Journal of Applied Informatics Research Vol. 01 No. 01 (2025) pp. 10~20

Journal of Applied Informatics Research
Faculty of Vocational Studies, Unesa Campus 1, Ketintang, Surabaya, Indonesia
Website:https://journal.unesa.ac.id/index.php/jair/| Email: jair@unesa.ac.id

Optimization of Personalized Fashion Recommendations for H&M: A Collaborative Filtering Algorithm Approach with Temporal Time Interval Analysis

Moch Deny Pratama[1], Dimas Novian Aditia Syahputra[2], M Adamu Islam Mashuri[3], Binti Kholifah[4], Rifqi Abdillah[5], Adinda Putri Pratiwi[6]

[1,2,3,4]Department of Informatics Management, Universitas Negeri Surabaya, Indonesia
[5]Department of Informatics Engineering, Universitas Negeri Surabaya, Indonesia
[6] Department of Informatics Engineering, Institut Teknologi Sepuluh Nopember, Indonesia

[1]mochpratama@unesa.ac.id, [2]dimassyahputra@unesa.ac.id, [3]mmashuri@unesa.ac.id, [4]bintikholifah@unesa.ac.id, [5]rifqiabdillah@unesa.ac.id, [6]adindaputripratiwi97@gmail.com

ARTICLE INFORMATION

Article history:

Received July 25, 2025
Revised July 28, 2025
Accepted July 29, 2025

Keywords:

Recommendation System;
Collaborative Filtering;
Fashion Personalization;
Cosine Similarity;
Temporal Segmentation

ABSTRACT

This study presents a personalized fashion recommendation system for the H&M dataset, utilizing a cosine similarity-based collaborative filtering algorithm. This study investigates the effect of temporal segmentation on recommendation performance by conducting three experiments using datasets divided into two-week, one-month, and two-year time intervals. The experimental results show that the two-year interval achieves the best performance, producing a Mean Average Precision (MAP) of 0.02254 with a computational time of 2741.7 seconds. In contrast, the two-week interval achieves a MAP of 0.00915 in 1609.2 seconds, while the one-month interval produces a MAP of 0.00554 with a computational time of 3118.9 seconds. The main contribution of this study lies in the optimization of data structure transformation through dictionary-based modeling, which significantly improves training efficiency. These findings underscore the crucial role of temporal granularity in improving the accuracy and computational efficiency of collaborative filtering-based personalized fashion recommendation systems.

1. INTRODUCTION

The rapid evolution of global e-commerce has fundamentally restructured the retail industry, changing not only how consumers access products but also how businesses deliver value. In this transformation, the fashion sector has emerged as one of the most dynamic, complex, and profitable verticals, driven by ever-changing consumer tastes, seasonal trends, and cultural influences. The proliferation of internet penetration, widespread adoption of mobile devices, seamless integration of digital payment systems, and a shift in consumer behavior towards on-demand personalization have collectively acted as powerful growth drivers in the fashion retail segment. According to global market trends, online fashion retail alone has reached a valuation of nearly USD 406 billion, with an impressive compound annual growth rate (CAGR) of 18% [1]. The United States, one of the world's largest digital retail markets, recorded online sales of USD 414 billion in 2018, contributing significantly to the increase in global e-commerce revenue from USD 2.29 trillion in 2017 to approximately USD 4.48 trillion in 2021 [2]. This exponential growth underscores the strategic importance of implementing intelligent systems to manage user preferences and product diversity at scale. The inherently visual, stylistic, and personal nature of fashion products, including clothing, accessories, and footwear, poses a unique set of challenges for online recommendation systems. Unlike books or electronics, where specifications are often objective and comparisons straightforward, fashion choices are closely tied to subjective user preferences, cultural identities, and rapidly changing trends. This results in an ever-expanding and diverse product vocabulary that can overwhelm users and challenge traditional filtering mechanisms [3]. Consequently, intelligent systems, particularly recommendation engines, have become an indispensable tool for fashion e-commerce platforms. These systems help consumers navigate vast inventories by tailoring product suggestions based on their inferred preferences, thereby reducing cognitive load and enhancing the shopping experience [4]. However, despite advances in algorithmic recommendation engines, fashion-specific platforms often face limitations in delivering contextually relevant and truly personalized output. Challenges such as data scarcity, the cold-start problem, and the intrinsic subjectivity of fashion perception remain key obstacles [5].

However, the increasing availability of user-generated data has opened new avenues for improving recommendation quality. From transaction history, product ratings, and click-through behavior, to textual reviews and search queries, e-commerce platforms now have access to both explicit and implicit feedback signals. This richness enables more sophisticated modeling of user behavior, allowing systems to move from surface-level matching to personalized inference models [6]. Similarly, the evolution of recommendation system methodologies from rule-based approaches to machine learning-based architectures has enabled greater generalization and adaptation across user segments [7]. Today's state-of-the-art recommenders are algorithmic systems capable of learning dynamic user preferences through the integration of historical interactions and content-based metadata, further refined with real-time behavioral analysis [8]. In the fashion world, personalization is not a luxury but a necessity. Purchasing decisions are heavily influenced by aesthetic alignment, perceived style fit, peer trends, and social validation mechanisms such as likes, reviews, or influencer endorsements [9]. Large-scale platforms such as Amazon, Netflix, and Facebook have demonstrated how personalized recommendation systems can significantly impact user retention, engagement, and conversion through adaptive content delivery [10]. In fashion e-commerce, recommendation systems enable significant reductions in search friction, increase product relevance, and contribute to user satisfaction. These systems typically leverage data points such as browsing history, product metadata, ratings, and behavioral patterns to tailor recommendations without violating personal privacy [11].

Among various algorithmic strategies, Collaborative Filtering (CF) has gained considerable traction due to its domain independence and ability to discover latent patterns in user-item interactions. CF works by analyzing the preferences of a group of users or items and identifying correlations to predict future interest, without relying on content-specific attributes [12]. Widely adopted in the industry, CF has become fundamental in services such as Amazon's "Customers who bought this also bought" and Netflix's rating-based movie suggestions [13]. CF is broadly classified into two types: User-Based Collaborative Filtering (UBCF), which identifies similarities among users, and Item-Based Collaborative Filtering (IBCF), which computes item similarity based on users' shared interaction history [14]. The computational core of CF typically involves the construction of a user-item interaction matrix, followed by the application of a similarity metric such as cosine similarity, Pearson correlation, or Jaccard index to generate a recommendation score [15]. The main advantage of CF lies in its ability to evolve over time, adapting dynamically as new user feedback is gathered. However, its reliance on sufficient historical data makes it vulnerable to cold-start conditions and matrix sparsity, which often limits its application to newer or niche segments [16].

To mitigate these issues, the recommendation system literature has proposed various enhancements. Techniques such as hybrid models that integrate content-based features with collaborative patterns have been shown to reduce sparsity and enhance personalization. Other approaches include context-based recommendations, temporal modeling, matrix factorization, and clustering, each aiming to improve performance metrics or reduce computational complexity [17]. More recently, unsupervised learning methods such as k-means, latent semantic analysis, or autoencoders have been combined with CF to better capture multidimensional user-item relationships, especially in large-scale or high-dimensional datasets [18]. Furthermore, contextual similarity measures, such as triangle similarity, have been introduced to more precisely quantify user-item distances in complex interaction spaces [19]. These improvements have resulted in significant improvements across evaluation metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), which have become standard benchmarks in evaluating the performance of recommendation systems [20]. Building on this evolving landscape, this study proposes a collaborative filtering model that utilizes cosine similarity as its core similarity metric, while incorporating novel explorations into temporal granularity.

This study investigates how varying the time intervals of historical user-item interactions, comparing two-week, one-month, and two-year periods, impact the system's predictive accuracy and computational efficiency. This addresses a critical gap in the CF literature, where the dynamics of temporal behavior are often overlooked. Furthermore, the proposed method introduces a dictionary-based data transformation strategy that restructures the user-item matrix to improve training efficiency and memory optimization [21]. This framework is validated using the H&M Personalized Fashion Recommendation Dataset, a real-world benchmark that represents the scale, scarcity, and complexity of modern fashion retail data. Through this empirical analysis, this study aims to contribute, both theoretically and practically, to the development of a high-performance and scalable recommendation system tailored to the demands of the digital fashion retail environment.

2. RESEARCH METHODS

In this study, the H&M Personalized Fashion data from Kaggle Data Science was analyzed using the Python programming language. The collaborative filtering method used in this study is expected to provide good recommendations based on user preferences.

2.1. Data collection

The data collection used was obtained from Kaggle from the well-known fashion company H&M with the title H&M Personalized Fashion Recommendation data, shown in Table 1. The loaded data is divided into 3 tables, namely, user, article, and transaction tables. The user data has seven attributes, with a total of 1,371,980 users. Article data has 25 attributes, where each type of item is divided by fabric type, type, and color with a total of 105,542 data items. Transaction data has five attributes with a total of 31,788,324 transactions, with a data time interval starting from September 20, 2018 to September 20, 2020 [22].

Table 1. H&M Fashion Recommendation Sample Data

t_dat	customer_id	article_id	price	sales_channel_id
2018-09-20	000058a12d5b43e67d225668fa1f8d618c13dc232df0cad8ffe7ad4a1091e318	0663713001	0,050830508	2
2018-09-20	000058a12d5b43e67d225668fa1f8d618c13dc232df0cad8ffe7ad4a1091e318	0541518023	0,030491525	2
2018-09-20	00007d2de826758b65a93dd24ce629ed66842531df6699338c5570910a014cc2	0505221004	0,015237288	2
2018-09-20	00007d2de826758b65a93dd24ce629ed66842531df6699338c5570910a014cc2	0685687003	0,016932203	2
2018-09-20	00007d2de826758b65a93dd24ce629ed66842531df6699338c5570910a014cc2	0685687004	0,016932203	2
2018-09-20	00007d2de826758b65a93dd24ce629ed66842531df6699338c5570910a014cc2	0685687001	0,016932203	2
2018-09-20	00007d2de826758b65a93dd24ce629ed66842531df6699338c5570910a014cc2	0505221001	0,020322034	2
2018-09-20	00083cda041544b2fbb0e0d2905ad17da7cf1007526fb4c73235dccbbc132280	0688873012	0,030491525	1
2018-09-20	00083cda041544b2fbb0e0d2905ad17da7cf1007526fb4c73235dccbbc132280	0501323011	0,053372881	1
2018-09-20	00083cda041544b2fbb0e0d2905ad17da7cf1007526fb4c73235dccbbc132280	0598859003	0,045745763	2
⋮	⋮	⋮	⋮	⋮

2.2. Pre-Processing

Pre-processing is a process in recommendation systems aimed at improving the quality and usability of raw data obtained from sources. This process is carried out with the aim of aligning and refining data so that it can be used more effectively and efficiently by the recommendation method. At this stage, the raw data is processed and modified, resulting in a more structured dataset ready for use as input by the recommendation algorithm. Pre-processing in recommendation systems uses less memory, speeds up the training process, and can produce more accurate and relevant recommendations for users. The pre-processing steps for user, article, and transaction data before use are as follows.

a) Perform clustering to analyze customer data with a focus on the age variable;
b) Segmenting data based on established clusters, more detailed analysis of age-related patterns and trends in user demographics;
c) Perform unique data checks;

d) Calculate by iteration of each unique data;
e) Add enumerate data to the transaction table;
f) Change the data type on t_dat (date time) and article id (string);
g) Check the date limit at the oldest and newest time;
h) Transforming dataFrame into dictionary, to lighten memory and speed up the training process.

2.3. Recommendation System

*Recommendation System*have a vital role in enhancing user experience across various online platforms, and these systems can be broadly categorized into non-personalized and personalized systems.[23]. Non-personalized recommendation systems adopt a universal approach by providing the same recommendations to all users, regardless of their preferences or behavior.[24]These systems often rely on general trends, overall popularity, or straightforward criteria to generate recommendations. Non-personalized systems are suitable for scenarios that require a broad, general approach, such as offering popular products to a diverse user base. These systems operate independently of user-specific data, making them easier to implement. Personalized recommendation systems, on the other hand, take a more individualized approach, tailoring recommendations to each user based on their unique preferences, behavior, and interaction history.[25]These systems dynamically adjust recommendations as users interact with the platform, learning from their feedback and evolving to reflect changing preferences. Personalized systems rely on a wealth of user data, including purchase history, browsing behavior, ratings, and explicit preferences. The adaptability and precision of personalized systems make them particularly effective in contexts that prioritize differentiated and tailored user experiences, such as e-commerce platforms, streaming services, or social media.

2.4. Collaborative Filtering

Collaborative filtering is an algorithm used in recommendation systems to provide personalized suggestions to users based on the preferences and behavior of similar individuals.[26]. The basic concept behind collaborative filtering is the assumption that users agree or align with their preferences, as illustrated in Figure 2. There are two main types of collaborative filtering: user-based collaborative filtering (UBCF) and item-based collaborative filtering (IBCF). In user-based collaborative filtering, recommendations are made by identifying users with similar tastes and preferences. On the other hand, item-based collaborative filtering recommends items based on their similarity to items the user has interacted with.[27].

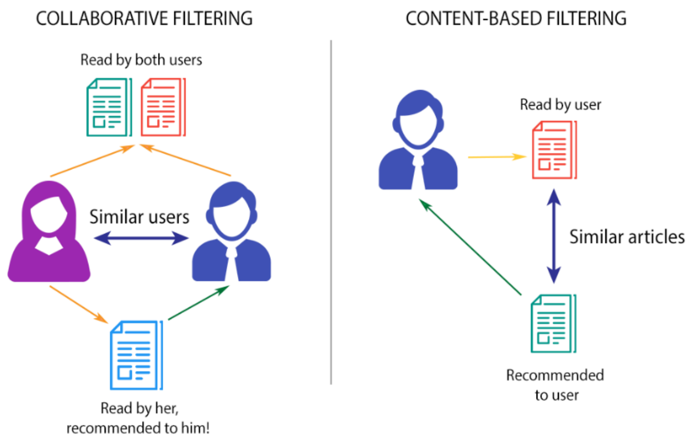

Figure 2. Illustration of Collaborative Filtering [28]

*Collaborative filtering*relies on a user-item interaction matrix, which captures the interaction history and ratings users give to items. While effective in providing personalized recommendations, it also faces challenges such as scalability issues with increasing numbers of users and items, sparse data issues, and issues when handling new or unrated items.[29]Collaborative filtering uses system interactions and data collected from other users to generate product feature recommendations based on user preferences. For example, if a user is searching for a watch, the system should suggest the watches the user likes most.[6]. These methods are divided into memory-based and model-based methods, where memory-based methods exploit the preferences of a group of like-minded users to generate recommendations that are likely to be of interest to target users. Since similarity calculations (between users or items) are an important component in memory-based methods, many similarity measures have been designed.[10]. Due to its high accuracy, it plays an important role based on a user item rating matrix created by statistically collecting user historical rating logs, with each entry representing the user's actual rating value for the corresponding web service.[30]. The collaborative filtering recommendation algorithm for making recommendations has several method steps used, as follows.

a) For user , it will search for similar users;(u)
b) For example, the 10 users most similar to and with their nearest neighbors;$u\ u'$
c) Search for the most popular items on neighboring users similar to u and items(i);
d) Then it will check for items that are popular in user/customer groups that are similar to , anduuWhichno transaction has been made, items that have not been purchased will be recommended to you.u.

2.5. Similarity Collaborative Filtering

The calculation of the similarity value of user-based collaborative filtering is explained in Equations (1)-(3) as follows.

$$U = \{u_1, u_2, u_3, \dots \dots . u_n\} \tag{1}$$

$$V = \{v_1, v_2, v_3, \dots \dots . v_m\} \tag{2}$$

$$C = \{c_1, c_2, c_3, \dots \dots . c_k\} \tag{3}$$

Based on user, based on item, and based on contextual attributes. Each row represents a user, and the column representing the item is the rating value based on the user's context.nmk[13]. The calculation of the product similarity value for each user is compared using collaborative filtering which has a similarity value between 0 and 1. Cosine similarity is a measure used in collaborative filtering to determine the similarity between two vectors to find similar users or items and how often in user-item interactions. The higher the cosine similarity value, the more similar the users or items are considered.[31]. This similarity measure is useful when dealing with sparse data, where users only rate or interact with a subset of items. In collaborative filtering, cosine similarity is applied to measure the similarity between users or items based on their preferences or behavior. The cosine similarity equation formula to determine the degree of similarity of both user 1 (A) and user 2 (B) between two vectors A and B, based on Equation (4), is as follows.

$$cos(A, B) = \frac{A * B}{||A|| * ||B||} \tag{4}$$

$$= \frac{\sum_{i=1}^{k} Ai\ Bi}{\sqrt{\sum_{i=1}^{k} Ai^2} \sqrt{\sum_{i=1}^{k} Bi}^2}$$

In this approach, the collaborative filtering recommendation algorithm for calculating similarity has several algorithms from the methods used, as follows.
*Input :*User rating data; User context data; User needs recommendations;(ua)
*Output:*List of items recommended to users;Nua
Start

a) Step 1: Create a similarity matrix based on users;
b) Step 2: Create a context similarity matrix based on users, calculate the context similarity value between two users;
c) Step 3: Create an integrated similarity matrix;
d) Step 4: Build a CF model based on the integrated similarity matrix;
e) Step 5: Identify a list of similar items for users who need a recommendation list;
f) Step 6: Recommend N user items based on the highest similarity value;

End

2.6. Evaluation of Mean Average Precision (MAP)

Evaluation to calculate the performance of the recommendation system used on personalized H&M fashion data, namely, mean average precision (MAP), where each class and value are calculated to obtain the average precision with an interval of 0-1.[32]. These evaluation steps involve several stages, such as determining the relevance of each item or recommendation. Next, precision is calculated at , where is the number of recommendations evaluated for each user. Precision measures the extent to which the recommendations provided match the user's preferences. After that, average precision (AP) is calculated for each user by averaging the precision values at each position where relevant items are found in the recommendation list.KKK[33]. AP provides an overview of the quality of recommendations at the individual user level. Finally, mean average precision (MAP) is calculated by averaging the AP values across all evaluated users. MAP provides an average measure of how well a recommendation system is able to provide relevant recommendations across different user contexts. The advantage of MAP lies in its ability to consider the order of recommendations, providing a more accurate picture of the system's overall performance. MAP indicates the extent to which a recommendation system is able to provide relevant and high-quality results.[34]. The MAP formula in equation (5) is as follows.

$$MAP\ @12 = \frac{1}{U}\sum_{u=1}^{U}\frac{1}{\min(m,12)} \tag{5}$$

$$\sum_{k=1}^{\min(n,12)} P(k) * rel(k)$$

Where the number 12 is the number of recommendations calculated based on 12 recommendation items for the user. is the number of users, is the precision at cut-off is the number of predictions per user, is the number of relevant ground truth values per user, and is an indicator function equal to 1 if the item at rank k is a relevant (correct) label, zero otherwise.$UP(k)k, nmrel(k)$

3. RESULTS AND DISCUSSION

In the results and discussion section, a fashion system recommendation process is carried out for each user based on transaction history. The data used is obtained from Kaggle from the H&M fashion company, divided into users, articles, and transactions. With a data time interval starting from September 20, 2018 to September 20, 2020. The recommended results can predict up to 12 article_id labels, which are predictions of items that users will purchase in the next seven days. The experiment was carried out in three stages, as follows; The first experiment is described in Table 2., using a time interval of the last two weeks of transactions used for train data in predicting fashion recommendations to users for the next seven days, the results of the experiment in predicting items per user. The second experiment is described in Table 3., carried out by adding a time interval in the last one month of transactions used for train data and also adding up the number of similar users, namely the total number of users to be determined to have a high similarity value and the results of the experiment. Then, for the final experiment, described in Table 4., with a time interval of two years, all transaction data is used for training, using a dictionary variable type to improve performance and shorten the training phase of the recommendation results. The recommendation result table (Tables 2., 3., and 4.) shows two main columns; customer_id and prediction, where each entry in the prediction column provides a list of recommended item ids for the user id that matches a particular customer_id. The presence of the same prediction data on the same customer_id also indicates that the recommendation system provides a series of similar or even the same products for several users. This phenomenon can be caused by similar preferences, shopping behavior between users and similar profiles or close preferences, thus receiving similar recommendations.

Table 2. Recommendation Results with a Two-Week Time Interval from the Last Transaction

customer_id	prediction
00000dbacae5abe5e23885899a1fa44253a17956c6d1c3d25f88aa139fdfc657	0751471001 0915529003 0918292001 0915526001 0751471043 0706016001 0898694001 0863595006 0448509014 0896152002 0714790020 0918522001
0000423b00ade91418cceaf3b26c6af3dd342b51fd051eec9c12fb36984420fa	0751471001 0915529003 0918292001 0915526001 0751471043 0706016001 0898694001 0863595006 0448509014 0896152002 0714790020 0918522001
000058a12d5b43e67d225668fa1f8d618c13dc232df0cad8ffe7ad4a1091e318	0751471001 0915529003 0918292001 0915526001 0751471043 0706016001 0898694001 0863595006 0448509014 0896152002 0714790020 0918522001
00005ca1c9ed5f5146b52ac8639a40ca9d57aeff4d1bd2c5feb1ca5dff07c43e	0751471001 0915529003 0918292001 0915526001 0751471043 0706016001 0898694001 0863595006 0448509014 0896152002 0714790020 0918522001
00006413d8573cd20ed7128e53b7b13819fe5cfc2d801fe7fc0f26dd8d65a85a	0751471001 0915529003 0918292001 0915526001 0751471043 0706016001 0898694001 0863595006 0448509014 0896152002 0714790020 0918522001

Table 3. Recommendation Results with One Month Time Interval from the Last Transaction

customer_id	prediction
00008469a21b50b3d147c97135e25b 4201a8c58997f78782a0cc706645e14493	0751471001 0915529003 0918292001 0915526001 0751471043 0706016001 0898694001 0863595006 0448509014 0896152002 0714790020 0918522001
0000945f66de1a11d9447609b8b41b1bc 987ba185a5496ae8831e8493afa24ff	0751471001 0915529003 0918292001 0915526001 0751471043 0706016001 0898694001 0863595006 0448509014 0896152002 0714790020 0918522001
000097d91384a0c14893c09ed047a963c4fc6 a5c021044eec603b323e8c82d1d	0751471001 0915529003 0918292001 0915526001 0751471043 0706016001 0898694001 0863595006 0448509014 0896152002 0714790020 0918522001
00009c2aeae8761f738e4f937d9be6b49861 a66339c2b1c3b1cc6e322729a370	0751471001 0915529003 0918292001 0915526001 0751471043 0706016001 0898694001 0863595006 0448509014 0896152002 0714790020 0918522001
00009d946eec3ea54add5ba56d5210e a898def4b46c68570cf0096d962cacc75	0751471001 0915529003 0918292001 0915526001 0751471043 0706016001 0898694001 0863595006 0448509014 0896152002 0714790020 0918522001

Table 3., presents the results of the first experiment, which utilized transaction data from the past two weeks to generate personalized fashion recommendations for H&M users. The model was tasked with predicting users' purchases over the next seven days, resulting in 12 recommended items per user. The model achieved a Mean Average Precision (MAP) score of 0.00943, indicating a moderate level of recommendation accuracy in a short-term temporal context. In contrast, Table 2., reports the results of the second experiment, in which the training dataset was expanded to include transactions from the previous month, increasing the temporal window from two weeks to one month. As in the first experiment, the model generated 12 personalized item recommendations for each user to predict purchasing behavior over the next seven days. However, the MAP score decreased to 0.00554, indicating that the inclusion of a broader, but potentially more heterogeneous, dataset did not improve performance. These results imply that while expanding the temporal scope introduces additional data, it may also introduce noise or dilute the strength of recent behavioral signals, thereby reducing the accuracy of short-term recommendations.

Table 4. Recommendation Results with an Overall Time Interval of Two Years from the Last Transaction

customer_id	prediction
00000dbacae5abe5e23885899a1fa44253a17956c6d1c 3d25f88aa139fdfc657	0568601043 0568601006 0859416011 0656719005 0745232001 0795440001 0785710001 0607642008 0568601006 0568597006 0706016001 0562245001
000058a12d5b43e67d225668fa1f8d618c13dc232df0c ad8ffe7ad4a1091e318	0794321007 0852643003 0852643001 0858883002 0727808007 0351484002 0723529001 0851400006 0750424014 0794321011 0852643004 0685813001
00005ca1c9ed5f5146b52ac8639a40ca9d57aeff4d1bd2 c5feb1ca5dff07c43e	0448509014 0573085028 0706016001 0751471001 0673677002 0715624001 0706016003 0158340001 0706016002 0579541001 0372860002 0372860001
00006413d8573cd20ed7128e53b7b13819fe5cfc2d801 fe7fc0f26dd8d65a85a	0896152002 0791587015 0730683050 0927530004 0818320001 0589440005 0896152001 0791587001 0706016001 0927530006 0732206001 0589440002
000064249685c11552da43ef22a5030f35a147f723d5b 02ddd9fd22452b1f5a6	0448509014 0573085028 0706016001 0751471001 0673677002 0715624001 0706016003 0158340001 0706016002 0579541001 0372860002 0372860001

Table 4., presents the results of the third experiment, which leveraged complete transaction history over a two-year period to improve recommendation accuracy while minimizing training time through data structure optimization. This approach was designed to predict user purchases over the next seven days by generating 12 personalized fashion item recommendations for each H&M user. The model achieved a MAP score of 0.02284, the highest among all experimental configurations. Compared to previous experiments with shorter time intervals, these results demonstrate the effectiveness of incorporating long-term user interaction data in capturing stable behavioral patterns and improving predictive performance. Furthermore, despite the significantly larger training dataset, the computation time of 2741.7 seconds remained within an acceptable range, outperforming the one-

month model in terms of accuracy and efficiency. This performance improvement is due to the use of an optimized dictionary-based data structure, which reduces computational overhead during training. These findings demonstrate that collaborative filtering models benefit from richer temporal data when properly optimized, resulting in higher recommendation accuracy without compromising system scalability.

Table 5. Comparison Results of MAP Recommendation System

Algorithms	MAP	Time (s)
Collaborative Filtering (2 weeks)	0.00913	**1,609.2**
Collaborative Filtering (1 month)	0.00554	3,118.9
Collaborative Filtering (2 years)	**0.02284**	2,741.7

Table 5., presents a comparative evaluation of collaborative filtering algorithms across different temporal training intervals, with the goal of predicting short-term purchasing behavior in the H&M fashion dataset. The results indicate that the choice of time interval plays a crucial role in determining the performance of the recommendation system, as reflected by the Mean Average Precision (MAP) metric. Among the three configurations, the model trained using a two-year interval achieved the highest MAP score of 0.02284, outperforming the two-week (0.00913) and one-month (0.00554) interval models. This finding suggests that a longer historical data window allows the model to better capture stable and recurring user-item interaction patterns, thus improving its ability to provide relevant recommendations. Notably, the one-month interval, despite offering broader data coverage than the two-week interval, yielded the lowest MAP value, suggesting that intermediate time frames may be less concise for capturing short-term preferences and less broad for identifying long-term behavioral trends. In terms of computational efficiency, the two-week interval model completed training in 1609.2 seconds, while the one-month and two-year models required 3118.9 and 2741.7 seconds, respectively. Although the two-year interval involves a larger dataset, the model benefits from an optimized data structure using dictionary-based transformations, which improves access speed and reduces memory overhead, thus maintaining a reasonable training time. This demonstrates that increasing historical data volume does not necessarily lead to a proportional increase in computational costs, especially when the data pipeline is efficiently engineered. Furthermore, the superior MAP score achieved by the two-year model in forecasting the next seven-day purchasing period emphasizes the predictive power of long-term transactional data in fashion e-commerce. Despite the temporal distance between the training and prediction periods, historical purchasing behavior still provides significant insights into future preferences. These findings highlight the importance of temporal granularity in collaborative filtering systems and confirm that expanding the temporal scope of training data when supported by computational optimization can lead to improved recommendation accuracy and scalable model performance.

4. CONCLUSION

This study evaluates the performance of a collaborative filtering-based recommendation system on the H&M personalized fashion dataset using different temporal training intervals. The results show that the model trained with two years of transaction history achieved the highest accuracy, with a Mean Average Precision (MAP) score of 0.02284 and a computation time of 2741.7 seconds. In contrast, a two-week interval yielded a lower MAP score of 0.00915, despite having the fastest computation time of 1609.2 seconds. Specifically, expanding the training interval to one month resulted in a decrease in the MAP score of 0.00554 and the longest computation time of 3118.9 seconds. These findings highlight the importance of choosing the right temporal granularity in the training data, as longer historical intervals provide richer behavioral patterns that improve recommendation accuracy. Furthermore, the optimized data transformation technique used in this study using a dictionary-based structure enables efficient model training even with large-scale data. Overall, the proposed approach demonstrates that collaborative filtering, when combined with an expanded user interaction history and efficient data processing, can generate accurate and scalable personalized mode recommendations according to user preferences.

ACKNOWLEDGEMENTS

The author would like to thank all parties who supported this research. In particular, he would like to express his appreciation to Surabaya State University, specifically the Faculty of Vocational Studies and the Diploma 4 Informatics Management Study Program, for the facilities provided during the system design and testing process. He would also like to thank his fellow lecturers, who assisted in the preparation of this article.

REFERENCES

[1] LJ Padilla Reyes, NB Oviedo, EC Camacho, and JM Calderon, "Adaptable recommendation system for outfit selection with deep learning approach,"*IFAC-PapersOnLine*, vol. 54, no. 13, pp. 605–610, 2021, doi: 10.1016/j.ifacol.2021.10.516.

[2] A. Alkhalifah, F. Alorini, and R. Alturki, "Enhancement of E-commerce Service by Designing Last Mile Delivery Platform,"*Comput. Syst. Sci. Eng.*, vol. 42, no. 1, 2022, doi: 10.32604/csse.2022.021326.

[3] Y. Deldjoo*et al.*, "A Review of Modern Fashion Recommender Systems," ArXiv J., vol. 37, no. 4, 2022, [Online]. Available: http://arxiv.org/abs/2202.02757

[4] W. Zhou*et al.*, "Fashion recommendations through cross-media information retrieval," J. Vis. Commun. Image Represent., vol. 61, pp. 112–120, 2019, doi: 10.1016/j.jvcir.2019.03.003.

[5] S. Wang and J. Qiu, "A deep neural network model for fashion collocation recommendation using side informationin e-commerce," Appl. Soft Comput., vol. 110, p. 107753, 2021, doi: 10.1016/j.asoc.2021.107753.

[6] J. Sharma, K. Sharma, K. Garg, and AK Sharma, "Product recommendation system a comprehensive review,"*IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1022, no. 1, 2021, doi: 10.1088/1757-899X/1022/1/012021.

[7] N. Nassar, A. Jafar, and Y. Rahhal, "A novel deep multi-criteria collaborative filtering model for recommendation systems,"*Knowledge-Based System*, vol. 187, p. 104811, 2020, doi: 10.1016/j.knosys.2019.06.019.

[8] H. Hwangbo, YS Kim, and KJ Cha, "Recommendation system development for fashion retail e-commerce,"*Electron. Commer. Res. Appl.*, vol. 28, pp. 94–101, 2018, doi: 10.1016/j.elerap.2018.01.012.

[9] AP Faria, J. Cunha, and B. Providência, "Fashion communication in the digital age: Findings from interviews with industry professionals and design recommendations,"*Procedia CIRP*, vol. 84, pp. 930–935, 2019, doi: 10.1016/j.procir.2019.04.204.

[10] C. Feng, J. Liang, P. Song, and Z. Wang, "A fusion collaborative filtering method for sparse data in recommender systems,"*Inf. Sci. (Ny).*, vol. 521, pp. 365–379, 2020, doi: 10.1016/j.ins.2020.02.052.

[11] D. Mican, D.-A. Sitar-Tăut, and O.-I. Moisescu, "Perceived usefulness: A silver bullet to assure user data availability for online recommendation systems,"*Decis. Support Syst.*, vol. 139, p. 113420, 2020.

[12] KASA VP and ZKA Baizal, "Improved Collaborative Filtering Recommender System Based on Missing Values Imputation on E-Commerce,"*Build. Informatics, Technol. Sci.*, vol. 3, no. 4, pp. 453–459, 2022, doi: 10.47065/bits.v3i4.1214.

[13] HX Huynh*et al.*, "Context-Similarity Collaborative Filtering Recommendation," IEEE Access, vol. 8, pp. 33342–33351, 2020, doi: 10.1109/ACCESS.2020.2973755.

[14] F. Wang*et al.*, "Robust Collaborative Filtering RecommendationWith User-Item-Trust Records," IEEE Trans. Comput. Soc. Syst., pp. 1–11, 2021.

[15] MK Syiam, AT Wibowo, and EB Setiawan, "Fashion Recommendation System using Collaborative Filtering,"*Build. Informatics, Technol. Sci.*, vol. 5, no. 2, 2023, doi: 10.47065/bits.v5i2.3690.

[16] A. Fareed, S. Hassan, SB Belhaouari, and Z. Halim, "A collaborative filtering recommendation framework using social networks,"*Mach. Learn. with Appl.*, vol. 14, p. 100495, 2023.

[17] Z. Yang, Z. Su, Y. Yang, and G. Lin, "From recommendation to generation: A novel fashion clothing advising framework,"*Proc. - 7th Int. Conf. Digits. Home, ICDH 2018*, pp. 180–186, 2019, doi: 10.1109/ICDH.2018.00040.

[18] Z. Cui*et al.*, "Personalized Recommendation System Based on Collaborative Filtering for IoT Scenarios," IEEE Trans. Serv. Comput., vol. 13, no. 4, pp. 685–695, 2020, doi: 10.1109/TSC.2020.2964552.

[19] A. Iftikhar, M.A. Ghazanfar, M. Ayub, Z. Mehmood, and M. Maqsood, "An Improved Product Recommendation Method for Collaborative Filtering,"*IEEE Access*, vol. 8, pp. 123841–123857, 2020, doi: 10.1109/ACCESS.2020.3005953.

[20] ASNS Ningrum, "Content-Based and Collaborative Filtering for Tourism Destination Recommendations in the Yogyakarta Region,"*Telematics*, vol. 16, no. 1, p. 44, 2019, doi: 10.31315/telematics.v16i1.3023.

[21] JK Arthur, C. Zhou, EA Mantey, J. Osei-Kwakye, and Y. Chen, "A discriminative-based geometric deep learning model for cross domain recommender systems,"*Appl. Sci.*, vol. 12, no. 10, p. 5202, 2022.

[22] H. Group, "H&M Personalized Fashion Recommendations - Provide product recommendations based on previous purchases," kaggle.com. [On line]. Available: https://www.kaggle.com/competitions/h-and-m-personalized-fashion-recommendations

[23] D. Mican and D.-A. Sitar-Taut, "The effect of perceived usefulness of recommender systems and information sources on purchase intention,"*Kybernetes*, 2023.

[24] AN Nikolakopoulos, X. Ning, C. Desrosiers, and G. Karypis, "Trust your neighbors: a comprehensive survey of neighborhood-based methods for recommender systems,"*Recomm. Syst. Handb.*, pp. 39–89, 2021.

[25] W. Fan*et al.*, "Recommender systems in the era of large language models (llms)," arXiv Prepr. arXiv2307.02046, 2023.

[26] S. Sharma, V. Rana, and M. Malhotra, "Automatic recommendation system based on hybrid filtering algorithm,"*Educ. Inf. Technol.*, pp. 1–16, 2022.

[27] L. Wu, X. He, X. Wang, K. Zhang, and M. Wang, "A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation,"*IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4425–4445, 2022.

[28] L. Ngoupeyou Tondji, "Web Recommender System for Job Seeking and Recruiting,"*IEEE Internet Comput.*, vol. 60,

no. 12, pp. 169–182, 2018, doi: 10.13140/RG.2.2.26177.61286.
[29] F. Fkih, "Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison,"*J. King Saud Univ. Inf. Sci.*, vol. 34, no. 9, pp. 7645–7669, 2022.
[30] D. Vijayalakshmi and J. Kanchana, "Evaluating the effectiveness of the collaborative learning in fashion studies,"*Procedia Comput. Sci.*, vol. 172, no. 2019, pp. 991–1000, 2020, doi: 10.1016/j.procs.2020.05.146.
[31] MD Pratama, R. Sarno, and R. Abdullah, "User Sentiment Analysis Regarding Hotel Reviews by Aspect Based Using Latent Dirichlet Allocation, Semantic Similarity, and Support Vector Machine Method,"*Int. J. Intell. Eng. Syst.*, vol. 15, no. 3, pp. 514–524, 2022, doi: 10.22266/ijies2022.0630.43.
[32] R. Sun, R. Chang, T. Yu, D. Wang, and L. Jiang, “U-Net Modeling-Based Imaging MAP Score for Tl Stage Nephrectomy: An Exploratory Study,”*J. Healthc. Eng.*, vol. 2022, 2022.
[33] M. Bałchanowski and U. Boryczka, “Aggregation of Rankings Using Metaheuristics in Recommendation Systems,”*Electronics*, vol. 11, no. 3, p. 369, 2022.
[34] V. La Gatta, V. Moscato, M. Pennone, M. Postiglione, and G. Sperlí, “Music recommendation via hypergraph embedding,”*IEEE Trans. neural networks Learn. Syst.*, 2022.

AUTHOR BIOGRAPHY

	**Moch Deny Pratama** is a lecturer in the Department of Informatics Management at the State University of Surabaya, Indonesia. He earned his Bachelor of Applied Engineering (S.Tr.T) in Informatics Engineering from the State Polytechnic of Malang (POLINEMA) in 2021. He later obtained his Master of Computer Science (M.Kom) in Informatics Engineering from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, in 2023. His primary research interests include Artificial Intelligence, Machine Learning, Algorithm Design, and Data Science. He can be contacted via email at: mochpratama@unesa.ac.id.
	**Dimas Novian Aditia Syahputra**is a lecturer at the Department of Informatics Management, Surabaya State University, Indonesia. He earned his Bachelor of Applied Electrical Engineering (S.Tr.T) from the Surabaya State Electronics Polytechnic in the field of Electrical Engineering, Surabaya in 2019. He also earned his Master of Applied Computer (M.Tr.Kom) from the Surabaya State Electronics Polytechnic (PENS) in the field of Electrical Engineering, Surabaya in 2022. He mainly researches the Internet of Things and AR/VR. He can be contacted via email: dimassyahputra@unesa.ac.id.
	**M Adamu Islam Mashuri** is a lecturer at the Department of Informatics Management, Surabaya State University, Indonesia. He earned his Bachelor of Applied Engineering (S.Tr.T) from the Surabaya State Electronics Polytechnic (PENS) in Telecommunication Engineering, Surabaya in 2021. He also earned his Master of Applied Computer (M.Tr.Kom) from the Surabaya State Electronics Polytechnic (PENS) in Informatics and Computer Engineering, Surabaya in 2023. He mainly researches the Internet of Things and Artificial Intelligence. He can be contacted via email: mmashuri@unesa.ac.id.
	**Binti Kholifah** is a lecturer in the Department of Informatics Management, Surabaya State University, Indonesia. She earned her Bachelor of Computer Science (S.Kom) from Maulana Malik Ibrahim State Islamic University in Informatics Engineering, Surabaya in 2019. She also earned her Master of Applied Computer Science (M.Tr.Kom) from Surabaya State Electronics Polytechnic (PENS) in Informatics and Computer Engineering, Surabaya in 2021. She primarily research Game Programming and Mobile Programming. She can be contacted via email: bintikholifah@unesa.ac.id.
	**Rifqi Abdillah** is a lecturer in the Department of Informatics Engineering, Surabaya State University, Indonesia. He earned his Bachelor of Applied Engineering (S.Tr.T) from the Electronic Engineering Polytechnic Institute of Surabaya (PENS) in 2021, majoring in Informatics Engineering. He subsequently earned his Master of Computer Science (M.Kom) from the Institut Teknologi Sepuluh Nopember (ITS), Surabaya in 2023. His primary research include Machine Learning, Data Mining, and Internet of Things. He can be contacted via email: rifqiabdillah@unesa.ac.id.
	**Adinda Putri Pratiwi** is a researcher in the Department of Informatics Engineering, Institut Teknologi Sepuluh Nopember (ITS), Indonesia. Her majoring in Informatics Engineering subsequently earned her Master of Computer Science (M.Kom) from the Institut Teknologi Sepuluh Nopember (ITS), Surabaya in 2023, focusing on Data Mining and Intelligent Systems. Her primary research interests include Machine Learning, Data Mining, and Internet of Things. He can be contacted via email: adindaputripratiwi97@gmail.com.