diminishes in high-dimensional spaces, which may affect performance as the dataset grows larger or more complex.

2.5 "Collaborative Filtering-Based Movie Recommendation Services Using Opinion Mining" (2024) by Luong Vuong Nguyen

This paper introduces a movie recommendation system combining Collaborative Filtering (CF) and Opinion Mining for enhanced recommendation accuracy. CF constructs user-item matrices and calculates similarities based on user interactions. Opinion mining extracts sentiment scores for specific movie aspects such as acting, plot, and cinematography from user reviews. These scores are aggregated into overall ratings. A hybrid model integrates CF and sentiment-informed ratings for generating top-N recommendations. Techniques such as Matrix Factorization (MF), SVD++, Transnets, and MTER are employed for enhanced recommendations.

Implementation Details:
The system combines CF to compute similarities from user-item matrices with opinion mining to analyze user reviews and extract sentiment scores for key movie aspects like acting, plot, and cinematography. Matrix Factorization decomposes user-item interaction data into latent factors, while SVD++ incorporates implicit feedback for better predictions. Transnets use attention mechanisms to model complex user-item interactions, and MTER ensures a balance between accuracy and explainability. Together, these methods create a scalable, context-aware recommendation system.

Advantages:
The integration of qualitative user feedback (e.g., reviews and ratings) improves accuracy and personalization. It scales with increasing user interactions and utilizes textual data to address cold-start problems.

Disadvantages:
Data sparsity and cold-start issues persist for users or items with minimal interactions. Computational costs are higher due to hybridization. Privacy concerns arise from user-generated content mining. Opinion mining risks overfitting, resulting in narrow or biased recommendations, and sparse environments may limit recommendation diversity.

2.6 ”Research and Implementation of Movie Recommendation System Based on Knowledge Graph” (2023) by Lixia Luo

This research explores the integration of knowledge graphs in developing movie recommendation systems to enhance user experience by leveraging contextual relationships and semantic information. The research emphasizes the shift from traditional collaborative filtering and content-based filtering techniques to knowledge-graph-driven approaches that offer more accurate and dynamic recommendations. By utilizing entities and relationships within knowledge graphs, the system can connect user preferences with movies effectively, accounting for semantic richness and domain knowledge.

Implementation Details:
The implementation focuses on building a recommendation system based on a knowledge graph that extracts entities like movies, actors, directors, and genres. It constructs relationships such as ”acted in,” ”directed by,” and ”belongs to,” while utilizing algorithms for graph traversal and ranking, such as PageRank and graph neural networks. Key steps include knowledge graph construction using movie databases like IMDb, graph representation and embedding to reduce dimensionality while preserving semantic meaning, and the deployment of a recommendation algorithm based on similarity scores between the user’s profile and the knowledge graph entities.

Advantages:
This research paper explores the recommendation system (RS) and its three key elements: users, items, and transactions. RS is a thriving field with various machine learning algorithms, information filtering methods, and data mining techniques being employed. The paper highlights the provision of recommendations to users based on their preferences, whether personalized or generalized. It also emphasizes the impact of location preferences influenced by social, economic, and cultural factors. Online platforms often rely on user reviews to make informed decisions regarding movies, genres, and other services. It helped to understand concepts like Pre-Processing, Vectorization, Clustering better.

Disadvantages:
The article is less effective in comparison to the general approach of Ranking. Ranking algorithms normally put more relevant items closer to the top of the showing list, whereas recommender systems sometimes try to avoid overspecialization. Data sparsity, memory-based or model-based, both leverage a user’s historical interaction with the recommender systems. So for inactive users, recommendations may be very inaccurate. The method used would get less recognition for new movies than existing ones.

2.7 ”Movie Recommendation System using RNN and Cognitive Thinking (2023)” by Shubhada Labde

The paper focuses on building a movie recommendation system by integrating multiple approaches: SVD (Singular Value Decomposition) with Collaborative Filtering, content-based filtering, popularity-based methods, and Recurrent Neural Networks (RNNs). It also incorporates cognitive thinking to enhance user engagement and personalization. The system constructs a user-item interaction matrix for collaborative filtering, factors the matrix into user, feature, and item matrices, and generates personalized recommendations. Content-based filtering gathers data on movies using features like genre, director, and ratings to compute similarities and match user preferences. Popularity-based methods identify top-rated movies using user-rating matrices. The RNN model preprocesses sequential user-rating data to predict unseen movie ratings. A hybrid model combines all techniques to deliver enhanced recommendations.

Implementation Details:
The implementation integrates SVD and CF to decompose user-item interaction matrices into user, feature, and item components. Content-based filtering uses similarity metrics like cosine similarity and Euclidean distance to align movies with user preferences based on metadata. Popularity-based filtering identifies top-rated movies by analyzing user-rating matrices. The RNN model preprocesses sequential user-rating data to predict unseen ratings. The hybrid model operates all techniques in parallel, delivering highly personalized recommendations.

Advantages:
The research integrates RNNs, making it well-suited for temporal patterns and sequential data. By mimicking human cognitive processes, the system enhances personalization and user satisfaction. It effectively combines multiple approaches for an optimized recommendation process.

Disadvantages:
The implementation is complex, requiring significant computational resources and expertise. Overfitting risks are higher due to the RNN model’s dependency on large and diverse datasets. The hybrid system is challenging to interpret, and the generalizability of the approach to other domains requires substantial re-tuning.

2.8 ”The Construction of Movie Recommendation System Based on Python(2021)” by Qin Xu

This paper presents a movie recommendation system built on a Browser/Server (B/S) framework, using Python as the primary development language and MySQL as the database. The system is designed to address the challenges of information overload and personalized content discovery. By leveraging web crawlers to extract movie data from Douban.com, the system integrates user-based collaborative filtering algorithms to analyze users' preferences and recommend relevant movies. Through its modular design and focus on user behavior, the system provides a robust and scalable solution for enhancing the movie discovery experience. This work highlights the application of advanced algorithms and efficient database structures to develop a user-friendly recommendation platform.
Implementation Details :
The movie recommendation system utilizes a Browser/Server (B/S) framework and MySQL database for managing and storing data. Python is used as the core development technology, integrating web crawlers to extract movie data from Douban.com. These crawlers, built with urllib2 and BeautifulSoup, traverse through webpages to identify and parse relevant information, which is then stored in a structured format within the MySQL database.
At the heart of the recommendation engine is the user-based collaborative filtering algorithm. This method identifies users with similar interests by analyzing their favourite movie categories and collections. Similarity is calculated using the cosine similarity formula, which considers both the overlap and quantity of preferred categories or movies. Recommendations are generated by analyzing the preferences of users with high similarity to the target user, ultimately providing tailored movie suggestions. The database design supports this process through key tables such as “moviefavorite” and “favoritecategory”, which store data on user preferences and movie details. The recommendation process involves two major steps: first, identifying users with similar tastes, and second, using their favorite movies to suggest titles not already explored by the target user. The system calculates interest levels for each candidate movie based on weighted similarity scores, ranking the results to recommend the most relevant titles.

Advantages:
One of the key advantages of the system is its ability to personalize recommendations based on user preferences and historical behaviors. This personalization enhances user engagement, encouraging further exploration and interaction with the platform. The collaborative filtering algorithm ensures scalability, making it suitable for handling larger datasets as the system grows.
The integration of web crawlers automates the process of data collection and updates, ensuring the database remains current with minimal manual intervention. Additionally, the algorithm's focus on leveraging user data results in more accurate and relevant movie recommendations, further improving the user experience.

Disadvantages:
Despite its strengths, the system faces some challenges. The cold-start problem is a significant limitation, as new users or items lack sufficient data to generate meaningful

recommendations. This issue reduces the system's effectiveness in scenarios with limited historical interactions.

Computational overhead is another challenge of collaborative filtering. As the number of users increases, the computation time to calculate similarity becomes resource-consuming and, therefore, may degrade overall system performance. Furthermore, the system is highly dependent on historical data, and the sparsity of user interactions makes it challenging to identify actionable patterns.

Another drawback is the limited diversity in recommendations. By focusing on user preferences, the system might overlook opportunities to introduce novel or diverse content, leading to a narrower user experience. These challenges highlight areas where the system could be further optimized to improve overall performance and user satisfaction.

2.9 "Movie Recommendation using Metadata based Word2Vec Algorithm (2018)" by Yeo Chan Yoon

The paper introduces a novel movie recommendation system that utilizes metadata-based Word2Vec embeddings combined with user preferences to improve recommendation accuracy. By embedding metadata such as movie tags, directors, and genres into vectors, the system captures intricate relationships between movies. It also employs a deep learning approach to process user preferences derived from ratings and viewing history. The proposed method addresses common challenges in recommendation systems, such as the cold-start problem and poor performance for newly released items. Experimental results show a performance improvement of 0.165 in Recall@100 compared to baseline methods.

Implementation Details:
The system leverages the Word2Vec algorithm to embed movie and metadata into vectors, using training data based on user ratings and viewing history. Metadata embeddings are pretrained using co-occurrence relationships within movie datasets. User vectors are computed by aggregating weighted embeddings of movies a user has watched, with weights based on viewing time. The system ranks movies for recommendations using the inner product of user and movie vectors. The model's efficacy was evaluated using the MovieLens 10M dataset, which includes over 10,000 movies and 10 million user ratings. Compared to standard Item2Vec and Singular Value Decomposition (SVD) methods, the proposed approach demonstrates superior performance, particularly in handling new or infrequently rated movies.

Advantages:
Effectively addresses the cold-start problem by leveraging metadata for new and unrated movies. Captures complex relationships between movies through deep learning-based vector embeddings. Outperforms traditional collaborative filtering methods in Recall@N metrics. Scales efficiently with large datasets due to the dimensionality reduction capabilities of Word2Vec.

Disadvantages:
Dependence on metadata quality may limit performance in cases of incomplete or noisy metadata. User's watch time history is required for calculating the user vector. Computational overhead during the embedding and training process may increase with large and diverse datasets. The inner product similarity measure might oversimplify user preferences compared to more sophisticated ranking algorithms. While effective in movie recommendation, the model may need additional tuning for cross-domain application

3 Tables

Table 1: Summary of publications

Publication Year	**Methodology**	**Disadvantage**
2024	Sentiment analysis is performed on reviews and cosine similarity algorithm calculates similarity between movies.	Results in over-specialization and fails to explore diverse options.
2024	Euclidean distance for movie recommendations, focusing on genre-specific user preferences.	Lower accuracy compared to hybrid approaches, limited scalability for larger datasets.
2024	Collaborative Filtering combined with Opinion Mining using MF, SVD++, Transnets, and MTER for movie recommendations.	Data sparsity, cold-start issues, higher computational costs, privacy concerns with review mining, and overfitting risks leading to biased recommendations.
2023	Integration of knowledge graphs, graph traversal algorithms (e.g., PageRank), graph neural networks.	Less recognition for new movies, challenges in handling inactive users, high complexity in ranking.
2023	Hybrid approach combining collaborative filtering, critique-based feedback, LDA, TF-IDF.	High computational demand, reliance on active user engagement, potential bias in critique emphasis.
2023	Integration of SVD, content-based filtering, RNNs, and popularity-based methods for hybrid movie recommendations.	High computational complexity, overfitting risks with RNNs, interpretability challenges, and limited generalizability to other domains.
2021	Python-based recommendation system using collaborative filtering and web crawlers.	High dependency on user history, inability to handle real-time feedback effectively.
2018	Ranks movies for recommendations using the inner product of user and movie vectors.	Limited performance with incomplete or noisy metadata.

4 Problem Statement

With the abundance of streaming options available today, users often find it challenging to choose a movie that suits their current mood. The overwhelming number of choices can lead to decision fatigue, making it difficult to settle on a movie. Additionally, existing recommendation systems primarily focus on genres, popularity, or past viewing history, failing to understand and cater to an individual's emotional state at a given moment. This gap results in users spending excessive time searching for a suitable movie, which can diminish the overall enjoyment of the experience. A mood-based movie recommendation system can address this issue by analyzing a user's emotions and suggesting films that align with their feelings, making the selection process more intuitive, efficient, and personalized.

5 Proposed Solution

Our proposed solution is a mood-based movie recommendation system designed to personalize movie suggestions based on the user's emotional state. By integrating mood detection with collaborative filtering based on user ratings, this hybrid approach ensures more accurate and emotionally aligned recommendations. The system operates through two primary components: content-based filtering, which analyzes mood to suggest suitable movies, and collaborative filtering, which leverages user ratings to refine recommendations. By fusing the results from both methods, the system generates an optimized movie list tailored to the user's current emotions, enhancing their overall viewing experience.

6 Conclusion

The literature survey explored the role of hybrid recommendation algorithms in enhancing personalized content delivery for users. This study implemented a mood-based movie recommendation system that leverages a hybrid recommendation approach, combining collaborative filtering and content-based filtering. The system effectively analyzes users' mood data to suggest movies that align with their emotional state, enhancing user satisfaction and engagement. While the approach shows promising results in improving recommendation accuracy, limitations were noted in capturing complex mood variations and diverse content preferences. Future work could focus on incorporating advanced emotional recognition technologies and real-time mood analysis to further refine recommendation precision and user experience.

References

[1] Q. Pu and B. Hu, ”Intelligent Movie Recommendation System Based on Hybrid Recommendation Algorithms,” in *Proceedings of the 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE)*, 2023, pp. 1–10.

[2] N. P. Sable, A. Yenkikar, and P. Pandit, ”Movie Recommendation System Using Cosine Similarity,” in *Proceedings of the 2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, 2024, pp. 1–5.

[3] K. Mahesha, B. T. G. S. Kumara, and K. Banujan, ”Movie Recommendation System Based on User Ratings and Critique,” in *Proceedings of the 2024 4th International Conference on Advanced Research in Computing (ICARC)*, 2024, pp. 218–222.

[4] M. S. Khan, Z. Hussain, M. I. Amaad, M. Z. Hasan, A. Khalid, S. H. Chuhan, R. Awan, M. A. Yaqub, and F. Ashraf, ”Movie Recommendation System Using Euclidean Distance,” in *Proceedings of the 2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0*, 2024, pp. 1–7.

[5] L. V. Nguyen, ”Collaborative Filtering-based Movie Recommendation Services Using Opinion Mining,” in *Proceedings of the 2024 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*, 2024, pp. 1–5.

[6] L. Luo, Z. Huang, and Q. Tang, ”Research and Implementation of Movie Recommendation System Based on Knowledge Graph,” in *Proceedings of the 2023 4th International Conference on Computer, Big Data and Artificial Intelligence (ICCBD+AI)*, 2023, pp. 595–599.

[7] S. Labde, V. Karan, S. Shah, and D. Krishnan, ”Movie Recommendation System using RNN and Cognitive Thinking,” in *Proceedings of the 2023 4th International Conference for Emerging Technology (INCET)*, 2023, pp. 1–7.

[8] Q. Xu and J. Han, ”The Construction of Movie Recommendation System Based on Python,” in *Proceedings of the 2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, vol. 2, 2021, pp. 208–212.

[9] Y. C. Yoon and J. W. Lee, ”Movie Recommendation Using Metadata Based Word2Vec Algorithm,” in *Proceedings of the 2018 International Conference on Platform Technology and Service (PlatCon)*, 2018, pp. 1–6.