IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 November 2023, accepted 19 January 2024, date of publication 29 January 2024, date of current version 9 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3359274

SURVEY

Systematic Literature Review on Recommender System: Approach, Problem, Evaluation Techniques, Datasets

ILHAM SAIFUDIN[1,2] **AND TRIYANNA WIDIYANINGTYAS**[1], **(Member, IEEE)**
[1]Departement of Electrical Engineering and Informatics, Universitas Negeri Malang, Malang 65145, Indonesia
[2]Informatics Engineering, Engineering Faculty, Universitas Muhammadiyah Jember, Jember Regency 68121, Indonesia

Corresponding author: Triyanna Widiyaningtyas (triyannaw.ft@um.ac.id)

This work was supported in part by the Doctoral Study Program in Electrical Engineering and Informatics and the Research and Publication Center of the Universitas Negeri Malang (UM), and in part by Universitas Muhammadiyah Jember (Unmuh Jember).

ABSTRACT Recommender systems become essential with the presence of the internet and social media. The perceived benefits of the recommender system can make it easier for users to find suitable products and recommend other products, specifically with lots of information. Recommender systems continue to develop over time. It has led many researchers to continue to find the latest approach and evaluation techniques by comparing the performance of previously existing recommender systems. The main approaches that are often used in recommender systems are Content-Based Filtering (CBF), Collaborative Filtering (CF), and Hybrid Filtering (HBF). This time, we focus on conducting a Systematic Literature Review (SLR) of several research articles and analyzing methods for algorithms developed in building recommender systems. The SLR method consists of three stages: planning, implementation, and reporting. The research used as a comparison is between 2019 and 2023 using various existing data sets. There were 72 primary studies, of which 46 employed the Collaborative Filtering approach, 11 used Content-Based filtering, and 15 used Hybrid Filtering. The results of this SLR process show the advantages and disadvantages of each method and type of evaluation developed in building a recommender system. Apart from that, several challenges arise with various existing problems. However, the model-based collaborative filtering method is one method that can minimize the problems of cold start, data sparsity, and scalability.

INDEX TERMS Content-based filtering, collaborative filtering, hybrid filtering, systematic literature review, recommender system.

I. INTRODUCTION

In recent years, recommender systems have become an exciting topic to discuss. Recommender systems (RS) have many uses, including searching for product information, music, movie entertainment, information on suitable educational places, world tourism spots, suitable hotels, etc. [1]. The recommender system is embedded in a software system and can be accessed by mobile devices and computers worldwide. The recommender system makes it easy for users to find information on products, services, and entertainment that will be selected and followed according to preferences to meet various needs. Every year, the implementation of the recommender system continues to be updated to achieve optimal results in its performance.

A Recommender System is an algorithm intended to recommend suitable/relevant items to users in scientific studies. This article will discuss various recommender system paradigms with multiple approaches. The following approaches will be discussed: Content-Based Filtering (CBF), Collaborative Filtering (CF), and Hybrid Filtering (HBF). Apart from that, it will also discuss how it works, its theoretical basis, and its strengths and weaknesses [2].

Content-based filtering (CBF) is built based on the assumption that users like products/items with features that are available in the past and future [3]. The features entered include information from the user, such as gender, age, etc. Likewise, for items/products, we can quickly get information

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina.

regarding features, such as genre, when the product was made, and so on. This Content-Based approach method does not experience many cold starts because new users/items can be described based on characteristics so that relevant suggestions can be made [4]. The weakness of this method is that if a particular user or item has little information, then few recommendations will be given.

The Collaborative Filtering (CF) approach is a recommendation method based on recorded past interactions between users and items to produce new recommendations [5]. The interactions in question are stored in the user-item interaction matrix. This method has advantages and disadvantages. This method's advantage is that it does not require information about the user or item, so it can be used in many situations. Additionally, the more a user interacts with an item, the more recommendations there are [6]. In some instances, if a particular group of users and items, then new interactions recorded over time will bring new information and make the information more effective. Meanwhile, the weakness of this method is that if the user and item have little interaction, there will be few recommenders for other users [7].

A recommender system using Hybrid Filtering (HBF) is obtained from a combination of two or more other recommender techniques to try to overcome the shortcomings of each recommender system technique. Furthermore, the combination of developing a hybrid recommender system approach depends on the characteristics of the data used. There are seven categories of hybrid recommender systems consisting of weighted, switching, mixed, feature combination, feature augmentation, cascade, and meta-level, which have been introduced [8], [9], [10].

This paper discusses a comprehensive review of research results between 2019-2023 using various existing data sets. There are several characteristics related to the review in this paper, including 1) the problem that is the main thing that produces a particular method for completing the recommender system; 2) The relevance of the method used according to the discussion in this paper; 3) evaluation methods that are following existing development methods and produce new contributions to the development of the recommender system; 4) This paper also provides information on developments and challenges for researchers to solve problems in thefuture.

From the explanation above, several research questions (RQ) are obtained related to achieving the objectives of this SLR.

RQ1 : What approach is suitable for solving problems in recommender systems?

RQ2 : What problems arise in solving issues in recommender systems?

RQ3 : What evaluation techniques are suitable for solving problems in recommender systems?

RQ4 : What datasets are currently popular for research in recommender systems?

In preparing this paper, a structure is explained in detail, such as 1) the background of the recommender system, 2) the SLR methodology used, 3) the approach and evaluation methods in looking at the performance of recommender system methods, 4) reporting the results of questions from previous researchers in the form of discussions and challenges; and last; 5) conclusions and challenges provided by researchers.

II. REVIEW METHOD

This second part explains the Systematic Literature Review (SLR) methodology. The stages include identifying problems and evaluating findings from research results related to recommender systems. This SLR aims to identify gaps in related research and provide a comprehensive review of the research results from 2019 to 2023. Many things are of concern regarding the development of methods, such as Content-Based Filtering (CBF), Collaborative Filtering (CF), and Hybrid Filtering (HBF). This will make it more straightforward and understand the developments that can be made in the future.

A. SELECTION CRITERIA PAPERS

The initial stage is to determine the paper criteria from previous studies. The data collected is in the form of review questions and from publishers of scientific article journals and several conference journals from 2019 to 2023 in the general language (English). Below is an explanation of the selection criteria.

- Article containing a new recommender system approach method.
- Articles in international language (English)
- Journal has an indexed reputation with sources (SpringerLink et al.)

B. SEARCH METHOD USED

The search method is carried out in two stages: automatic and manual [4]. This was done to explore the main study. Automatic searches are carried out in online databases by adding specific keywords. For example, using the keyword: recommender system. We can also use keywords in direct approach methods, such as Content-Based Filtering, Collaborative Filtering, or Hybrid Filtering. The search was conducted using access to the State University of Malang (UM) campus facilities via the Online Library service. This access can fully view the journal on the ScienceDirect, Springer, Elsevier, Taylor & Francis, MDPI, ACM, and IEEExplore pages.

Meanwhile, the second method is manual, carried out on the Google Scholar search platform with the keyword Recommender System based on Content-Based Filtering Collaborative Filtering or Hybrid Filtering. Then, filtering was carried out manually and sequentially from 2019 to 2023, and several pieces of information related to the desired article appeared, such as IEEExplore and others. Both methods are applied and produce the desired data/information for articles and journals. All recorded information that has been obtained is then recorded in a Microsoft Excel file to organize and reduce duplicate information in the article study used in this paper.

C. SELECTION PROCESS PAPERS

In this activity, articles are selected to support SLR by following the flow in Figure 1 and Table 1 below.

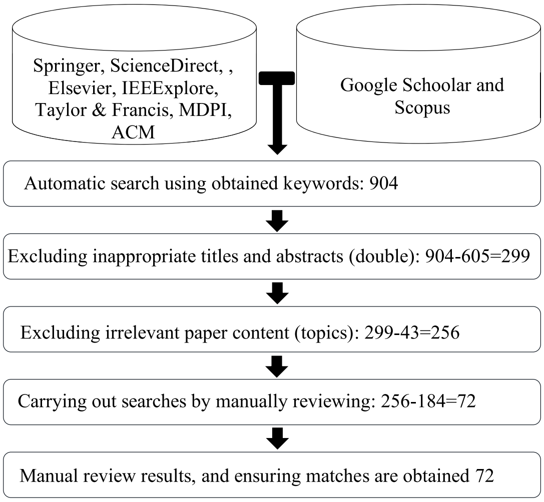

FIGURE 1. Search method.

TABLE 1. Article selection process.

No	Source	Automated Search	Jurnal Q1&Q2	Related Abstract
1	IEEExplore	150	45	7
2	MDPI	71	20	5
3	Taylor & Francis	54	15	5
4	ACM	101	52	15
5	Springer	256	99	31
6	ScienceDirect	197	8	2
7	Elsevier	75	17	7
	Total	904	256	72

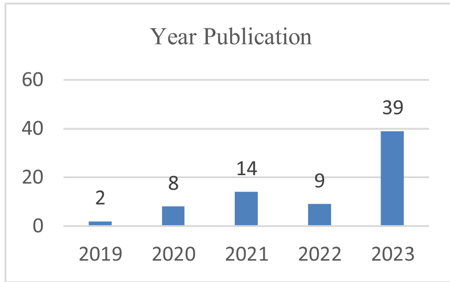

FIGURE 2. Distribution of references by year.

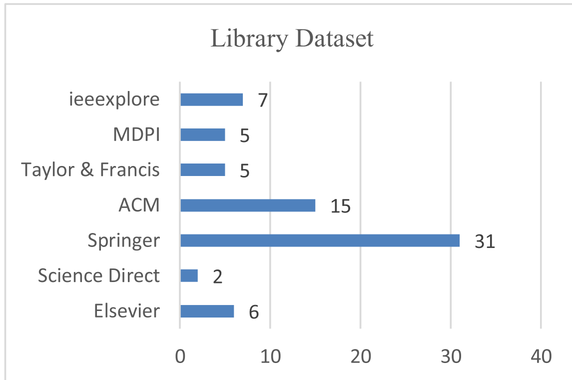

FIGURE 3. Graph of number of articles based on library data sets.

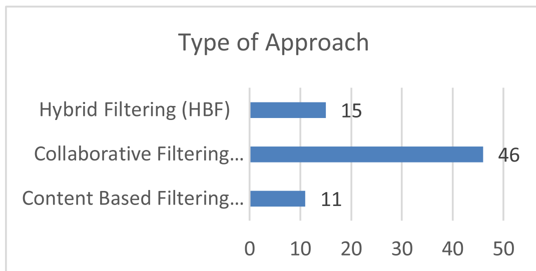

FIGURE 4. Type of approach used.

Next, Figure 2 obtained Reference Distribution based on the last five years. The range of article distribution starts from 2019 to 2023. The results obtained from the search and selection results were 72 articles. The most extensive distribution was in 2023, with 39 articles. Second place in 2021 was 14 research articles. The third place and so on were nine articles in 2022, 8 in 2020, and 2 in 2019. Thus, most of the SLR process uses the latest research as a reference and contains various methods used with increasingly complex problems.

In Figure 3, we can see the distribution of the Library Data set with the most searches on Springer with 31 articles. The second sequence is ACM (Association for Computing Machinery), with 15 articles. Then in third place, followed by IEEExplore with 7, Elsevier with 6, MDPI and Taylor & Francis with five articles, and Science Direct with 2 articles.

Figure 4 shows the results of the paper selection process and the results of the research carried out regarding the approach to the Recommender System. There were 72 primary studies where the Collaborative Filtering method was used in 46 studies, Content-Based Filtering was used in 11 studies, and the Hybrid Filtering method was used in 15 studies. There are four stages (can be seen in Figure 1 Search Method) that serve as a reference in the process of data retrieval, selection, and article results that follow the discussion in this paper. In the first stages, carry out searches using several article data sources (ScienceDirect et al.) and combine them with data sources from Google (Google Scholar) according to predetermined keywords. After the search was carried out, 904 articles were obtained. Next, we sorted the titles and abstracts irrelevant to the study topic by reading the articles, resulting in 605 articles that did not match, and some articles were double (the same). The results of this second step were 299 articles. The third process, excluding the content of articles that were not following the study on this topic, resulted in 43, so the filtering process resulted in

256 remaining articles following the topic. The final/fourth step was a manual search by reviewing the 125 available articles. As a result of the manual review, 182 articles were reviewed, resulting in 72 articles used and discussed in this paper.

III. RESEARCH QUESTION RESULTS

In the following subsections, we present the results of our systematic review of our research question.

A. APPROACH

RQ1 : What approach is suitable for solving problems in recommender systems?

There are several methods discussed in this SLR. This method is often used, and developments in this approach are carried out for various reasons. Figure 5 shows an overview of the recommender system classification and an explanation of the formulas.

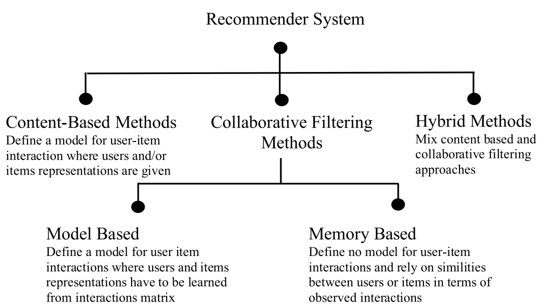

FIGURE 5. Recommender system classification.

In Figure 6, several results of the approach model are obtained, with 72 articles related to the Recommender System. The distribution consists of 5 types of approach models: 1) memory-based (Ranking Oriented), 2) memory-based (Rating Oriented), 3) model-based (Ranking Oriented), 4) Model Based (Rating Oriented), and 5) model-based. The highest score in the approach method often used is memory bases (Rating Oriented), with 28 studies. This was followed by a model-based approach (Rating Oriented) with 14 studies.

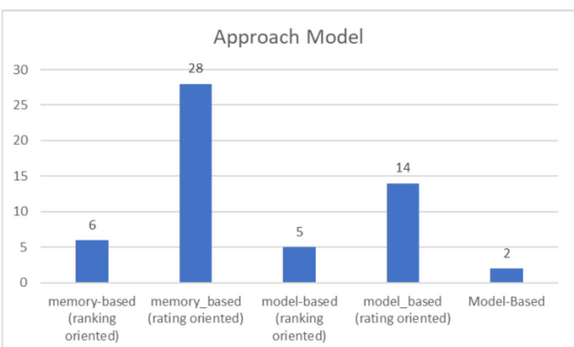

FIGURE 6. Approach model distribution in the recommender system.

Content Based Filtering (CBF) Approach: The remaining six are memory-based (ranking-oriented) approach models, five-based (ranking-oriented) models, and two studies based on models based on other data orientations. Thus, memory bases (Rating Oriented) are being widely researched to produce related performance and reduce errors with the large amount of data presented. Next, several methods often used in recent research will be explained. Each type is provided with illustrations of 3 studies from Content-Based Filtering (CBF), Collaborative Filtering (CF), and Hybrid Filtering (HBF).

The CBF technique relies on product descriptions and user profiles. In addition, product characteristics analysis is carried out to produce recommendations. Generating a user profile depends on two aspects: 1) the user's interaction history with the system and 2) the result of each user's preferences. The results of CBF research include various models of approaches and methods used, such as recommendations for movies, tourist attractions, and other sites [11], [12], [13]. This method uses a similarity metric to calculate the similarity of each user item in a vector space to produce the best recommendation. There are advantages obtained with the CBF method if seen from its use. One of the advantages is that CBF does not need to require other user information because this information does not change or influence the recommendation results. This makes CBF more flexible than other methods. However, this method also has disadvantages, requiring the necessary domain skills to produce optimal results. Below is some information related to CBF research that is relevant and up-to-date.

Below, we will illustrate a CBF formula for finding similar movies, which uses movie attributes to find the similarities. For example, a feature set. The tagline of a movie [11] is added with an overview attribute. Combining these two attributes requires several stages before knowing the similarities.

Stage 1: Clean data properly. It will also allow us to reduce sentences, paragraphs, and documents to a single set of words.

Stage 2: Carry out a stemming process to address words inflected to their base form, root, or word origin.

Stage 3: Remove stop words from the set of words. An English word dictionary has been used to remove stop words.

Stage 4: After creating the clean dataset and filters, the next is to implement the core functionality. Term frequency is calculated by the number of times term t is repeated in document d.

$$TF\ (t, d) = count\ of\ tind/number\ of\ wordsind \quad (1)$$

Stage 5: Inverse document frequency (IDF) reduces the weight of frequently used terms. In contrast, by calculating the IDF, the logs the number of documents N in the corpus D divided by the number of documents df_t including term t.

$$IDF\ (t) = log \frac{N}{df_t} \quad (2)$$

Stage 6: Finally, for the term t, the weight in a particular document d is determined as a result of the two previous calculations:

$$TF\!-\!IDF(t,d) = TF(t,d).IDF(t) \quad (3)$$

Stage 7: Calculate the similarity between two movie descriptions using cosine similarity. Table 2 show a study related to Content-Based filtering.

TABLE 2. Related work on content-based filtering.

Study	Year	Model	Methods
[11]	2023	model-based (rating oriented)	a CNN deep learning (DL) model to build a multiclass
[13]	2023	memory-based (rating oriented)	Bipartite Networks
[14]	2021	memory-based (rating oriented)	algorithm (p-CAR) in IoT

Based on Table 2, the study [11] contains flaws, such as the requirement for an expert system that can reasonably accurately forecast a film's likelihood of success before to production for both researchers and filmmakers. The most study has been done to forecast the post-production movie industry's appeal. To help makers forecast upcoming movies and make necessary changes. Early in the filmmaking process, forecasts must be made, and precise observations regarding films that are soon to be released must be made. This study suggests a Content-Based (CB) movie recommender system (RS) that makes use of starting features such keywords, movie description, cast, genre, and director. We develop a set of novel characteristics and suggest a deep learning (DL) CNN model to construct a multiclass movie popularity prediction system using RS output, movie ratings, and similar movie voting data. Additionally, a system to forecast the level of popularity of upcoming films among various audience segments was suggested by the research. The audience has been split up into four age groups: juniors, teenagers, middle-aged, and seniors. The following is the formula used in this research.

Calculating the similarities between two movies m_i & m_j regarding features F_k are

$$disF_{k_{i,j}} = similarity(Fk_i, Fk_j) \quad (4)$$

$$dist_{i,j} = \cup_k^m \{dist_{Fk_{i,j}}\} \quad (5)$$

Next, the overall similarity measure between the two movies is calculated using a similarity measure such as cosine similarity:

$$c\_dist_{ij} = cosine\_sim(dist_{ii}, dist_{ij}) \quad (6)$$

$$c\_dist_i = \{c\_dist_{ij}\}_{j=1}^n \quad (7)$$

The next step recommended movies from $m_i.V_r^i$ represents the total number of votes with r ratings for all recommended movies m_i.

$$V_r^i = \Sigma_{k=1}^N v_{r,k}^i \quad (8)$$

$$V^i = \{V_r^i | r = 1, 2, 3, \ldots, 10\} \quad (9)$$

In (9), the letter V^i represent the voting details of the movie m_i. In (10) also the R_j^i rating of each recommended movie is considered as:

$$R^i = \{R_j^i | j = 1, 2, 3, \ldots, N\};$$
$$V = \{V^i\}_{i=1}^n \text{ and } R = \{R^i\}_{i=1}^n,$$
$$\text{Creates Input Dataset } X = V \cup R \quad (10)$$

The target will be determined based on each data set consisting of rating and voting information from all recommended movies m_i.

$$Gr_j^i = \{R_{Gr_{j,k}}^i, Vr_{Gr_{j,k}}^i | k = 1, \ldots |, N\} \quad (11)$$

Fuzzy c mean is used to determine the cluster centroid of each group.

$$(C\_R_{Gr_j}^i, C\_V_{Gr_j}^i)_{j=1}^4 = FCM(\{C\_R_{Gr_{j,k}}^i, C\_V_{Gr_{j,k}}^i | \times k = 1, \ldots, N\&j = 1, \ldots 4\}) \quad (12)$$

$$C\_centroid_{Gr_j}^i = (C\_R_{Gr_j}^i, C\_V_{Gr_j}^i) \quad (13)$$

$$G_{centroid} = (G_R, G_{Vr}) \quad (14)$$

Next, consolidating Global parameters, we set to measure the similarity between Global centroids and movie centroids for a group. If $C\_R_{Gr_j}^i > G\_R$ we set value $C\_R_{Gr_j}^i > G\_R$, similarly if $C\_V_{Gr_j}^i > G\_Vr$ we set the value of $C\_V_{Gr_j}^i > G\_Vr$.

$$similarity_{Gr_j}^i = cosine\_sim(G\_centroid, movie\_centroid_{Gr_j}^i) \quad (15)$$

This research uses publicly available Internet Movie Database (IMDb) and Movie Database (TMDb) data. A multiclass classification model was used by researchers, and they outperformed all benchmark models with an accuracy of 96.8%. This study emphasizes how information systems may support industrial choices by using predictive and prescriptive data analysis.

Research [13] discusses e-commerce platforms and much information can be explored further. similar to a user-run community that assists other customers in making decisions by exchanging product information through reviews and ratings. By connecting products through evaluations, this research creates a bipartite network of products that Amazon.com sells in the ''musical instruments'' category. Using clustering, regression trees, and random forest techniques, researchers classified and identified trends in 2214 reviews. They also examined the relationship between review centrality, product ratings, and review usefulness. Below is a model that was built to solve the problem.

A social network is typically a collection of people or organizations that are linked together by a relationship. One can make links both offline and online [15]. Social links, sometimes known as relationships, bind people together. These ties can be formal or informal [16]. In theory, a simple network or graph (G) is described as a set of separate social entities (called nodes), represented by $V(G)$, and links (sometimes called edges), represented by $E(G)$. Symbolically, we can write the graph as a whole entity G, such as the tuple $G = (V(G), E(G))$. Actors are entities or nodes in a social network; they might be people, businesses, or other types of entities [17]. When two nodes are connected, they are regarded as neighbors, and the number of nodes and system elements is equal. As an illustration, Figure 7 has eight nodes or vertices, $V(G) = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$, and eight links or edges: $E(G) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$.

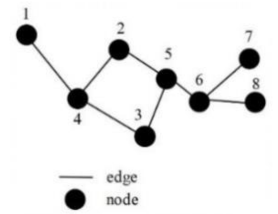

FIGURE 7. Representation of a social network G.

The building of an adjacency matrix A is frequently implied by information regarding the presence (or lack) of linkages between nodes. Let A be an adjacency matrix with the entries aij, so $aij = 1$ if node i s connected to node j and $aij = 0$.

Therefore, the adjacency matrix is square $|V| \times |V|$ such as:

$$A_{ij} = \begin{cases} 1, & \text{if node i is connented to node j} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

Bonacich's approach is quite adequate for centrality calculations because it considers neighbors' centrality and quality. Then, the vector centrality for node i, xi, is given in (17):

$$x_i = \frac{1}{\lambda} \sum\nolimits_{j=1}^{n} a_{ij} x_j, \quad i = 1, 2, 3, \ldots, n \tag{17}$$

Next, the number of short paths that connect pairs of nodes and pass through a specific node is reflected by intermediation centrality, sometimes called betweenness centrality, which quantifies the degree to which a node serves as a necessary mediator between other node links in the network. Because they connect communities that would not otherwise be connected, these nodes have an extremely high centrality [18]. One of the most used centrality metrics is connectedness centrality, sometimes known as intermediation [19]. It gauges how important a node is as a middleman between other nodes in the network. The intermediate centrality $C_B(x)$ of a node x in the network is given in Figure 8.

$$C_B(x) = \sum\nolimits_{s \neq t \in V(G)} \frac{\sigma_{st}(x)}{\sigma_{st}} \tag{18}$$

where $\sigma_{st}(x)$ shows the number of shortest paths between s and t that contain x, dan σ_{st} shows the number of all shortest paths between s and t in the network

The study has two types of nodes: product and reviewer, as shown in Figure 9. This is the case of a Bipartite network, or Bipartite graph whose nodes can be divided into separate and independent subsets V_1 and V_2. In a bipartite graph, each edge of E connects one vertex V_1 and one vertex V_1 [20]. Mathematically, the definition can be stated as follows.

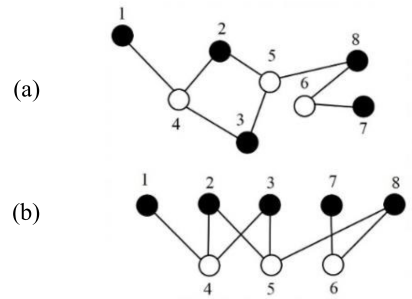

FIGURE 8. a and b—a bipartite or two-mode social network G'.

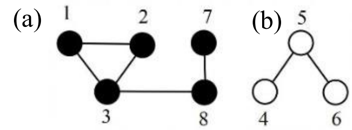

FIGURE 9. a) One mode projection of G1 (ex. products). B) One mode projection of G2 (ex. reviewers).

Definition 1 (Adapted from Banerjee et al. [21]): $G(V_1, V_2, E)$ called as bipartite graph, if $V(G) = V_1(G) \cup V_2(G)$ and $V_1(G) \cap V_2(G) = \emptyset$, and each edge is connected by two nodes $(v_1, v_2) \in E(G)$. G will be a complete bipartite graph. if $\forall v_1 \in V_1(G)$ and $\forall v_2 \in V_2(G)$, $(v_1, v_2) \in E(G)$.

In the case of a Bipartite Network, assuming $r = \#V1(G)$ and $s = \#V2(G)$, then the dual adjacency matrix corresponds to the following:

$$A = \left(\frac{O_{rr}}{B^T} \frac{B}{O_{ss}} \right) \tag{19}$$

where B is the $r \times s$ matrix, and $0_{r,r}$ and $0_{s,s}$ represent $r \times r$ and $s \times s$ zero matrix.

The results of the study show: (1) that a product's high number of reviews does not imply a high quality rating; (2) We see a rise in the number of reviews when they aid consumers in making decisions; (3) a definite positive correlation between review usefulness and product ratings; and (4) Weak correlations were found between quality indicators (product ratings and review usefulness) and centrality measures (betweenness and eigenvectors), which quantify the importance of products in the network, in respect to musical instruments. These findings imply that even with poor ratings and reviews that offer little value to customers, some products might be network necessity. The results of this study make several significant contributions to the enhancement of review service management, which in turn supports online consumer decision-making and the customer experience of e-commerce firms.

Content-Based filtering is a research technique [14] that uses a recommender system to connect information and offer consumers personalized services as a general information filtering tool. However, environmental circumstances have a major role in shaping user behavior, particularly in the context of the Internet of Things (IoT), making it challenging to model user preferences. In the study, a customized context-aware re-ranking (p-CAR) algorithm for the Internet of Things was proposed. Our primary objective is to enhance the recommender performance across multiple criteria, including popularity, variety, recall, and precision. The main concept is to use user preference behavior in various scenarios to reorder the ranking list. Each time, an ideal item that satisfies the target criteria is chosen from the candidate items and put to the reranking list. The reranking procedure is an iterative selection process. The context that is provided and the user's interests determine which things are chosen. Our method expresses user preferences and context interest in terms of probabilities. Furthermore, we describe the contextual personalization of various users through local personalization factors and manage the context influence using weight parameters. Using the Movielens 100K data set, we conduct experiments to validate our system and observe its real-world performance superiority over other techniques. The following is the model design built in this research.

Using an initial recommendation list $R(u)$ and a user u, we iteratively choose items from $R(u)$ to generate a re-ranked list $S(u)$. The criteria that are chosen are based on the following probabilistic mixed model.

$$P_r\left(v \mid u\right) + \lambda P_c(v, \bar{S}|u) \tag{20}$$

where, $P_r(v|u)$ is the relevance preference of user $u \in U$ on item $v \in V$, predicted by the initial recommendation algorithm. $P_c(v, \bar{S}|u)$ is the contextual variability of user preferences $u \in U$ not in the re-ranking list $S(u)$. λ is the load parameter.

Next, the researcher explains contextual diversity, namely $P_c(v, \bar{S}|u)$. Obtaining $P_c(v, \bar{S}|u)$, requires considering a set of contexts relating to the user and the items $C = \{c_1, c_2, \ldots, c_l\}$, produced by the marginal probability of the user on contextual variables. For convenience, we use $P(v, \bar{S}|u)$ instead of $P_c(v, \bar{S}|u)$ for the description. The calculation method is shown in Equation (21)

$$P\left(v, \bar{S} \mid u\right) = \sum\nolimits_{c\in C} P\left(v, \bar{S} \mid u\right) P(c|u) \tag{21}$$

where, $P(c|u)$ can be thought of as the user's preferences u for c context and $c \in CP(c|u) = 1$. The likelihood could be indicative of how many users favor context c. The entries in the reranked list S are fixed at the end of each cycle. Assuming that the item to be predicted is independent of the other items, we may decompose $P\left(v, \bar{S} \mid u\right)$ into the following components.

$$P\left(v, \bar{S} \mid c\right) = P(v|c)P(\bar{S}|c) \tag{22}$$

In this way, $P(v|c)$ can be thought of as coverage of items in c context and $P(\bar{S}|c)$ as consideration of contextual diversity, which measures the importance of c contexts not covered in the re-ranking S list.

In order to compute $P(\bar{S}|c)$, we additionally require that the importance estimations of the items in a rearranged Slist for the same context are unrelated to one another. Since we have to determine the likelihood that rearranging a S list won't satisfy the specified context, this assumption is plausible. Based on this presumption, the following is our method of calculation. We additionally presume that the assessments of the significance of items in a re-ranked S list for an identical context are unrelated to one another. This presumption makes sense because we have to figure out how likely it is that rearranging a S list won't satisfy the requirements. Based on this supposition, our computation technique is as follows.

$$P\left(\bar{S} \mid c\right) = \prod\nolimits_{i\in S} \left(1 - P(i|c, S)\right) \tag{23}$$

Equation (20) should be filled up with the values from Equations (21), (22), and (23). Equation (24) displays our final selection criterion.

$$\left(1-\lambda\right) P\left(v \mid u\right)+\lambda \sum\nolimits_{c\in C} P(c|u)P(v|c) \prod\nolimits_{i\in S} \left(1-P(i|c, S)\right) \tag{24}$$

The implementation of a probabilistic model comes next. To indicate item user preferences, researchers use the ranking scores produced by the first recommendation list. Researchers must normalize the scores to be normalized $(0, 1)$ in order to modify this model. Researchers use Min-Max standardization of processing methods, as seen in Equation (25).

$$P\left(v \mid u\right) = \frac{r\left(u, v\right) - r_{min} + 1}{r_{max} - r\left(u, v\right) + 1} \tag{25}$$

$r(u, v)$ is the u user's predicted value of the item v, given by the algorithm's initial ranking. Moreover, r_{min} and r_{max} represents the maximum and minimum predicted values. User preferences for context. Assuming that context preferences follow a uniform distribution—that is, that the preferences are the same for all c contexts—is the most fundamental approach to calculating context preferences.

$$P\left(c \mid u\right) = \frac{1}{|C|} \tag{26}$$

The total number of contexts is denoted by |C|. In order to quantify the variations in those settings based on the variations in those counts, researchers can also acquire the quantity of objects a user accesses in various situations from the user's historical record.

$$P\left(c \mid u\right) = \frac{n_c}{\sum_{d\in C} n_d} \tag{27}$$

The item is probably included in the context; n_c; denotes the number of items in the context c. Whether or whether an item is contained in the context, the relationship between it and the context is deterministic. Thus, we can make the following approximation.

$$P\left(v \mid c\right) = \begin{cases} 1, & v \in c \\ 0, & v \notin c \end{cases} \tag{28}$$

Reranking list coverage based on context. We must first understand the extent of S list reranking in each context before we can simulate context variety. The logical concept is that items in a c context that are included in a reranking S list that are ranked v will not be given priority when selections are made again. It can be deterministically estimated in this manner.

$$P(i \mid c, S) = \begin{cases} 1, & i \in c \\ 0, & i \notin c \end{cases} \tag{29}$$

In this manner, the currently selected $i \in c$ items will maximize their contribution to the context when every item in the S reordered list does not include c context.

$$P(\bar{S} \mid c) = \begin{cases} 1, & \forall\, i \in S, P(i \mid c) = 0 \\ 0, & \exists i \in S, P(i \mid c) = 1 \end{cases} \tag{30}$$

There is also a more efficient method that involves calculating the percentage of items that cover the c context in order to determine how well the list has been re-ranked in terms of context coverage. We experiment with both approaches in our paper.

$$P(i \mid c, S) = \frac{n_c}{|S|} \tag{31}$$

The term nc describes how many items in the S re-ranked list are covered by the c context. In order to represent the needs of various users for contextual personalization, researchers introduced the τ_u personalization factors in the contextual diversity module. Equation (32) displays the items' selection criteria.

$$P(v \mid u) + \lambda \tau_u P(v, \bar{S}|u) \tag{32}$$

$P(v|u)$ is the initial recommendation algorithm's estimated probability that the $u \in U$ user is interested in the $v \in V$ item. The weight parameter in the re-rank list is λ. $P(v, \bar{S}|u)$ is the likelihood that an u user is interested in a v item that is not in the $S.\tau_u$ is the u user's personal weight.

To calculate τ_u, user personalization preferences, we first calculate $P(c|u)$ user preferences for each $c \in C$ context. The calculation method is as follows

$$P(c|u) \triangleq \frac{\sum_v r(u, v) \mathbb{1}_{\{v \in c\}}}{\sum_{c'} \sum_v r(u, v) \mathbb{1}_{\{v \in c'\}}} \tag{33}$$

The u user rating on the v item is denoted by $r(u, v)$, and the indicator function $\mathbb{1}_A$, whose value is 1, indicates when the A condition is true. If not, the user's interests in each context are represented by 0. $P(c|u) \in [0, 1]$, and $\sum_c P(c|u) = 1$. While some consumers might be more engaged in particular contexts, others might have preferences that are the same in every context. We express the user's weight using information entropy in order to capture these features.

$$\tau_u \triangleq -\sum_{c \in C} P(c|u) log P(c|u) \tag{34}$$

The greater the τu value, the greater the diversity of user preferences for context.

Next, we will discuss the Collaborative Filtering (CF) approach and model from several studies.

Collaborative Filtering (CF) Approach:

The explanation below is based on Table 3 above. The amount of research discussed is based on the top 3 articles, namely studies [6], [23], and [24].

TABLE 3. Related work on collaborative filtering (CF).

Study	Year	Approach Model	Methods
[6]	2022	memory-based (ranking oriented)	Combination SVD (matrix decomposition) dan WPR (rating aggregation)
[23]	2021	memory-based (ranking oriented)	TDD-BPR
[24]	2021	memory-based (rating oriented)	Clustering-Based UPCSim (CB-UPCSim)

Research [6] describes a filtering rating-oriented collaborative Recommender System that uses rating aggregation. Recent collaborative filtering studies apply rating aggregation that considers item weight points to achieve more accurate recommendation ratings. However, as the number of things increases, this algorithm becomes more complex to execute. In order to decrease time complexity, this study suggests a new recommendation system that combines matrix decomposition and rating aggregation techniques. Singular Value Decomposition (SVD) is a technique used by the matrix decomposition approach to forecast unranked elements. The rating aggregation method applies a weighted point rating (WPR) to obtain recommended items. In order to see the construction of this research, the steps for building the SVD-WPR algorithm are explained.

There are four steps in the WPR process: figuring out how many ratings are equal, figuring out item points, figuring out weight points, and figuring out weight point ratings. The following are the formulas as a complete sequence of the WRP process. To explain the steps of the WPR process, the writers represent users as $U = \{u_1, u_2, ..., u_j, ..., u_m\}$ and $I = \{i_1, i_2, ..., i_j..., i_n\}$ item sets respectively. R_{u_j,i_j} and R_{u_k,i_j} shows the ratings given by u_j users in i_j and the ratings given by u_k user in i_j.

Using the formula found in Equation (35) determine the number of ratings that two users provide that are identical. Equation (36) calculates the number of identical ratings (S_{u_j,i_j}) by calculating the equal ratings for each product. ($SR(R_{u_j,i_j}, R_{u_k,i_j})$).

$$S_{u_j,i_j} = \sum_k^m SR\left(R_{u_j,i_j}, R_{u_k,i_j}\right) \tag{35}$$

$$SR\left(R_{u_j,i_j}, R_{u_k,i_j}\right) = \begin{cases} 1, & if\ R_{u_j,i_j} = R_{u_k,i_j} \\ 0, & if\ R_{u_j,i_j} = 0 \\ 0, & otherwise \end{cases} \tag{36}$$

Calculate the item ranking points (P_{u_j,i_j}) using the formula specified in Equation (37), then add the value 1 to the total ranking points (PR_{u_j,i_j,u_k}). Equation (38) formulates the number of ranking points.

$$P_{u_j,i_j} = 1 + \sum_{k=1}^{n} PR_{u_j,i_j,i_k} \tag{37}$$

$$PR_{u_j,i_j,u_k} = \begin{cases} 1, & if\ R_{u_j,i_j} > R_{u_j,i_k} \\ 1, & if\ R_{u_j,i_j} = R_{u_j,i_k}, S_{u_j,i_j} > S_{u_j,i_k} \\ 1, & if\ R_{u_j,i_j} = R_{u_j,i_k}, S_{u_j,i_j} = S_{u_j,i_k} \\ 0, & otherwise \end{cases} \tag{38}$$

Calculate the point weights using Equation (39).

$$WP_{u_j,i_j} = (S_{u_j,i_j} + R_{u_j,i_j})P_{u_j,i_j} \tag{39}$$

Calculate the weight point ranking (WPR) using Equation (40).

$$WPR_{i_j} = \sum_{k=1}^{m} WP_{u_k,i_j} \tag{40}$$

Next, we will explain SVD. The ranking matrix (R) is split into two matrices (P and Q) by SVD. $U \times \Sigma$ is represented by the matrix P, where Σ is a scalar such that the dimensions of the matrix U are unaffected by matrix multiplication. Equation (41) is the result, and the SVD in the recommender system formulation.

$$R_{m\times n} = P_{m\times f}(Q_{n\times f})^T \tag{41}$$

R is the rating matrix, P is the user matrix, and Q is the item matrix.

Next, researchers used the SVD algorithm to predict items that were not rated. SVD is a popular matrix decomposition technique that divides a big matrix into smaller ones. In this research, SVD functions to predict ratings that have not been rated. A user-factor matrix (Pmk) and a transposed item-factor matrix ($QTnk$) are the only remaining components of the ranking matrix (Rmn).m, n, and k represent users, items, and factors. A user's or an item's qualities are described by factors. Rank prediction calculation using the formula specified in Equation (42).

$$\hat{r}_{u,i} = \mu + b_u + b_i + p_u q_i^T \tag{42}$$

$\hat{r}_{u,i}$ represents the predicted u user rating on an i.μ item stating the average rating of all items. b_u and b_i is the bias value to reduce user prediction error and item average rating. p_u denotes the vector of user factors, where $p_u \in P_{mk}$ and $u = 1, 2, 3, \ldots, m$. While q_i^T denotes the transferred item-factor vector. $q_i \in Q_{nk}$ and $i = 1, 2, 3, \ldots, n$.

It determines user similarity from the user factor matrix after forecasting an unrated rating. To find environmental users, or the top users who have the highest similarity value to the target consumers, employ user similarity functions. The rating aggregation procedure will be supported by all users of the environment. In order to determine user similarity, our study used the Pearson correlation coefficient, which is related to Equation (43).

$$S_r(u, v) = \frac{\sum_{i\in I_u \cap I_v}(r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i\in I_u \cap I_v}(r_{ui} - \bar{r}_u)^2}.\sqrt{\sum_{i\in I_u \cap I_v}(r_{vi} - \bar{r}_v)^2}} \tag{43}$$

I_u and I_v each represents the set of items rated by u and v users. Next, r_{ui} and r_{vi} denote the rank value on the item by i item respectively by u and v users. Moreover, r_u and r_v determines the average ranking by u and v users. Finally, we explained one of the items that was rated jointly by u and v users.

Lastly, a faster running time of 13.502 seconds was obtained from the experimental findings using the MovieLens 100K dataset. Furthermore, as compared to the WP-Rank method, the normalized discounted cumulative profit score (NDCG) increased by 27.11%.

The following explanation is related to CF research [23] in Table 3. According to the study, one of the key elements that enables users to choose the following item is topic information. Researchers created a topic diversity discovery (TDD) model to record user preferences on topics in order to replicate this behavior. This is not the same as earlier work that just took item themes into account when applying subject information to recommender systems. The study takes into account user topic information in the recommender system and concentrates on item topic information. We create a TDD model to learn the item's subject information, and then we apply transfer learning to get the topic knowledge of the user. Next, we create a novel TDD-BPR recommender system by integrating the TDD model with the Bayesian Personalized Ranking (BPR) model. Below is an illustration using the formulas for building the research construction.

In order to capture the subject distribution of people and items, we created a f_{TDD} model to embed information about multi-embedding themes. From f_{TDD}, we may create a $d \in \mathbb{R}^{T\times 1}$ topic diversity vector, where the d values indicate the topic's proportion. We use 1-D convolution, a well-known method in natural language processing (NLP), to limit the data between themes [25], [26], in order to extract features from many embeddings. Afterwards, we generate the diversity of each f_{TDD}^{item} topic on an item using a feedforward layer with a sigmoid activation function. f_{TDD}^{item} is studied to minimize the binary cross entropy, which is formulated as follows:

$$\mathcal{L}_{TDD}^{item} = \frac{1}{M}\sum_{i=1}^{M}\sum_{t=1}^{T} y_i^T \log\left(\hat{y}_i^t\right) + (1 - y_i^t)\log(1 - \hat{y}_i^t) \tag{44}$$

So, the study adopted [27] as the basis for a new recommendation model called TDD-BPR. However, in order to represent things and users embedding topics, these researchers devised a multi-embedding approach. Rather than designing typical inner products to forecast user preference scores for each item, we instead develop topic-aware inner products to efficiently leverage subject diversity. Users will get a higher preference s score with items with a topic distribution similar to the users. The s reference score between i users and j

represented items based on topic-aware inner products is formulated as follows:

$$s\left(u_i, v_j\right) = \sum_{t=1}^{T} \left(w_t u_i^t\right) .v_j^t \tag{45}$$

where w_t denotes the t weight of the topic in i users from df_{TDD}^{user}. By making reference to themes, we enhance the preference scoring function, which enhances the performance of the model and adds context to help understand the recommendation outcomes. Consequently, we convert the diversity topic into L_R when we redefine the BPR loss function, and we subsequently minimize the loss function as

$$L_{RecSys} = -\frac{1}{N}\sum_{(i,j,k)\in D} loh\left(\sigma\left(s\left(u_i, v_j^+\right) - s\left(u_i, v_k^-\right)\right)\right) + \lambda\Theta \tag{46}$$

N stands for all of the training data, λ for the regularization terms' weights, and Θ for all of the model's training parameters. Thus far, we have shown how to use a multi-embedding technique to train an item f_{TDD}^{item} TDD model. Subsequently, the researcher modelled user-to-user TDD using transfer learning, which gives user topic distribution for user preference analysis. Furthermore, it demonstrates how to enhance recommendations by merging TDD and recommender systems. As a result, TDD is crucial to the model's functionality and dependability. We use combined training to optimize f_{TDD}^{item} and recommend models at the same time, ensuring that knowledge from TDD can be completely shared with recommender systems. You may rewrite the final loss function as follows:

$$\mathcal{L}_{all} = \alpha \times \mathcal{L}_{RecSys} + (1-\alpha) \times \mathcal{L}_{TDD}^{item} \tag{47}$$

Scholars employ the hyper-parameter, or trade-off α, which has a value range of 0 to 1, to modify the percentage of task recommendations and TDD in training. If set α is very close to 1, the model will be reduced to a broad recommendation without topic diversity information.. As a result, we will talk about how an affects how well the experimental recommendations work.

Experiments on three popular datasets—MovieLens-1 M, Pinterest, and TMDB—showed that the research findings were 71.8%, 91.2%, and 96.1% in Hit Ratio@10. Based on empirical data, TDD-BPR is able to identify potentially relevant topic information in order to better understand user preferences and improve suggestion performance.

Table 3 in research [24] explains similarity metrics-based memory-based collaborative filtering. User ratings and behavior scores have been taken into account in similarity metrics recently. The consumer's preferences for each product category (genre) are displayed in the user behavior score. User behavior scores added to similarity metrics produce more computationally complex results. To reduce the computational complexity, the researchers combined clustering and user similarity methods based on behavioral scores. By employing the Silhouette Coefficient to calculate the number of clusters, the clustering approach employs k-means clustering. User Profile Correlation Based similarities (UPC-Sim) is used to score similarities between user behavior and the model. Below is a complete explanation of the steps for preparing the algorithm.

The Silhouette Coefficient is one of the metrics utilized in the intrinsic technique. This method measures the similarity of an object to its cluster (cohesion) compared to other clusters (separation). The following steps explain how to calculate the Silhouette Coefficient value [28].

Apply the formula given in Equation (48) to determine the average distance between documents in a cluster.

$$a\left(i\right) = \frac{1}{|A|-1}\sum_{j\in A, j\neq i} d\left(i,j\right) \tag{48}$$

$d(i,j)$ is the distance between documents i and j, where j is a different document inside the same cluster A. Next, use the formula found in Equation (49) to determine the average distance between each document in each of the other clusters and the I document. Then, use Equation (50) to determine the minimal average distance.

$$d\left(i, C\right) = \frac{1}{|C|}\sum_{j\in C} d(i,j) \tag{49}$$

$$b\left(i\right) = min_{c\neq A} d(i, C) \tag{50}$$

Next, using Equation (51) to get the value of the Silhouette Coefficient.

$$S\left(i\right) = \frac{b\left(i\right) - a(i)}{\max(a\left(i\right), b\left(i\right))} \tag{51}$$

The density of clusters containing object i is denoted by the value $a(i)$. The denser the cluster, the smaller the value $a(i)$. In the meantime, the object's separation from other clusters is indicated by the value $b(i)$. The object i is farther from another cluster the larger the value $b(i)$. The object's Silhouette Coefficient i will be near to 1 if the values of $a(i)$ and $b(i)$ are tiny and large, respectively. This means that the cluster contains i very dense objects, and objects i are far from other clusters. Conversely, the object's silhouette coefficient i will be near to -1 if the value of $a(i)$ is very large and the value of $b(i)$ is small. This means that the cluster containing the object i is not congested, and the object i is congested very close to another cluster.

The two procedures that make up memory-based approaches are ranking prediction and similarity calculation. Equation (52) is referred to as UPCSim [24] in the similarity calculations conducted in this study.

$$S\left(x, y\right) = \alpha S_r\left(x, y\right) + \beta S_b\left(x, y\right) \tag{52}$$

$S(x, y)$ represents the final similarity between users x and $y.S_r\left(x, y\right)$ shows the user the similarity score based on the rating between users x and y, whose formula refers to Equation (53). $S_b\left(x, y\right)$ states the similarity of user behavior scores x and y, whose equation is Equation (54). Lastly, using multiple linear regression, a and b represent the correlation coefficients between user profile attributes and user assessment/behavior ratings [28].

$$S_r(x,y) = \frac{\sum_{p\in P_x\cap P_y}(r_{xp}-\bar{r}_x)(r_{yp}-\bar{r}_y)}{\sqrt{\sum_{p\in P_x\cap P_y}(r_{xp}-\bar{r}_x)^2}.\sqrt{\sum_{p\in P_x\cap P_y}(r_{yp}-\bar{r}_y)^2}} \tag{53}$$

P_x and P_y state the set of products rated by x and y users respectively. Next, r_{xp} and r_{yp} state the product rating values p by x and y users respectively. Moreover, $\bar{r}_x$ and $\bar{r}_y$ depicts the average rating for x and y users. Last, p is one of the shared values of the product by x and y users.

$$S_b(x,y) = \frac{\sum_{g\in G_x\cap G_y}(P_{xg}-\bar{P}_x)(P_{yg}-\bar{P}_y)}{\sqrt{\sum_{\in G_x\cap G_y}(P_{xg}-\bar{P}_x)^2}.\sqrt{\sum_{\in G_x\cap G_y}(P_{yg}-\bar{P}_y)^2}} \tag{54}$$

G_x and G_y shows the set of product types rated by x and y users. Next, P_{xg} and P_{yg} denotes the probability of the product g type given by the x and y users. Moreover, Px and Py denote the average probability of the product type of the x and y users. Finally, g is a type of joint product of x and y users. Next, we need to figure out how many neighbors are closest to us (k), before we can forecast a rating. The k value in this study increased by 10 and varied from 10 to 100 [29], [30]. The formula for calculating rating predictions for products that have not been rated is expressed in Equation (55) [31].

$$\hat{r}_{xp} = \bar{r}_x + \frac{\sum_{y\in NN_x} S(x,y).(r_{yp}-\bar{r}_y)}{\sum_{y\in NN_x}|S(x,y)|} \tag{55}$$

$\hat{r}_{xp}$ is a prediction of rating scores from x users to p products. $y \in NN_x$ represents the set of users who have the closest similarity to the x users. $S(x,y)$ shows the final similarity between x and y users. $\bar{r}_x$ dan $\bar{r}_y$ The average user rating scores indicate the final similarity between x and y users. Finally, r_{yp} rating score y users give to the p products. In conclusion, the MovieLens 100k dataset experiment findings demonstrate a faster compute time of 4.16 seconds. Furthermore, in comparison to the baseline algorithm, the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) dropped by 1.88% and 1.46%, respectively.

Hybrid Filtering (HBF) Approach:

This explanation discusses three studies related to Hybrid Filtering (HBF). This activity aims to provide an overview of the latest research shown in Table 4 and the methods used in HBF. The following is an explanation of the first [32], second [10] and third [9] HBF studies.

The research [32] attempts to provide practical suggestions to each user based on their interests and behavior. Recommender systems typically support users in making decisions by matching their personal preferences. The necessity to create such systems is unavoidable given the proliferation of online information, e-commerce, online education, and, eventually, social networks. Among the key methods in recommender systems are collaborative and content-based filtering. In the meantime, recommender systems have made extensive use of this technology due to the notable advancements in deep learning in recent years. This paper presents a deep autoencoder network-based hybrid social recommender system. The suggested method makes advantage of user social influence, Content-Based filtering, and collaborative. Based on their social traits and Twitter behavior, each user's social influence is determined. The following illustration will be provided to provide complete information about the approach and methods used.

TABLE 4. Related work on hybrid filtering (HBF).

Study	Year	Approach Model	Methods
[32]	2020	memory-based (rating oriented)	a hybrid social recommender system utilizing a deep autoencoder network is introduced
[9]	2023	memory-based (rating oriented)	two recommendation techniques based on content and matrix-based decomposition
[10]	2021	memory-based (rating oriented)	Te combination of DNN and DBSCAN clustering algorithm in the CBR core, Te combination of hybrid similarity criteria and the new Pro-FriendLink algorithm in CRS, The proposed Pro-FriendLink algorithm for a new method in RSs

This work seeks to provide a hybrid recommender system that offers precise and useful usage recommendations of user preferences and interests, social data, and feature films, as was indicated in the preceding section. By utilizing a deep network autoencoder, the suggested approach lessens the issue of data sparsity. For evaluation reasons, a required data set has been gathered from MovieTweetings and the Open Movie Database. The evaluation's findings demonstrate that, in comparison to other cutting-edge techniques, the suggested approach's efficacy and accuracy have increased.

First, one of the significant differences between SRDNet and other available approaches in movie recommendation systems is its social aspect. For this purpose, the social influence of each user is calculated based on his social characteristics. To obtain the user's social influence, we calculate the average number of published Tweets liked by the user ($\bar{T}$), the average number of retweets ($\bar{R}$), and the average number of Likes ($\bar{F}$) assigned to each tweet, as calculated using Eq. (56), (57), and (58), respectively.

$$\bar{T} = \frac{M}{N} \tag{56}$$

$$\bar{R} = \frac{1}{M} \sum_{i=1}^{M} Retweet\_countT_i \tag{57}$$

$$\bar{R} = \frac{1}{M} \sum_{i=1}^{M} Favorite\_countT_i \tag{58}$$

In the above equation (56), N is the total number of users, M is the total number of tweets, Retweet_count T is the number of $Retweet\_countT_i$ is assigned to tweet T_i, and $Favorite\_countT_i$ s the number of likes assigned to tweet T_i (favorite tweet).

If we assume that $Tweer\_count_u$ represents the total number of tweets that a user U_i has published, we can use Equation (59) to determine the expected average number of likes and Equation (60) to determine the expected average number of retweets that a user U_i has published.

$$\overline{FC} = \overline{F} * Tweet\_count_{U_i} \tag{59}$$

$$\overline{RC} = \overline{R} * Tweet\_count_{U_i} \tag{60}$$

As a result, when a user's tweets receive more retweets than $\overline{RC}$ and more likes than $\overline{FC}$, their social influence gains significance. Lastly, Equation (61) is used to determine each user's social influence based on the social data that was retrieved from the dataset and Equations (56)–(60):

$$Sosial\ InfluenceU_i = \left[\frac{Tweet\_count_{U_i}}{\overline{T}} + \frac{\sum_{i=1}^{Tweet\_count_{U_i}} Favorite\_countTU_i}{\overline{Fc}} + \frac{\sum_{i=1}^{Tweet\_count_{U_i}} Retweet\_countTU_i}{\overline{Rc}} \right] \tag{61}$$

After implementing the procedure, one of the most often used metrics in recommender systems' the Pearson Correlation Coefficient is typically utilized to measure the similarity between the target user and other users. Equation (62) can be used to find the Pearson correlation between u and v users.

$$Sim\,(u, v) = \frac{\sum_{\alpha \in o_{u,v}} (r_{u,\alpha} - \overline{r_u})(r_{v,\alpha} - \overline{r_v})}{\sqrt{\sum_{\alpha \in o_{u,v}} (r_{u,\alpha} - \overline{r_u})^2} \sqrt{\sum_{\alpha \in o_{u,v}} (r_{v,\alpha} - \overline{r_v})^2}} \tag{62}$$

Finally, SRDNet takes advantage of each machine's unique qualities because it employs two machine recommendations. Each recommendation engine has a different weighting factor based on how many user ratings it has. In general, the recommendation engine combination is formulated by Equation (63):

$$Final_{rs} = \alpha CF_{rs} + \beta CB_{rs} \text{ Where : } \alpha + \beta = 1 \tag{63}$$

The aforementioned formula demonstrates that two suggestions from Content-Based filtering (CB_{rs}) and Collaborative filtering (CF_{rs}) are combined to create the final list of rs Dations. The weighting coefficients of every set in the final recommendation list are represented by the α dan β values. Users' film ratings determine the values of α and β. For instance, if a user has watched and rated two movies, α and β are regarded as 0,2 and 0,8, respectively. Based on the data set's average of 14 ratings per user, this value has been determined. Collaborative filtering is necessary, even for those who have rated and watched more movies. 123 rs is more likely to take advantage of other users' interests in order to produce relevant recommendations. Nevertheless, this is also evident when examining the data set's user-rated movie statistics. The suggested dataset has cold start issues, similar to other real-world situations, meaning that many viewers still need to rate a sizable number of movies. The average number of ratings per user in our sample is 14. Consequently, the weights in this interval are divided so that, in addition to minimizing the effect of cold start, this increases the accuracy of recommendations for users who have rated more films. This is because the collaborative filtering recommender engine should focus on users who rated the movie lower.

In research [9] in Table 4, the current film recommendation algorithm is improved using an item-based collaborative filtering algorithm for the item size similarity process. In the recommendation process, two more applicable recommendation methods are considered: collaborative filtering content-based recommendation and matrix decomposition-based recommendation. The research saves users time searching, viewing, filtering, and finding information about their potential film preferences. Below are several explanations of the approach used and the methods that make up the algorithm.

The main idea of the research algorithm is to calculate the similarity between two users using some similarity measure based on the user's item association data, such as the user's evaluation of information about the item, and thus obtain a group of users containing similarity values. The recommendation value is related to the similarity between the user and the ratings of items recommended by the user's neighbors, and the recommendation results are ranked in descending order according to the recommendation value. User-based recommendations are highly community-based, time-sensitive, and influenced by user size. However, it also suffers from high data sparsity and poor scalability. Therefore, a recommender system based on the nearest neighbor algorithm may be unable to make practical recommendations for users whose relevant data is too little. The core algorithm in item-based collaborative filtering recommendation algorithms calculates item similarity. Then, it uses the similarity between items to predict the user's rating system for candidate items. Therefore, the choice of similarity measure is decisive in the final results. The three main methods are included as follows:

(1) Euclidean distance calculation formula

According to the general evaluation of Items as dimensions between users, a multidimensional space is formed, and the coordinate system $X(s_1, s_2, \ldots\ldots, s_i)$ can find the user's location in this multidimensional space. Then, the other two dimensions Location Distance (X, Y) can reflect the degree of similarity between two users or items. The calculation formula is shown in Equation (64):

$$dist\ (X, Y) = \sqrt{\sum\nolimits_{i=1}^{n} (x_i - y_i)^2} \tag{64}$$

(2) Cosine similarity

Cosine similarity is a measure of similarity between two texts using the cosine of the angle between two vectors in vector space, focusing more on the directional difference between two vectors rather than a distance measure. In general, after obtaining the vector representation of two users or items in the resulting collaborative filtering matrix, the similarity between the two texts can be calculated using cosine similarity, the result of which does not depend on the length of the vector and is only related to the direction in which the vector points. The cosine between vectors can be found using the Euclidean dot product formula:

$$a.b = ||a||\, ||b||\, cos\theta \tag{65}$$

(3) Pearson correlation coefficient

Pearson correlation coefficient, also known as Pearson product-moment correlation coefficient, is a measure of the linear correlation between two variables, X and Y, with a value between -1 and 1. Pearson correlation coefficient between two variables is defined as the quotient of the covariance and standard deviation between the two variables and is calculated as shown in Equation (66):

$$\rho\ (X, Y) = \frac{cov(x, y)}{\sigma_x \sigma_y} = \frac{\mathrm{E}[(X - \mu_x)(Y - \mu_y)]}{\sigma_x \sigma_y} \tag{66}$$

The above Equation defines the overall correlation coefficient, usually expressed with the lowercase letter ρ. Estimating the sample covariance and standard deviation gives the Pearson correlation coefficient, which is often expressed in lower case γ and calculated as shown in Equation (67):

$$\gamma = \frac{\sum_{i=1}^{n} (X_i - \mu_x)(Y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n} (X_i - \mu_x)^2}\sqrt{\sum_{i=1}^{n} (Y_i - \mu_y)^2}} \tag{67}$$

Next, the research's main problem is implementing a personalized film recommendation algorithm based on the MovieLens website's film data collection. According to the characteristics of the recommended films, the similarity measure used in the collaborative filtering model is increased gradually, and the number of users who jointly evaluate two films based on their ratings will be used to determine the degree of similarity between two films more precisely. In a film recommendation site, each user only views films for a limited time, so the data obtained is limited. In such cases, a solution is required using Equation (68).

$$Sparsity = 1 - \frac{Number\ of\ ratings}{Number\ of\ users \times Number\ of\ films} \tag{68}$$

In addition, the study implemented time decay. For example, the user rating time for film x is Tx and the user rating time for film y is Ty, the formula for the time factor in this paper is shown in Equation (69). The smaller the time interval, the smaller the time factor, and the minimum case results from a time factor of 1 when the time interval is 0.

$$\alpha = \ln(e + (|T_x - T_y|)) \tag{69}$$

The formula for calculating cosine-similarity after adding the time factors is shown in Equation (70).

$$Similarity = cos\theta = \frac{A - B}{||A|\,|B||} = \frac{\sum_{i=1}^{n} A_i \times B_i \times \frac{1}{\alpha}}{\sqrt{\sum_{i=1}^{n} (A_i)^2}\sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{70}$$

Next, we will discuss matrix decomposition, namely the decomposition of a high-dimensional matrix into a low-dimensional matrix representation. This approach reduces the dimensions of the target matrix and allows the program to run faster. The most widely used decomposition is single-value decomposition (SVD). SVD decomposition of matrix R results in (71).

$$R \approx U.S.V^t \tag{71}$$

U and V denote two orthogonal matrices of dimensions $m \times r$ and $r \times n$, respectively, used to represent potential factors of users and items, respectively. S is a diagonal matrix of order $r \times r$, consisting of a single value of the original matrix, similar to the orthogonal matrix decomposition of a matrix. After getting a single value of r the selection of singular values can be carried out, the first five larger single values are selected, where the times of the diagonal matrix S becomes S_k. U and V become k- dimensional accordingly, and the sum is obtained; the turn achieves the subtraction dimensions required in this case paper. Next, these three matrices are given points and multiplied to get the desired reconstruction matrix (72)

$$R_k \approx U_k.S_k.V_k^t \tag{72}$$

Using a decomposed matrix, a suitable prediction can be made for an unrated user item by dotting the i th row and j th column and adding up the mean value of the previous user rating data, i.e., after decentralization, to get the predicted i user rating value for j item, as shown in (73):

$$r_{i,j} = \overline{r_1} + U_k.\sqrt{S_k}^T(i)\,.\sqrt{S_k}.V_k^T(j) \tag{73}$$

The SVD matrix decomposition algorithm has excellent applications in recommender systems, improving the performance of recommender systems and the realization of real-time recommendations even in the extreme case of sparse matrices, laying a good foundation for more matrix decomposition recommendation algorithms in the future.

The previously introduced SVD algorithm needs to calculate the eigenvectors and eigenvalues of the matrix during the computation. The matrix cannot have zero values during the computation, which is not friendly to the program, and the computing process is complicated. The Regularized Singular Value Decomposition (RSVD) algorithm further refined the SVD algorithm, a viral recommender system algorithm nowadays. In the RSVD model, the initial collaborative filtering matrix R is decomposed into two low-rank matrices U and V with matrix dimensions $m \times f$ dan $n \times f$, respectively, where $f \ll min(m, n)$, The mathematical representation of the algorithm is given in (74).

$$R \approx U.V^T \tag{74}$$

The U matrix is a potential user factor matrix showing the h user's preference for the item's feature factors, and the V matrix is the item potential factor matrix, which shows the composition of each item's feature factors. Each row of the U matrix shows the popularity of each user for different item feature attributes, and each row of the V matrix shows the weight of each item's feature attributes. Each column of the potential factor matrix represents a potential factor, and the potential factors form a space vector, representing the user's or item's characteristics. In other words, the model matrix decomposition extracts f potential factors from the original matrix, which indicates this user's degree of preference for the item. User i's r preferences for item j can be obtained from equation (75).

$$\hat{R}_{i,j} = U_i.V_j^T \tag{75}$$

where and denote the i th row of potential factors of the prototype matrix U and the j th row of V, respectively, and the objective function L can be determined to calculate the matrices U and V, namely Equation (76):

$$lossfunction = \sum\nolimits_{r(i,j)\neq 0} \left(R_{i,j} - \hat{R}_{i,j}\right)^2 + \lambda \sum U_i^2 + \lambda \sum V_j^2 \tag{76}$$

$r(i, j)$ shows the original rating of the $\hat{R}_{i,j}$ user shows the predicted value of the user's rating using this model, and λ s a regularization parameter that varies continuously with the operation. The regularization terms U and V are added to the objective function in Equation (76), which controls project complexity. If λ is too small, these constraints are not enough to reduce complexity, and if λ is too large, it may indicate the loss of some critical parameters, leading to a decrease in model precision. In the actual operation process, it is necessary to balance the degree of suitability and accuracy of the model with continuous network operation to find a suitable λ.

In further research [10], we will discuss a recommendation system based on a hybrid approach. It contains four stages: (1) Currently, CBRS is being used to classify new users using the Deep Neural Network method (DNN) and cluster all users using application density-based spatial grouping with a noise method (DBScan). (2) System of Collaborative Recommendations (CRS) Similarity is determined using a threshold (lambda) between new users and users in the chosen category, in accordance with Hybrid Similarity Criteria. Age, gender, and occupation are used to construct the similarity criteria. The most similar users are selected by collaborative recommendation algorithms for new users. Next, using a proximity matrix, better-rated streaming services are recommended to brand-new customers. (3) Analyzing the dataset to determine the similarity of individuals connected by the link and making adjustments to the Friendlink algorithm. (4) This stage is concerned with combining the results of the cooperative recommender system with the Friendlink algorithm's enhancement. Below is a complete explanation of these stages.

First, calculate the similarity between the new and adjacent users using the following equation.

$$sim\,(n.u) = \frac{\sum_{j=1}^{I} SF_j * w_j}{\sum_{j=1}^{I} w_j} \tag{77}$$

The weight of the relevant attribute is denoted by wj, while the similarity value of characteristic j is represented by SF_j. These researchers have grouped users and calculated their similarities using attributes like age, gender, and occupation. Then, they have assigned varying weights to these similarities based on how important they are.

In this study, user similarity is determined using a hybrid set of criteria. We have defined an SF function (at1, at2) with values in the range of [0, 1] for each characteristic. The similarity of two attributes linked to a pair of users is computed by this function. There are two main classes of features that we evaluate while determining similarity, given the nature of those qualities. The similarity criterion for numerical features, like age, is defined as follows (78):

$$W_{age} = \begin{cases} \left(1 - \dfrac{|Diff|}{Diff_{max}}\right)^{\beta}, & if\ |Diff| \leq Diff_{max} \\ 0, & if\ |Diff| > Diff_{max} \end{cases} \tag{78}$$

String features In Equation (79), the Equation for calculating similarity based on string features is shown.

$$W_{gender} = \begin{cases} 1, & if\ att\ 1 == att2 \\ 0, & if\ att\ 1 == att2 \end{cases} \tag{79}$$

Either 1 or 0 is reset regardless of whether feature values 1 and 2 are the same. Next, the following Equation is used to determine each user's anticipated ranking.

$$R_{n_j.i_b} = \frac{\sum_{u\in NG} Sim\left(n_j.u\right).r_{u.i_b}}{\sum_{u\in NG} Sim\left(n_j.u\right)} + TF \tag{80}$$

The rating that user u gave item i in the list of neighboring users is represented by $r_{u.i_b}$. The anticipated rating for each item will therefore be the predicted forecast that the one with the highest rating will be chosen, using the prediction formula above. The TF number shows how useful both new and seasoned users are for ranking. The various ratings provided by the researchers are likely to range from 0.1 to 1, which can have an impact on the anticipated total rating, and if the user is new, the value is TF = 0.

Following the determination of the movie service for new users, the FriendLink Algorithm, or stage 3, gets started. The movie is now being transmitted using the communication data set of the user. During this stage, movies are chosen based on the connections between new and similar users. Various similarity criteria have been put out thus far to determine how similar users X and Y of social network G graphs are to one another. The majority of these factors determine the overall similarity based on links by calculating the degree of nodes and their proximity inside the network. The following four important social network graph components serve as the foundation for the suggested similarity criterion. The degree of the nodes, user popularity, number of routes, and node balance are some of these variables. Given these elements, relation (81) is described as follows:

$$Sim_{MyApp}\left(X,Y\right)=AVG\left(\left(\frac{Neib(Y)}{N}\right)+\frac{|Paths^{l}_{X.Y}|}{|Paths_{X.Y}|}+\frac{K_Y}{\sum_{i=1}^{n}D(K_i)}+\frac{|Neib(X)\cap Neib(Y)|}{Avg(K_X.K_Y)}\right) \tag{81}$$

$Sim_{MyApp}\left(X,Y\right)$ in the connection above represents the suggested similarity criterion for social network link prediction. The number of users next to node Y is shown by $|Neib\left(Y\right)|$, the total number of nodes in the graph is represented by N, and the number of paths of length L that connect node X and node Y in graph G is indicated by $|Paths^{l}_{X.Y}|$. K_y represents the degree of node Y, $\sum_{i=1}^{n} D(y_i)$ is the total number of pathways going to node Y, and | pathways $Neib(X)\cap Neib(Y)$ is the number of nearby nodes that users of node X and node Y have; $Avg(K_X \cdot K_Y)$ is the average degree of node X and node Y. $\sum_{i=1}^{n} D(y_i)$ is the overall degree of the network. In conclusion, the degree of similarity between users X and Y can be determined using relation (82). This criterion's ability to precisely determine the similarity between two users on a social network is among its most important characteristics. This level of accuracy makes recommended users or friends more attractive.

Next, utilize the Friendlink tool, a popular link prediction tool on social networks for forecasting upcoming relationships, particularly romantic ones.

$$Sim\left(n_j.u_j\right)=\sum_{i=2}^{l}\frac{1}{i-1}\cdot\frac{|path^{i}_{n_j.u_j}|}{\prod_{y=2}^{i}(n-y)} \tag{82}$$

In the final stage, this research has also refined the Friend Links formula to increase the accuracy of link predictions. The formula will change to apply the degree of a neighborhood of nodes.

$$Sim\left(n_j.u_j\right)=\left(\sum_{i=2}^{l}\frac{1}{i-1}\cdot\frac{|path^{i}_{n_j.u_j}|}{\prod_{y=2}^{i}(n-y)}\right)*\left(\frac{D_{u_j}}{N}\right) \tag{83}$$

N is the total number of node degrees. The target node's degree is represented by D_{u_j}. We attempted to enhance the paper's results by enhancing its similarity formula and applying the Friendlink method. Pro-Friendlink is the name of the method that is proposed.

The next exciting thing to discuss is the problems in the Recommender System and their explanations. Below is an explanation.

B. PROBLEM

RQ2 : What problems arise in solving problems in recommender systems?

Figure 10 shows several problems encountered in the process of using the Recommender System, both in Content-Based Filtering (CBF), Collaborative Filtering (CF), and Hybrid Filtering (HBF). Five problems were identified from the results of the article review: accuracy, scalability, running time, sparsity, and cold start. We found that the most significant percentage of problems with the Recommender System was an accuracy problem of 65%. Sparsity problem follows second place at 14%. Then, orders 3, 4, and 5 are the cold start problem at 10%, the running time problem at 8%, and the Scalability Problem at 3%.

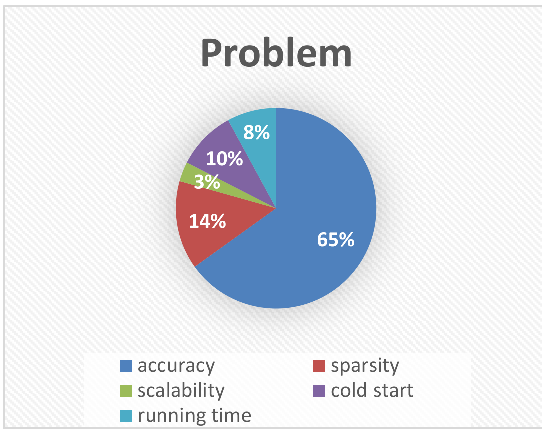

FIGURE 10. Distribution of problems in percentage in recommender systems.

Below, we explain the problems that occurred while implementing the Recommender System.

Finding 1 Cold Start Problem. This problem refers to the situation where the recommender does not have sufficient information about the user or item to make relevant predictions [33]. This is one of the big problems that degrades the performance of recommender systems. The new user's profile or item will be empty because he or she has not rated any items; thus, his or her tastes are unknown to the system.

TABLE 5. Distribution of problem data from the recommender system.

RS Approach	Accuracy	Sparcity	Scalability	Cold Start	Running Time
CBF	[11], [36], [13], [37], [14]		[38], [39]		
CF	[40], [41], [42], [43], [44], [45], [46], [24], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [58], [61], [62], [63], [15], [64], [65], [66]. [67]	[68], [23], [69], [70], [71], [72], [73], [9]	[5]	[74], [75], [71], [50]	[6], [76], [43], [24], [77]
HBF	[32], [12], [78], [36], [79], [8]				

Finding 2 Sparsity Problem. This problem occurs due to a need for sufficient information, even though only a few of the total number of items available in the database are assessed by the user. This invariably leads to sparse user-item matrices, an inability to find successful resemble, and weak recommendations. In addition, data sparsity always gives rise to coverage problems, namely the percentage of items in the system that can be used as recommendations [34].

*Finding 3 Scalability Problem.*Another problem in recommendation algorithms is that regular computing grows linearly with the number of users and items. When the number of data sets is limited, efficient recommendation techniques may not be able to produce a satisfactory number of recommendations. This is caused when the volume of a limited data set is increased. Therefore, it is critical to apply recommendation techniques that can scale successfully as the number of datasets in the database increases. The method used to solve scalability problems and speed up making recommendations is based on Dimension reduction techniques, such as the Singular Value Decomposition (SVD) method, which can produce reliable and efficient recommendations [35].

Finding 4 Running Time Problem. Running Time Problem is an exciting thing to improve in every research. By using algorithms with both the CF and HBF abbreviations, improvements can be made to increase the time duration faster than the previous time (using other methods). Thus, new findings in the form of developing the Recommender System algorithm can continue to be carried out over an increasing duration of time to achieve perfection.

Finding 5 Accuracy Problem. Accuracy problems consistently arise in every Recommender System research. More than half of the referenced article data has the same problem. Of the three approaches, the majority leads to Collaborative Filtering. Followed by Content-Based Filtering and Hybrid Filtering. Each algorithm improvement hopes to increase accuracy with various research methods. Thus, improving the performance of the Recommender System accuracy is exciting and continues to be carried out from time to time.

Below in Table 5, we present some problem distribution data, which we summarize in the Recommender System.

C. EVALUATION METHOD

RQ3 : What evaluation techniques are suitable for solving problems in recommender systems?

To answer question RQ3 can be seen in Figure 11 above. The Evaluation Method from the SLR results obtained seven types of results. These types include Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Normalized Discounted Cumulative Gain (NDCG), Precision, Recall, F1, and others.

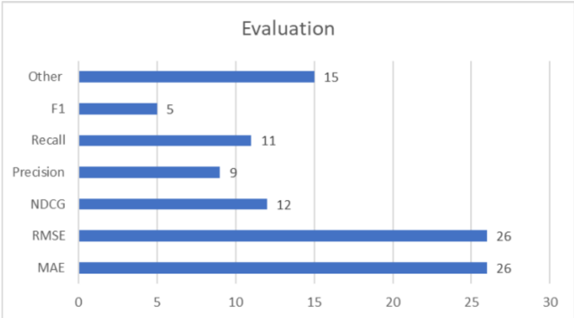

FIGURE 11. Distribution of evaluation methods in recommender systems.

Considering the total distribution, RMSE and MAE have the same number, namely 26 studies. Furthermore, NDCG has 11 studies, Recall has eight studies, Precision has seven studies, F1 has five studies, and finally, the most others are 40 studies. Considering the distribution's composition, something is interesting; namely, the number of studies that use the MAE and RMSE Evaluation Methods is the same. This means that the two are only sometimes compared in every study. There are several factors related to why researchers often use MAE and RMSE as evaluation methods, such as:

1) Probability theory provides a logical answer to the choice between RMSE and MAE [80]. One of these metrics is optimal in its correct application, although more may be needed in practice. Refining the model, transforming the data, using robust statistics, or constructing better probabilities can provide better results for these cases. The last option is the most versatile, although there are pragmatic reasons for choosing the others.
2) RMSE is optimal for normally distributed errors, although they incorrectly suggest that MAE only applies to uniformly distributed errors. While it is true that MAE is more powerful, there are better alternatives [81]. Most importantly, neither party provides the theoretical justification behind the metrics, nor do they

TABLE 6. Evaluation method based on researchers.

Evaluation Techniques	Formula	References
Root Mean Squared Error (RMSE)	$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|\|y(i)-\hat{y}(i)\|\|^2}{N}}$	[76], [68], [82], [43], [69], [44], [45], [83], [24], [46], [84], [32], [12], [78], [1], [36], [47], [48], [49], [13], [70], [51], [58], [65], [66], [67]
Mean Absolute Error (MAE)	$MAE = \frac{1}{N}\sum_{i=1}^{N} \|\|y(i) - \hat{y}(i)\|\|$	[76], [68], [82], [43], [69], [45], [83], [23], [46], [84], [32], [12], [78], [1], [36], [47], [48], [49], [13], [70], [75], [73], [52], [53], [65], [67], [79]
Normalized Discounted Cumulative Gain (NDCG)	$NDCG@K = \frac{DCG@k}{IDCG@K}$	[6], [68], [23], [41], [71], [50], [77], [5], [54], [56], [61], [62]
Precision	$Precision(u) = \frac{\|Recommended(u) \cap Testing(u)\|}{\|Recommended(u)\|}$	[49], [50], [72], [73], [14], [15], [56], [65], [67]
Recall	$Recall(u) = \frac{\|Recommended(u) \cap Testing(u)\|}{\|Testing(u)\|}$	[49], [71], [72], [73], [53], [5], [56], [64], [65], [67]
F1	$F1 = 2.\frac{precision.recall}{precision+recall}$	[49], [73], [64], [65], [67]
other	t-value, Hit Ratio (HR), the MOABC and NSGA-II, etc.	[40], [82], [23], [29], [11], [49], [77], [72], [37], [39], [60], [62]

adequately introduce the extensive literature on this topic.

In other research, evaluating methods depends on the case being studied. For example, use t-value, Hit Ratio (HR), the MOABC and NSGA-II, etc. In this way, all the information in the evaluation method becomes easier to understand according to its use with its respective advantages and disadvantages.

D. DATASETS

RQ4 : What datasets are currently popular for research in recommender systems?

To answer question RQ4, it can be seen in Figure 12 above. There are several types of datasets obtained on this SLR, including Movielens 100K, Movielens 1M, Movilens 10M, Netflix, Yahoo Music, Film Trust, Epinions, BookCrossing, Movie Tweetings, Yelp, Ciao, Amazon, and others. Almost all of this data is secondary data rather than primary data.

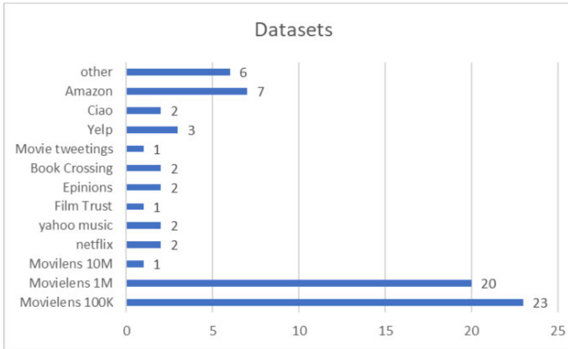

FIGURE 12. Distribution of datasets obtained from the recommender system.

The dataset that is often used in the Recommender System is MovieLens 100K. The data is ready to be applied to various Recommender algorithms.

MovieLens bases its recommendations on input provided by website users, such as movie ratings [85]. The site uses various recommender algorithms, including collaborative filtering algorithms like item-item, user-user, and regulated SVD.

Additionally, to overcome the cold-start problem for new users, MovieLens uses a preference elicitation method [86]. The system asks new users to rate how much they enjoyed watching different groups of films (for example, films with dark humor versus romantic comedies). The preferences recorded by these surveys allow the system to make initial recommendations, even before users have rated many films on the website, hoping to improve performance and other problems with the Recommender System.

IV. DISCUSSION

The Recommender System is a complex system that uses various algorithms to process, analyze, and recommend results based on contextual information that suits the user's needs. These needs sometimes require quite a lot of time to choose because the data presented is enormous. The system handles various forms of data regarding users, items, context, and ratings. The analysis results from this process are the extent to which users like certain items. Besides that, we can obtain contextual and complete information to continue evaluating the performance of the algorithm being developed. To clarify it, we will explain the findings from this SLR in the form of challenges and opportunities by including the approach, Problem, Evaluation Method used, and Dataset.

A. CHALLENGES

In this SLR, there are various challenges created by Recommender System researchers. If classified, there are three types of challenges. First, before the recommendation process (dataset acquisition), which is a necessary process. Second, during the recommendation process, and third are problems that arise outside the recommendation process. These three types of challenges are interrelated and influence the results of a Recommender System.

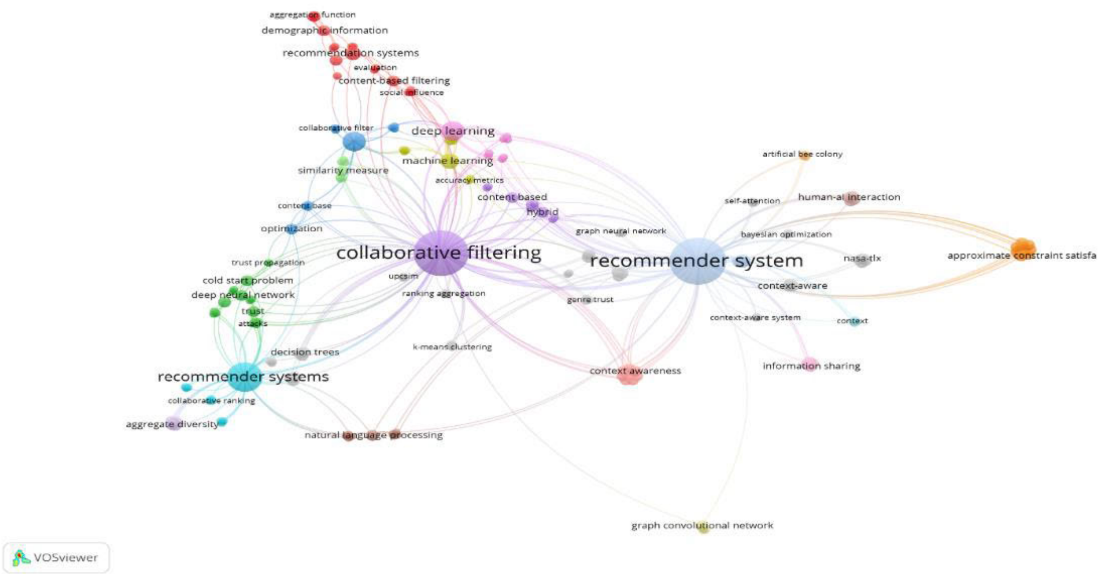

FIGURE 13. Challenges and opportunities in recommender system research.

To be clear, we explain the essential components of a Recommender System. From the SLR explanation above in subsection ''Problems'', there are significant problems in the Recommender System: Accuracy, Scalability, Running Time, Sparsity, and Cold Start. The researchers tried to solve this problem by using approaches such as Content-Based Filtering (CBF), Collaborative Filtering (CF), and Hybrid Filtering (HBF).

After the approach was determined, the researchers used suitable techniques to solve the problems above. We have provided several illustrations above in the form of techniques/methods, such as matrix factorization (MF) and Graph Neutral Network (GNN), to model contextual information on user relationships with items, as shown in Figure 7, Figure 8, and Figure 9 above. This application can be used in various Content-Based and Collaborative filtering approaches. However, the Hybrid approach can integrate several recommendation techniques. The data can be in the form of the historical behavior of user items, item content data, user profiles, interaction behavior between users, and user context information (trust relationships). MF techniques that are used and are popular include Singular Value Decomposition (SVD), SVD++, and combined with ranking aggregation, for example, WPR (Weight Point Ranking) or Bayesian Personalized Ranking (BPR), and others. Techniques vary and depend on the characteristics of each study. We summarize several research techniques on RS, such as Cosine (COS), Pearson (PCC), Gowers, a CNN deep learning (DL) model to build a multiclass, User Profile Correlation-based Similarity (UPCSim), Clustering-Based UPCSim (CB-UPCSim), K-means Eucledian, K-means cosine, and Fuzzy K-mean, Neural Attention Network for Neighborhood Augmented Knowledge Graph (NKGAT), Graph modeling Preference matching, the genre trust-based recommender system (GTD), SVD-GCN, a DNN-based recommendation model, Learning Automata (LA) technique and others.

These stages are carried out in the hope of making improvements and increasing the performance of the algorithm being designed and being able to solve problems in the Recommender System. The dataset is obtained as follows regarding the data sets from the SLR results. Movielens 100K, Movielens 1M, Movilens 10M, netflix, Yahoo Music, Film Trust, Epinions, Book Crossing, Movie tweetings, Yelp, Ciao, Amazon, and others.

B. OPPORTUNITY

This SLR research presents Recommender System research opportunities from the perspective of approaches, problems, evaluation methods, and data sets. This is done to find opportunities and potential for new research related to the Recommender System (RS). These opportunities look at trends and potential findings with quite complex problems. However, this problem can be solved from various points of view.

First, we look at the potential/opportunities of each problem to be solved. We obtained findings in accuracy, which is the main problem with this SLR. Accuracy includes whether the presentation is appropriate to the user regarding the items presented. In determining recommendations, user interest data is required, which is stated explicitly in the item ranking data or inferred by guessing user actions. Opportunities that can be taken include presenting new algorithms by combining existing and other methods. It can also be done using a model with a feasible approach. However, every model built could be better, and improvements continue to be made. Likewise, other opportunities can be implemented if there is a problem of limited data information, resulting in data gaps (data sparsity) and also causing cold starts. On the other hand, as the dataset increases, it will also impact scalability. Thus, this opportunity is fundamental for researchers in the future to improve the algorithm in the Recommender System.

Second, we present an illustration in Figure 13 below to strengthen the opportunities in SLR. This illustrative image was obtained based on the results of SLR 72 articles related to the Recommender System. We include abstracts and keywords as the primary material for searching gap research. If we look more clearly, the most dominant thing is the Collaborative Filtering and Recommender System. In contrast, the rest are colors with increasingly smaller dots. Small dots are methods that can be followed up as real opportunities, such as Graph Neurol Networks (GNN) and Hybrid opportunities, which are significant. Besides that, some problems arise in the research that need to be followed up. Starting from rating and ranking-based model approaches to memory-based models. Another exciting thing about this picture is the opportunity for modern approaches, such as demographic information approach, time-based approach, etc.

Third, in the Recommender System, the vital thing to do is evaluate the method that has been implemented. From this SLR, we found that the evaluation method applied to the research was under the characteristics of the research. We present various evaluation methods that can be considered in future research. The evaluation methods include Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Normalized Discounted Cumulative Gain (NDCG), Precision, Recall, F1, etc. (found in Table 6 Evaluation Method based on researchers). Algorithm performance using specific methods can be measured based on the loss and accuracy values from testing the model on test data, depending on the problem faced. Model training is carried out to get a model with the smallest possible loss value and the maximum possible accuracy value. Thus, this opportunity can also be integral to the Recommender System process.

V. CONCLUSION

This paper reviews the Recommender System (RS) literature, intending to comprehend the approach used, the problems faced, the evaluation methods, and related datasets used in each study. This research uses a Systematic Literature Review (SLR), which sets questions and defines them clearly and structured to continue research on hospitals. Seventy-two research works in paper form from 2019 to 2023 were reviewed, analyzed, and discussed in this review. We also illustrate several research results related to the approach used, the methods developed, and the resulting prominent work.

Besides that, we received various challenges and opportunities in exploring this recommender system. The challenge that often arises is how we produce good data, process it, and produce maximum results. This cannot be separated from the algorithms built in the form of new methods, solving problems of increasing accuracy, cold stardom on data, scalability, and so on.

Meanwhile, the opportunities that can be taken from SLR are in the form of being able to create new algorithms by combining various ranking techniques, similarity techniques, and aggregation techniques. As an illustration, we provide such using a Collaborative Filtering (memory-based) approach with the SVD (Matrix Decomposition) and WPR (Rank Aggregation) Combination techniques. Another example is Clustering-Based UPCSim (CB-UPCSim). The other opportunity that can be taken and is quite large in this SLR is in the form of a hybrid approach, namely combining Content-Based and Collaborative filtering approaches. For example we can employee two recommendation techniques based on Content-Based and matrix decomposition. All the opportunities we have explained above aims to provide readers with information about the extent of RS development and contribute to solving problems in RS, increasing accuracy, solving cold star problems, scalability, and so on.

REFERENCES

[1] J. Kumar, B. K. Patra, B. Sahoo, and K. S. Babu, "Group recommendation exploiting characteristics of user-item and collaborative rating of users," *Multimedia Tools Appl.*, Sep. 2023, doi: 10.1007/s11042-023-16799-4.

[2] Z. Zhang, B. G. Patra, A. Yaseen, J. Zhu, R. Sabharwal, K. Roberts, T. Cao, and H. Wu, "Scholarly recommendation systems: A literature survey," *Knowl. Inf. Syst.*, vol. 65, no. 11, pp. 4433–4478, Nov. 2023, doi: 10.1007/s10115-023-01901-x.

[3] H. Qin, X. Han, X. Ma, and W. Yan, "Personalized literature recommendation based on heterogeneous entity academic network," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 8, Sep. 2023, Art. no. 101649, doi: 10.1016/j.jksuci.2023.101649.

[4] A. B. Suhaim and J. Berri, "Context-aware recommender systems for social networks: Review, challenges and opportunities," *IEEE Access*, vol. 9, pp. 57440–57463, 2021, doi: 10.1109/ACCESS.2021.3072165.

[5] S. Peng, K. Sugiyama, and T. Mine, "SVD-GCN: A simplified graph convolution paradigm for recommendation," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2022, pp. 1625–1634, doi: 10.1145/3511808.3557462.

[6] T. Widiyaningtyas, M. I. Ardiansyah, and T. B. Adji, "Recommendation algorithm using SVD and weight point rank (SVD-WPR)," *Big Data Cognit. Comput.*, vol. 6, no. 4, p. 121, Oct. 2022, doi: 10.3390/bdcc6040121.

[7] C. Musto, G. Spillo, and G. Semeraro, "Harnessing distributional semantics to build context-aware justifications for recommender systems," *SSRN Electron. J.*, pp. 1–32, Jul. 2022, doi: 10.2139/ssrn.4160310.

[8] P. Kouki, J. Schaffer, J. Pujara, J. O'Donovan, and L. Getoor, "Generating and understanding personalized explanations in hybrid recommender systems," *ACM Trans. Interact. Intell. Syst.*, vol. 10, no. 4, pp. 1–40, Dec. 2020, doi: 10.1145/3365843.

[9] Z. Liu and F. Ren, "Algorithm improvement of movie recommendation system based on hybrid recommendation algorithm," *Frontiers Comput. Intell. Syst.*, vol. 3, no. 3, pp. 113–117, May 2023, doi: 10.54097/fcis.v3i3.8581.

[10] M. V. Farashah, A. Etebarian, R. Azmi, and R. E. Dastjerdi, "A hybrid recommender system based-on link prediction for movie baskets analysis," *J. Big Data*, vol. 8, no. 1, pp. 1–24, Dec. 2021, doi: 10.1186/s40537-021-00422-0.

[11] S. Sahu, R. Kumar, M. S. Pathan, J. Shafi, Y. Kumar, and M. F. Ijaz, "Movie popularity and target audience prediction using the content-based recommender system," *IEEE Access*, vol. 10, pp. 42044–42060, 2022, doi: 10.1109/ACCESS.2022.3168161.

[12] T. Anwar and V. Uma, "Comparative study of recommender system approaches and movie recommendation using collaborative filtering," *Int. J. Syst. Assurance Eng. Manage.*, vol. 12, no. 3, pp. 426–436, Jun. 2021, doi: 10.1007/s13198-021-01087-x.

[13] P. Campos, E. Pinto, and A. Torres, *Rating and Perceived Helpfulness in a Bipartite Network of Online Product Reviews*. Cham, Switzerland: Springer, 2023, doi: 10.1007/s10660-023-09725-1.

[14] X. Liu, G. Wang, and M. Z. A. Bhuiyan, "Personalised context-aware re-ranking in recommender system," *Connection Sci.*, vol. 34, no. 1, pp. 319–338, Dec. 2022, doi: 10.1080/09540091.2021.1997915.

[15] M. Mansoury, H. Abdollahpouri, M. Pechenizkiy, B. Mobasher, and R. Burke, "A graph-based approach for mitigating multi-sided exposure bias in recommender systems," *ACM Trans. Inf. Syst.*, vol. 40, no. 2, pp. 1–31, Apr. 2022, doi: 10.1145/3470948.

[16] M. D. Ekstrand, B. Carterette, and F. Diaz, "Distributionally-informed recommender system evaluation," *ACM Trans. Recommender Syst.*, pp. 1–26, Aug. 2023, doi: 10.1145/3613455.

[17] M. Jalui, T. Kadam, K. Karkera, R. Wani, and M. Pathak, "FlixTime: Group movie recommendation system," *SSRN Electron. J.*, pp. 1–15, May 2023, doi: 10.2139/ssrn.4428451.

[18] S. H K, K. P. Iyer, H. K R, and R. Pokharel, "Personalized movie recommendation system," *Int. J. Inf. Technol., Res. Appl.*, vol. 2, no. 1, pp. 1–6, Mar. 2023, doi: 10.59461/ijitra.v2i1.40.

[19] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender Systems Handbook*, 2011, pp. 145–186, doi: 10.1007/978-0-387-85820-3.

[20] R. Alabduljabbar, M. Alshareef, and N. Alshareef, "Time-aware recommender systems: A comprehensive survey and quantitative assessment of literature," *IEEE Access*, vol. 11, pp. 45586–45604, 2023, doi: 10.1109/ACCESS.2023.3274117.

[21] S. Banerjee, M. Jenamani, and D. K. Pratihar, "Properties of a projected network of a bipartite network," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, 2017, pp. 0143–0147, doi: 10.1109/ICCSP.2017.8286734.

[22] J. Pan, X. Zhang, and X. Qi, *An Improved Singular Value Decomposition Recommender Algorithm Based on User Trust Relationship*. The Netherlands: Atlantis Press, Oct. 2015, pp. 617–622, doi: 10.2991/icmii-15.2015.105.

[23] C.-S. Wang, B.-S. Chen, and J.-H. Chiang, "TDD-BPR: The topic diversity discovering on Bayesian personalized ranking for personalized recommender system," *Neurocomputing*, vol. 441, pp. 202–213, Jun. 2021, doi: 10.1016/j.neucom.2021.02.016.

[24] T. Widiyaningtyas, I. Hidayah, and T. B. Adji, "Recommendation algorithm using clustering-based UPCSim (CB-UPCSim)," *Computers*, vol. 10, no. 10, p. 123, Oct. 2021, doi: 10.3390/computers10100123.

[25] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proc. ACM Conf. Recommender Syst.*, 2016, pp. 233–240, doi: 10.1145/2959100.2959165.

[26] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empir. Method Nat. Lang. Process. (EMNLP)*, Doha, Qatar, 2014, pp. 1746–1751, doi: 10.3115/v1/d14-1181.

[27] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell. (AUAI)*. Montreal, QC, Canada: AUAI Press, 2009, pp. 452–461.

[28] J. Ha, M. Kambe, and J. Pe, *Data Mining: Concepts and Techniques*. The Netherland: Elsevier, 2011, doi: 10.1016/C2009-0-61819-5.

[29] N. Polatidis and C. K. Georgiadis, "A multi-level collaborative filtering method that improves recommendations," *Expert Syst. Appl.*, vol. 48, pp. 100–110, Apr. 2016, doi: 10.1016/j.eswa.2015.11.023.

[30] J. Feng, X. Fengs, N. Zhang, and J. Peng, "An improved collaborative filtering method based on similarity," *PLoS ONE*, vol. 13, no. 9, Sep. 2018, Art. no. e0204003, doi: 10.1371/journal.pone.0204003.

[31] B. K. Patra, R. Launonen, V. Ollikainen, and S. Nandi, "A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data," *Knowl.-Based Syst.*, vol. 82, pp. 163–177, Jul. 2015, doi: 10.1016/j.knosys.2015.03.001.

[32] H. Tahmasebi, R. Ravanmehr, and R. Mohamadrezaei, "Social movie recommender system based on deep autoencoder network using Twitter data," *Neural Comput. Appl.*, vol. 33, no. 5, pp. 1607–1623, Mar. 2021, doi: 10.1007/s00521-020-05085-1.

[33] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informat. J.*, vol. 16, no. 3, pp. 261–273, Nov. 2015, doi: 10.1016/j.eij.2015.06.005.

[34] Y. Lu, K. Nakamura, and R. Ichise, "HyperRS: hypernetwork-based recommender system for the user cold-start problem," *IEEE Access*, vol. 11, pp. 5453–5463, 2023, doi: 10.1109/ACCESS.2023.3236391.

[35] J. Xiang, G. Kang, H. Cheng, J. Liu, Y. Wen, and J. Xu, "Scalable and optimal QoS-aware manufacturing service composition via Bus. Process decomposition," *Electronics*, vol. 12, no. 4, p. 991, Feb. 2023, doi: 10.3390/electronics12040991.

[36] G. Behera and N. Nain, "The state-of-the-art and challenges on recommendation system's: Principle, techniques and evaluation strategy," *Social Netw. Comput. Sci.*, vol. 4, no. 5, p. 677, Sep. 2023, doi: 10.1007/s42979-023-02207-z.

[37] A. H. Afridi and H. A. Mengash, "NASA-TLX–based workload assessment for academic resource recommender system," *Pers. Ubiquitous Comput.*, vol. 26, no. 3, pp. 881–899, Jun. 2022, doi: 10.1007/s00779-020-01409-z.

[38] G. Musick, E. S. Gilman, W. Duan, N. J. McNeese, B. Knijnenburg, and T. O'Neill, "Knowing unknown teammates: Exploring anonymity and explanations in a teammate information-sharing recommender system," *Proc. ACM Hum.-Comput. Interact.*, vol. 7, pp. 1–34, Sep. 2023, doi: 10.1145/3610075.

[39] J. Adamczak, Y. Deldjoo, F. B. Moghaddam, P. Knees, G.-P. Leyson, and P. Monreal, "Session-based hotel recommendations dataset," *ACM Trans. Intell. Syst. Technol.*, vol. 12, no. 1, pp. 1–20, Feb. 2021, doi: 10.1145/3412379.

[40] M. Bałchanowski and U. Boryczka, "A comparative study of rank aggregation methods in recommendation systems," *Entropy*, vol. 25, no. 1, p. 132, Jan. 2023, doi: 10.3390/e25010132.

[41] K. Abbas, S. Dong, and A. Khan, "Point process based time sensitive personalised recommendation," *Proc. Comput. Sci.*, vol. 218, pp. 1791–1804, Jan. 2023, doi: 10.1016/j.procs.2023.01.157.

[42] J. A. Concha-Carrasco, M. A. Vega-Rodríguez, and C. J. Pérez, "A multi-objective artificial bee colony approach for profit-aware recommender systems," *Inf. Sci.*, vol. 625, pp. 476–488, May 2023, doi: 10.1016/j.ins.2023.01.050.

[43] G. Behera and N. Nain, "Collaborative filtering with temporal features for movie recommendation system," *Proc. Comput. Sci.*, vol. 218, pp. 1366–1373, Jan. 2023, doi: 10.1016/j.procs.2023.01.115.

[44] C. Lee, D. Han, K. Han, and M. Yi, "Improving graph-based movie recommender system using cinematic experience," *Appl. Sci.*, vol. 12, no. 3, p. 1493, Jan. 2022, doi: 10.3390/app12031493.

[45] T. Widiyaningtyas, I. Hidayah, and T. B. Adji, "User profile correlation-based similarity (UPCSim) algorithm in movie recommendation system," *J. Big Data*, vol. 8, no. 1, pp. 1–21, Dec. 2021, doi: 10.1186/s40537-021-00425-x.

[46] T. Widiyaningtyas, I. Hidayah, and T. Bharata Adji, "Recommendation algorithm using weighted user behavior score-based similarity," in *Proc. 2nd Int. Conf. Inf. Technol. Educ. (ICIT&E)*, Jan. 2022, pp. 87–91, doi: 10.1109/ICITE54466.2022.9759851.

[47] G. Huang, X. Zhu, S. H. Wasti, and Y. Jiang, *Multi-Knowledge Resources-Based Semantic Similarity Models With Application for Movie Recommender System*. Amsterdam, The Netherlands: Springer, 2023, doi: 10.1007/s10462-023-10573-6.

[48] R. Kirubahari and S. M. J. Amali, "An improved restricted Boltzmann machine using Bayesian optimization for recommender systems," *Evolving Syst.*, no. 0123456789, pp. 1–13, Sep. 2023, doi: 10.1007/s12530-023-09520-1.

[49] F. Rezaimehr and C. Dadkhah, *T&TRS: Robust Collaborative Filtering Recommender Systems Against Attacks*. Cham, Switzerland: Springer, 2023, doi: 10.1007/s11042-023-16641-x.

[50] Z. Liang, Z. Yang, and J. Cheng, "Weight normalization optimization movie recommendation algorithm based on three-way neural interaction networks," *Complex Intell. Syst.*, vol. 9, no. 6, pp. 6241–6254, Dec. 2023, doi: 10.1007/s40747-023-01066-8.

[51] R. Fiagbe. (May 2023). *Movie Recommender System Using Matrix Factorization*. [Online]. Available: http://library.ucf.edu

[52] S. Bhaskaran and R. Marappan, "Enhanced personalized recommendation system for machine learning public datasets: Generalized modeling, simulation, significant results and analysis," *Int. J. Inf. Technol.*, vol. 15, no. 3, pp. 1583–1595, Mar. 2023, doi: 10.1007/s41870-023-01165-2.

[53] L. Yang and X. Niu, "A genre trust model for defending shilling attacks in recommender systems," *Complex Intell. Syst.*, vol. 9, no. 3, pp. 2929–2942, Jun. 2023, doi: 10.1007/s40747-021-00357-2.

[54] Q. Wang, H. Yin, T. Chen, J. Yu, A. Zhou, and X. Zhang, "Fast-adapting and privacy-preserving federated recommender system," *VLDB J.*, vol. 31, no. 5, pp. 877–896, Sep. 2022, doi: 10.1007/s00778-021-00700-6.

[55] T. Chatzidimitris, D. Gavalas, V. Kasapakis, C. Konstantopoulos, D. Kypriadis, G. Pantziou, and C. Zaroliagis, "A location history-aware recommender system for smart retail environments," *Pers. Ubiquitous Comput.*, vol. 24, no. 5, pp. 683–694, Oct. 2020, doi: 10.1007/s00779-020-01374-7.

[56] C. Wang, H. Zhu, C. Zhu, C. Qin, and H. Xiong, "Setrank: A setwise Bayesian approach for collaborative ranking from implicit feedback," in *Proc. AAAI 34th AAAI Conf. Artif. Intell.*, 2020, pp. 6127–6136, doi: 10.1609/aaai.v34i04.6077.

[57] Z. Chen et al., "Practitioners versus users: A value-sensitive evaluation of current industrial recommender system design," in *Proc. ACM Hum.-Comput. Interact.*, vol. 6, 2022, pp. 1–32, doi: 10.1145/3555646.

[58] T. Nguyen and J. Uhlmann, "Tensor completion with provable consistency and fairness guarantees for recommender systems," *ACM Trans. Recommender Syst.*, vol. 1, no. 3, pp. 1–26, Sep. 2023, doi: 10.1145/3604649.

[59] T. Markchom, H. Liang, and J. Ferryman, "Explainable meta-path based recommender systems," *ACM Trans. Recommender Syst.*, pp. 1–29, Sep. 2023, doi: 10.1145/3625828.

[60] T. Zhang and V. S. Iyengar, "Recommender systems using linear classifiers," *J. Mach. Learn. Res.*, vol. 2, no. 3, pp. 313–334, 2002, doi: 10.1162/153244302760200641.

[61] E. Ceh-Varela, H. Cao, and H. W. Lauw, "Performance evaluation of aggregation-based group recommender systems for ephemeral groups," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 6, pp. 1–26, Dec. 2022, doi: 10.1145/3542804.

[62] N.-T. Tran and H. W. Lauw, "Memory network-based interpreter of user preferences in content-aware recommender systems," *ACM Trans. Intell. Syst. Technol.*, vol. 14, no. 6, pp. 1–28, Dec. 2023, doi: 10.1145/3625239.

[63] Q. Shen et al., "Multi-interest multi-round conversational recommendation system with fuzzy feedback based user simulator," *ACM Trans. Recommender Syst.*, pp. 1–28, Aug. 2023, doi: 10.1145/3616379.

[64] H. A. Yazdi, S. J. S. M. Chabok, and M. Kheirabadi, "Dynamic educational recommender system based on improved recurrent neural networks using attention technique," *Appl. Artif. Intell.*, vol. 36, no. 1, Dec. 2022, Art. no. 2005298, doi: 10.1080/08839514.2021.2005298.

[65] I. Y. Choi, Y. U. Ryu, and J. K. Kim, "A recommender system based on personal constraints for smart tourism city," *Asia Pacific J. Tourism Res.*, vol. 26, no. 4, pp. 440–453, Apr. 2021, doi: 10.1080/10941665.2019.1592765.

[66] J. Redelinghuys and S. Er, "SoMLier: A South African wine recommender system," *J. Wine Res.*, vol. 34, no. 1, pp. 54–80, Jan. 2023, doi: 10.1080/09571264.2023.2184333.

[67] S. Taghipour, J. A. Torkestani, and S. Nazari, "A learning automata-based approach to improve the scalability of clustering-based recommender systems," *Cybern. Syst.*, pp. 1–32, 2022, doi: 10.1080/01969722.2022.2145650.

[68] Z. Su, Z. Lin, J. Ai, and H. Li, "Rating prediction in recommender systems based on user behavior probability and complex network modeling," *IEEE Access*, vol. 9, pp. 30739–30749, 2021, doi: 10.1109/ACCESS.2021.3060016.

[69] S. Airen and J. Agrawal, "Movie recommender system using parameter tuning of user and movie neighbourhood via co-clustering," *Proc. Comput. Sci.*, vol. 218, pp. 1176–1183, Jan. 2023, doi: 10.1016/j.procs.2023.01.096.

[70] M. Jallouli, S. Lajmi, and I. Amous, "REMOVE: REcommendation model based on sOcio-enVironmental contExt," *Multimedia Tools Appl.*, vol. 82, no. 16, pp. 24803–24840, Jul. 2023, doi: 10.1007/s11042-022-14239-3.

[71] Q. Wang, H. Cui, J. Zhang, Y. Du, Y. Zhou, and X. Lu, "Neighbor-augmented knowledge graph attention network for recommendation," *Neural Process. Lett.*, vol. 55, no. 6, pp. 8237–8253, Dec. 2023, doi: 10.1007/s11063-023-11310-4.

[72] W. Zhou and W. Han, "Personalized recommendation via user preference matching," *Inf. Process. Manage.*, vol. 56, no. 3, pp. 955–968, May 2019, doi: 10.1016/j.ipm.2019.02.002.

[73] S. Mohamadi, V. Aghazarian, and A. Hedayati, "An effective profile expansion technique based on movie genres and user demographic information to improve movie recommendation systems," *Multimedia Tools Appl.*, vol. 82, no. 25, pp. 38275–38296, Oct. 2023, doi: 10.1007/s11042-023-15141-2.

[74] D. Jin, L. Wang, H. Zhang, Y. Zheng, W. Ding, F. Xia, and S. Pan, "A survey on fairness-aware recommender systems," *Inf. Fusion*, vol. 100, Dec. 2023, Art. no. 101906, doi: 10.1016/j.inffus.2023.101906.

[75] M. R. Zarei, M. R. Moosavi, and M. Elahi, "Adaptive trust-aware collaborative filtering for cold start recommendation," *Behaviormetrika*, vol. 50, no. 2, pp. 541–562, Jul. 2023, doi: 10.1007/s41237-022-00161-3.

[76] G. Jain, T. Mahara, and S. C. Sharma, "Performance evaluation of time-based recommendation system in collaborative filtering technique," *Proc. Comput. Sci.*, vol. 218, pp. 1834–1844, Jan. 2023, doi: 10.1016/j.procs.2023.01.161.

[77] J. Li, Y. Wang, and J. McAuley, "Time interval aware self-attention for sequential recommendation," in *Proc. 13th Int. Conf. Web Search Data Mining (WSDM)*, Jan. 2020, pp. 322–330, doi: 10.1145/3336191.3371786.

[78] S. S. Choudhury, S. N. Mohanty, and A. K. Jagadev, "Multimodal trust based recommender system with machine learning approaches for movie recommendation," *Int. J. Inf. Technol.*, vol. 13, no. 2, pp. 475–482, Apr. 2021, doi: 10.1007/s41870-020-00553-2.

[79] O. Jeunen and B. Goethals, "Pessimistic decision-making for recommender systems," *ACM Trans. Recommender Syst.*, vol. 1, no. 1, pp. 1–27, Mar. 2023, doi: 10.1145/3568029.

[80] A. Elbeltagi, M. Kumar, N. L. Kushwaha, C. B. Pande, P. Ditthakit, D. K. Vishwakarma, and A. Subeesh, "Drought indicator analysis and forecasting using data driven models: Case study in jaisalmer, India," *Stochastic Environ. Res. Risk Assessment*, vol. 37, no. 1, pp. 113–131, Jan. 2023, doi: 10.1007/s00477-022-02277-0.

[81] S. M. Robeson and C. J. Willmott, "Decomposition of the mean absolute error (MAE) into systematic and unsystematic components," *PLoS ONE*, vol. 18, no. 2, Feb. 2023, Art. no. e0279774, doi: 10.1371/journal.pone.0279774.

[82] B. Sharma, A. Hashmi, C. Gupta, O. I. Khalaf, G. M. Abdulsahib, and M. M. Itani, "Hybrid sparrow clustered (HSC) algorithm for top-N recommendation system," *Symmetry*, vol. 14, no. 4, p. 793, Apr. 2022, doi: 10.3390/sym14040793.

[83] T. Widiyaningtyas, I. Hidayah, and T. B. Adji, "Comparing user rating-based similarity to user behavior-based similarity in movie recommendation systems," in *Proc. Int. Conf. Electr. Inf. Technol. (IEIT)*, Sep. 2022, pp. 52–58, doi: 10.1109/IEIT56384.2022.9967884.

[84] T. Widiyaningtyas, I. Hidayah, and T. B. Adji, "Performance of user behavior-based similarity on top-N recommendation," in *Proc. 7th Int. Conf. Electr., Electron. Inf. Eng. (ICEEIE)*, Oct. 2021, pp. 559–563, doi: 10.1109/ICEEIE52663.2021.9616734.

[85] J. Latrech, Z. Kodia, and N. Ben Azzouna, "CoDFi-DL: A hybrid recommender system combining enhanced collaborative and demographic filtering based on deep learning," *J. Supercomput.*, vol. 80, no. 1, pp. 1160–1182, Jan. 2024, doi: 10.1007/s11227-023-05519-2.

[86] F. Horasan, A. H. Yurttakal, and S. Gündüz, "A novel model based collaborative filtering recommender system via truncated ULV decomposition," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 8, Sep. 2023, Art. no. 101724, doi: 10.1016/j.jksuci.2023.101724.

ILHAM SAIFUDIN was born in Bondowoso Regency, East Java, Indonesia, in 1989. He received the M.Sc. degree in mathematics from the University of Jember, Indonesia, in 2015. He is currently pursuing the Ph.D. degree in electrical and information engineering with the State University of Malang, Indonesia.

Since 2017, he has been a Lecturer with the Department of Informatics Engineering, Muhammadiyah University of Jember, Indonesia. He is the author of more than 20 articles in scientific journal publications. His research interests include theory and applications in graph theory, design geometry, and recommender systems.

TRIYANNA WIDIYANINGTYAS (Member, IEEE) received the Ph.D. degree in electrical engineering from Gajah Mada University (UGM), Indonesia, in 2022.

She has been an Associate Professor with the Department of Electrical Engineering and Informatics, Universitas Negeri Malang, Indonesia, since 2023. She is the author of three books, more than 70 articles, and more than ten discoveries. Her research interests include databases, data mining, and recommender systems. She often researches in developing new algorithms for recommender systems.