

Recommendation System Based on a Compact Hybrid User Model Using Fuzzy Logic Algorithms

Nina Khairova¹, Nataliia Sharonova¹, Dmytro Sytnikov², Mykyta Hrebeniuk², Polina Sytnikova²

¹National Technical University "Kharkiv Polytechnic Institute", Kyrpychova str. 2, Kharkiv, 61002, Ukraine

²Kharkiv National University of Radio Electronics, Nauky Ave. 14, Kharkiv, City, 61166, Ukraine

Abstract

The paper presents algorithm designed to address the challenges of traditional collaborative filtering methods by integrating a compact hybrid user model. This model incorporates hybrid features, demographic information, and fuzzy logic principles to improve recommendation accuracy. A key contribution of this work is the development of an innovative approach for calculating user similarity using fuzzy logic algorithms. By considering fuzzy concepts, the proposed approach effectively captures the inherent uncertainty and imprecision in user preferences, leading to more nuanced and accurate recommendations. Experimental evaluations conducted on the widely used MovieLens dataset provide insights into the performance of the proposed algorithm compared to traditional collaborative filtering techniques such as Pearson correlation and cosine similarity. The dataset, which contains both user ratings and demographic details, serves as a comprehensive testbed for assessing recommendation systems. The results of the experiments demonstrate the superiority of the proposed approach in capturing user similarities and enhancing recommendation accuracy. This paper contributes to the ongoing progress in recommendation systems by proposing a solution that addresses the challenges associated with traditional collaborative filtering methods. Through the integration of hybrid user models, demographic data, and fuzzy logic principles, the proposed algorithm offers a promising approach for enhancing recommendation accuracy across diverse application domains.

Keywords

Recommendation system, fuzzy logic, hybrid feature, compact hybrid user model, fuzzy user model, similarity, fuzzy distance

1. Introduction

With the proliferation of data-driven technologies and the increasing volume of information across various domains, the demand for efficient recommendation systems has grown substantially. These systems play a vital role in assisting users in discovering relevant items or content based on their preferences and behavior. Collaborative filtering, a widely used approach in recommendation systems, leverages user interactions and similarities to generate personalized recommendations. However, traditional collaborative filtering methods often face challenges related to scalability and accuracy, particularly when dealing with sparse or incomplete data.

To address these challenges, hybrid recommendation systems have emerged as a promising solution by integrating multiple recommendation techniques, such as collaborative filtering and demographic filtering. By combining collaborative data with demographic information, hybrid systems aim to enhance recommendation accuracy and mitigate sensitivity to data sparsity. Additionally, the incorporation of fuzzy logic allows for a more nuanced representation of user preferences, accommodating the inherently fuzzy nature of human decision-making.

COLINS-2024: 8th International Conference on Computational Linguistics and Intelligent Systems, April 12–13, 2024, Lviv, Ukraine

✉ khairova.nina@gmail.com (N. Khairova); nvsharonova@ukr.net (N. Sharonova); dmytro.sytnikov@nure.ua (D. Sytnikov); mykyta.hrebeniuk@nure.ua (M. Hrebeniuk); polina.sytnikova@nure.ua (P. Sytnikova)

ORCID: 0000-0002-9826-0286 (N. Khairova); 0000-0002-8161-552X (N. Sharonova); 0000-0003-1240-7900 (D. Sytnikov); 0009-0008-0989-7957 (M. Hrebeniuk); 0000-0002-6688-4641 (P. Sytnikova)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

This paper explores the development and evaluation of a hybrid recommendation system that integrates collaborative filtering with demographic information and fuzzy logic. The system utilizes a compact user model, incorporating genre interest indicators (GII) derived from user ratings and demographic attributes such as age, gender, and profession. By hybridizing at both the feature and model levels, the system aims to improve recommendation accuracy while maintaining scalability.

Experimental evaluations are conducted using the MovieLens dataset, comprising user ratings for movies along with demographic information. The performance of the proposed hybrid recommendation system is compared against traditional collaborative filtering methods, including Pearson correlation and cosine similarity. The experiments aim to assess the effectiveness of the fuzzy distance function in capturing user similarities and improving recommendation accuracy.

Through this research, insights into the efficacy of hybrid recommendation systems and the impact of incorporating fuzzy logic are gained. The findings contribute to advancing the understanding of recommendation system design and optimization, with implications for enhancing user experiences across various applications and domains.

2. Related works

The landscape of recommender systems has been shaped by extensive research efforts over the past decade, with a focus on improving recommendation accuracy and addressing various challenges [1]. Collaborative filtering (CF) has remained a cornerstone in recommendation system research, owing to its ability to provide personalized recommendations based on user-item interactions [2].

Hybrid recommendation systems, which integrate multiple recommendation techniques, have emerged as a promising approach to overcome the limitations of individual methods. [3,4] explores the landscape of hybrid recommender systems, offering insights into different hybridization techniques and identifying trends in hybrid recommender system research. Paper [5] describes a hybrid music recommendation system that combine collaborative filtering with content-based filtering techniques. By integrating user preferences with item features such as genre and artist, these systems aim to provide more diverse and personalized music recommendations for automated playlist continuation. Another area of exploration involves weighting strategies [6] for recommender systems that cluster items based on genres. By assigning weights to items within each cluster, these systems aim to enhance recommendation accuracy by giving more importance to relevant genres in the user's preferences. [7] introduces the concept of genre interest measure (GII), which is a hybrid feature that combines user ratings and movie genres and represents user preferences at the model level.

Also, an important part of the recommendation system is the calculation of similarity. [8] has delved into evaluating different similarity measures used in collaborative filtering-based recommender systems. Traditional measures such as Pearson correlation and cosine similarity, as well as more advanced techniques like adjusted cosine similarity and Jaccard coefficient, are reviewed and compared to evaluate their performance in recommendation tasks through experimental comparisons.

Additionally, the integration of demographic information into recommendation systems has shown promise. Addressing the cold-start problem in recommender systems, another study leverages user demographic attributes to provide personalized recommendations to new users with limited interaction history. By incorporating demographic information such as age, gender, and location into recommendation algorithms, the system aims to mitigate the cold-start problem and offer relevant recommendations [9,10]. Specifically tailored for movie recommendations, a demographic collaborative recommender system leverages demographic information such as age and gender to enhance collaborative filtering algorithms, offering accurate and personalized movie recommendations to users [10].

The incorporation of fuzzy logic principles into recommendation algorithms has garnered attention for its capacity to model the inherent uncertainty in user preferences. A paper [11] provides an overview of fuzzy logic techniques applied in recommender systems, discussing how fuzzy logic can address uncertainty and imprecision in user preferences and item attributes. Various fuzzy logic-based recommendation approaches are reviewed, along with their effectiveness in improving recommendation accuracy. By integrating fuzzy logic with CF techniques, as demonstrated in [12,13], recommendation accuracy can be improved, highlighting the importance of incorporating fuzzy concepts in user modeling. [14] addresses this problem by using fuzzy C-means method and comparing its performance against other clustering techniques used in user-based Collaborative filtering recommendation systems. By incorporating fuzzy logic principles, the system offers a more nuanced understanding of user preferences, leading to more accurate recommendations.

In summary, the integration of collaborative filtering, hybrid recommendation techniques, and fuzzy logic principles has led to significant advancements in recommendation system effectiveness.

3. Methods and materials

Formally, we have M users, $U = \{u_1 \dots, u_M\}$, explicit or implicit ratings of K items, $S = \{s_1 \dots, s_K\}$, such as news, web pages, books, games, or movies. The spaces S and U are large and can be very large in some cases. Each user u_i , where $i = 1, \dots, M$ rated a subset of items S_i . The rating of user u_c for item s_k , where $k = 1, \dots, K$ is denoted as $r_{c,k}$. All available ratings are collected in an $M \times K$ user-item matrix denoted as R . The architecture of different recommendation systems can be centralized or distributed. In this work, we assume a centralized architecture where the recommendation system is in one specific place.

During the development of a recommendation system, the following five phases can be identified:

1. Data collection
2. User model formation
3. Similarity computation
4. Neighbor selection
5. Predictions and recommendations

3.1 Data collection

The recommendation system should have as much information as possible about users to provide them with satisfactory results from the very beginning. This information includes user interests, origin, habits, personal data, and other details. Typically, three types of data can be collected from users in addition to product descriptions, namely demographic information during registration, explicit ratings (expressing users' opinions on items) for a subset of available items, and implicit data from user behavior on the service. Implicit ratings relate to the interpretation of user behavior or choice for assigning a rating or preference based on viewing data, purchase history, or other types of information access models. Additionally, the recommendation system should have access to a database of items being evaluated (in our case, movies).

3.2 User model formation

Memory-based collaborative filtering is more accurate, but its scalability compared to model-based recommendation systems is poor. In addition, actual user preferences may not always be captured solely through ratings, and therefore, some item content descriptions are needed. This can be achieved if we build a hybrid user model [3] that integrates user ratings with some item content descriptions.

The user-based memory-based collaborative filtering model consists of a vector of elements whose ratings increase as the user interacts with the system over time. This huge amount of data requires a very large space and extremely long processing time. During a query search across the entire database to find the best set of neighbors is computationally expensive. On the other hand, model-based collaborative filtering receives a model from a group of users that may be far from the actual preferences of the users. While memory-based collaborative filtering is simple, it provides recommendations with high accuracy and allows for easy addition of new data, but it is expensive in terms of computation as the size of the input data set increases. Ultimately, the user may leave the website until processing is complete. On the other hand, applying only model-based collaborative filtering to such sparse data, although reducing the cost of online processing, often comes at the price of recommendation accuracy. However, one of the common threats in current recommendation system research is the need to combine recommendation methods to mitigate sparsity and scalability problems. But most common hybridization methods create two separate models and implement an online process for each filtering technique separately. Finally, some merging is used to obtain the result. What if we build a user model according to a certain filtering technique, and then apply another filtering method to the created model? Thus, only one online filtering process (Collaborative filtering) should be used, while another filtering method (Content-based filtering) is used to densify the data. To achieve this, the utilization of hybrid features is proposed.

In our methodology, we incorporate the concept of Genre Interest Indicator (GII) [6] to enhance the user model formation process. The GII is a measure of a user's interest in specific genres of items, such as movies. It is calculated based on explicit ratings provided by users for items belonging to different genres. This approach allows us to capture nuanced user preferences beyond simple ratings, enabling the recommendation system to better understand user tastes.

To implement the GII, we utilize a hybrid approach that combines collaborative filtering with demographic data. Specifically, we leverage explicit user ratings to link users to genres, while also incorporating demographic information such as age, gender, and profession. This hybrid user model provides a more accurate representation of user preferences by considering both explicit ratings and demographic factors.

3.2.1 Combining collaborative and demographic data

The compact user model described above uses genre interest indicator (GII) to build a model by linking explicit ratings to genres. However, the assertion that two people are similar is based not only on whether they have similar thoughts on a particular topic but also on other factors, such as their background and personal data. In many cases, the ratings claimed by some users are not sufficient to describe them adequately. Therefore, a hybrid user model with age, gender, and professions as demographic information, in addition to GII, may be a good choice for creating more accurate and individual recommendations.

Combining features of collaborative and demographic filtering allows for considering explicit ratings without relying solely on them, thus reducing the sensitivity of collaborative filtering to the number of ratings [8]. Conversely, it enables having demographic information about users that would otherwise be unavailable. Moreover, most current hybrid recommendation systems are weighted systems, where the online process is realized for each filter separately and then some merging is used to obtain the final result. In this work, we will attempt to introduce hybridization at two different levels, namely at the model level and at the approach level. Figure 1 illustrates the hybrid user model that we introduce to obtain hybrid collaborative/demographic filtering.

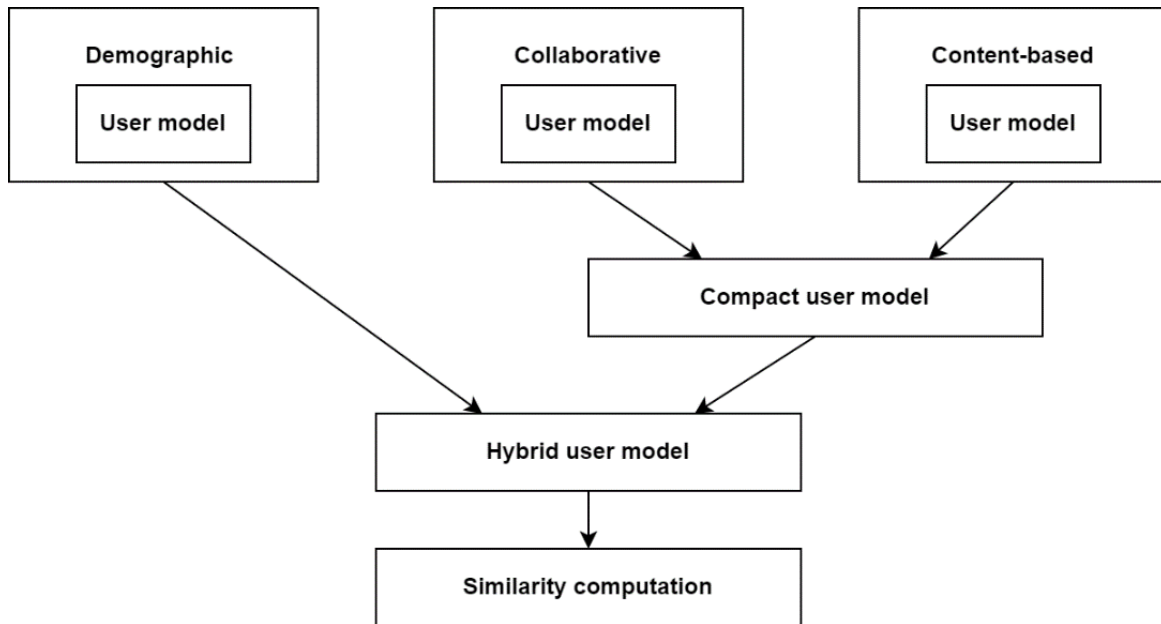


Figure 1: Hybrid user model structure

Accordingly, the hybrid user model consists of age, gender, and profession as demographic information, and GII for genres, as shown in Table 1.

Table 1
An example of a hybrid user model

No	Gender	Age	Profession	Degree	Drama	...	Triller
1	Male	24	Programmer	Master's	$GII_{u_i}(G_1)$...	$GII_{u_i}(G_N)$
...

3.2.2 Fuzzy user model

Fuzzy sets were introduced as a generalization of classical crisp sets in order to deal with fuzzy concepts such as "young," "rich," "tall," etc. Instead of the rigid membership of elements in a crisp set (1 if an element belongs to the set and 0 otherwise), a fuzzy set allows elements to have a partial degree of membership, i.e., any value in the interval $[0,1]$. In the theory of fuzzy sets, a fuzzy subset A of the universe of discourse U is described by a membership function $\mu_A(x): x \in U \rightarrow [0,1]$, which represents the degree of membership of x in the set A . Fuzzy logic refers to all theories and technologies that use fuzzy sets. For recommendation systems, most user preferences are fuzzy, so fuzzy logic is an appropriate tool for representing these preferences. In these methods, each object is represented by a set of primitive propositions, whose truth is determined in the object space by a value in the interval $[0,1]$. For example, a proposition could be "This movie is a comedy." The associated value with this proposition indicates the degree to which this movie is a comedy.

The crisp description of age and GII in the hybrid user model (Table 1) does not reflect the real case of human decisions. For example, the distance between two users aged 15 and 19 is 4, while both users belong to the same age group, namely teenagers. These fuzzy characteristics need to be taken into account when comparing users. Below we will discuss how to implement the hybrid user model and introduce a fuzzy distance function for finding the nearest neighbors.

The fuzzy user model will help create a set of neighbors as close as possible to the active user. However, to build a fuzzy model, it is first necessary to label the features of the user model. First of all, age is divided into three fuzzy sets: young, adult, and old (Figure 2), with the following membership functions:

$$A_{Young}(a) = \begin{cases} 1 & a \leq 20 \\ \left(\frac{35-a}{15}\right) & 20 < a \leq 35 \\ 0 & a > 35 \end{cases} \quad (1)$$

$$A_{Adult}(a) = \begin{cases} 0 & a \leq 20, a > 60 \\ \left(\frac{a-20}{15}\right) & 20 < a \leq 35 \\ 1 & 35 < a \leq 45 \\ \left(\frac{60-a}{15}\right) & 45 < a \leq 60 \end{cases} \quad (2)$$

$$A_{Old}(a) = \begin{cases} 0 & a \leq 45 \\ \left(\frac{a-45}{15}\right) & 45 < a \leq 60 \\ 1 & a > 60 \end{cases} \quad (3)$$

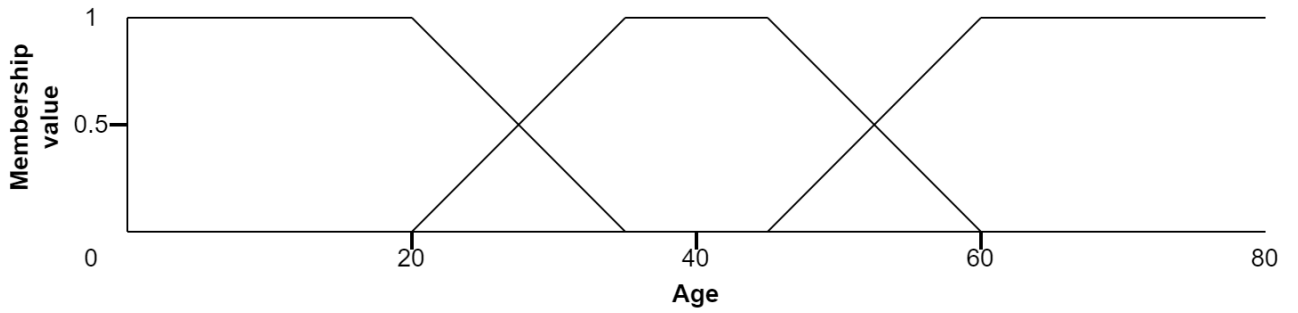


Figure 2: Membership function for the age feature

The values of gender and profession are considered as fuzzy points with a membership value of one. Finally, GII is divided into six fuzzy sets, very bad (VB), bad (B), average (AV), good (G), very good (VG), and excellent (E) with the following membership functions (Figure 3):

$$B_{VB}(a) = \begin{cases} 1 - a & a \leq 1 \\ 0 & a > 1 \end{cases} \quad (4)$$

$$B_{A(i)}(a) = \begin{cases} 0 & a \leq i - 2, a > i \\ a - i + 2 & i - 2 < a \leq i - 1 \\ i - a & i - 1 < a \leq i \end{cases} \quad i = 2,3,4,5 \quad (5)$$

Here, $A(i) = B, AV, G, VG$ for $i = 2,3,4,5$ respectively.

$$B_E(a) = \begin{cases} 0 & a \leq 4 \\ a - 4 & 4 < a \leq 5 \end{cases} \quad (6)$$

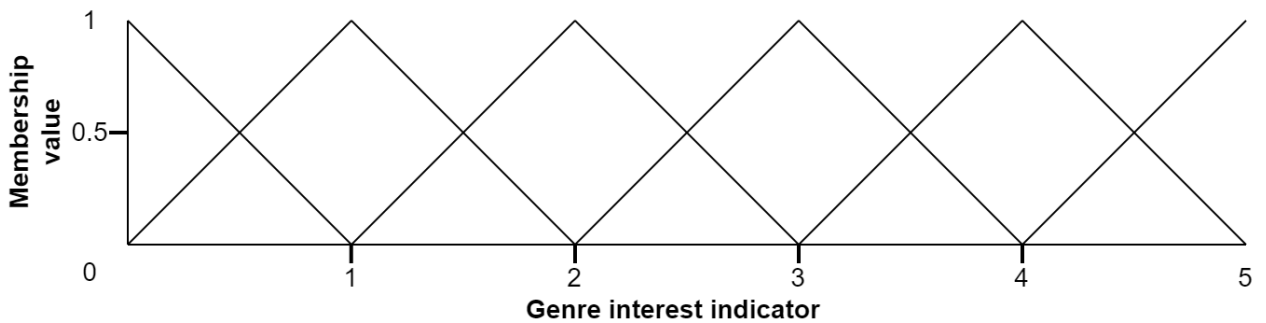


Figure 3: Membership function for the GII feature

3.3 Similarity computation

After building a user model, a recommendation system compares the active user to the available database according to the corresponding similarity function. Based on the calculated similarity values, a connection is established between the active user and other users, allowing the recommendation system to form a set of neighbors for the active user. The choice of similarity function depends on the program and is based on the nature of the user model's features. Some similarity function modifiers have been introduced to refine or enhance the recommendation system's ability to find close neighbors. It should be noted that similarity calculations for collaborative filtering can be performed between items instead of users. This work only discusses user-based methods of similarity (user-to-user similarity), as it is the most popular.

The similarity between two users is a measure of how similar they are to each other. Formally, the similarity function $sim(\mathbf{u}_x, \mathbf{u}_y)$ for a set of users U is a function with a non-negative value: $sim: U \times U \rightarrow R^+ + \{0\}$. Here, we distinguish between u_x and \mathbf{u}_x based on their context. When we use u_x , we refer to user-x, while \mathbf{u}_x represents the feature vector for the user-x model. The similarity function may have some of the following properties:

- (P1) Identity: $\forall u_x \in U, sim(\mathbf{u}_x, \mathbf{u}_x) > 0$
- (P2) Positivity: $\forall u_x (\neq u_y) \in U, sim(\mathbf{u}_x, \mathbf{u}_y) \geq 0$
- (P3) Symmetry: $\forall u_x, u_y \in U, sim(\mathbf{u}_x, \mathbf{u}_y) = sim(\mathbf{u}_y, \mathbf{u}_x)$

Any function that satisfies (P1) is a similarity function. Although symmetry is a convenient property, it is not satisfied in all programs. Non-negativity is not satisfied for two standard examples: correlation coefficients and scalar products. Different similarity functions $sim(\mathbf{u}_x, \mathbf{u}_y)$ were used in the study of collaborative filtering between users u_x and u_y . The most popular similarity function for memory-based collaborative filtering is the Pearson correlation coefficient [7], where the similarity between two users is based only on their common ratings S_{xy} . The Pearson correlation coefficient:

$$corr(\mathbf{u}_x, \mathbf{u}_y) = \frac{\sum_{S_k \in S_{xy}} (r_{x,k} - m_x)(r_{y,k} - m_y)}{\sqrt{\sum_{S_k \in S_{xy}} (r_{x,k} - m_x)^2 \sum_{S_k \in S_{xy}} (r_{y,k} - m_y)^2}} \quad (7)$$

Another similarity function is the cosine similarity function [7], which considers two users as two vectors in an $|S_{xy}|$ dimensional space.

$$cosine(\mathbf{u}_x, \mathbf{u}_y) = \frac{\sum_{S_k \in S_{xy}} r_{x,k} \times r_{y,k}}{\sqrt{\sum_{S_k \in S_{xy}} (r_{x,k})^2 \sum_{S_k \in S_{xy}} (r_{y,k})^2}} \quad (8)$$

On the other hand, dissimilarity is the opposite of similarity and is related to the concept of distance, where two terms are used interchangeably: small distances mean small differences, and large distances mean large differences. Formally, the distance function $dis(\mathbf{u}_x, \mathbf{u}_y)$ for a set of users U is a function $dis: U \times U \rightarrow R^+ + \{0\}$. The dis function may have some of the following properties:

- (P1) Identity or reflexivity: $\forall u_x \in U, dis(\mathbf{u}_x, \mathbf{u}_x) = 0$
- (P2) Positivity: $\forall u_x (\neq u_y) \in U, dis(\mathbf{u}_x, \mathbf{u}_y) > 0$
- (P3) Symmetry: $\forall u_x, u_y \in U, dis(\mathbf{u}_x, \mathbf{u}_y) = dis(\mathbf{u}_y, \mathbf{u}_x)$
- (P4) Uniqueness or definiteness: $dis(\mathbf{u}_x, \mathbf{u}_y) = 0 \Rightarrow u_x = u_y$
- (P5) Triangle inequality: $\forall u_x, u_y, u_z \in U, dis(\mathbf{u}_x, \mathbf{u}_z) \leq dis(\mathbf{u}_x, \mathbf{u}_y) + dis(\mathbf{u}_y, \mathbf{u}_z)$

Generally, identity and positivity are crucial for determining the correct distance function.

Obviously, formulas (1) and (2) are not suitable if the model includes diverse features because these formulas consider only the elements of the joint assessment of both users. The Euclidean distance function provides another way of computing differences for recommendation systems, which considers numerical peculiarities.

$$dis(\mathbf{u}_x, \mathbf{u}_y) = \sqrt{\sum_{j=1}^n (x_j - y_j)^2} \quad (9)$$

, where x_j — is the j -th feature of \mathbf{u}_x , and n — is the number of features.

3.3.1 Fuzzy distance function

Using a fuzzy user model has many advantages, but how can we compare two user models that have many fuzzy features? In general, each function has many fuzzy sets. Actually, the choice of distance function is an important issue for the system and depends largely on the problem itself. For the hybrid user model in Figure 1, a vector with N features represents the user, and therefore, for each function, a local fuzzy distance should be found. Therefore, for each pair of users, we have N local fuzzy distances. The global fuzzy distance could be obtained by two methods. The first method uses the fuzzy IF-THEN rule: IF(x_1 is close to y_1) and (x_2 is close to y_2) ... and (x_N is close to y_N) THEN (\mathbf{u}_x is similar to \mathbf{u}_y). In this case, the global fuzzy distance is defined as:

$$fdis(\mathbf{u}_x, \mathbf{u}_y) = \min\{fdis(x_1, y_1), fdis(x_2, y_2), \dots, fdis(x_N, y_N)\} \quad (10)$$

The second method considers each local fuzzy distance as an opinion. The global fuzzy distance is the global opinion of all. An aggregation operator is needed for this in fuzzy logic. The aggregation operator can be the average of N local fuzzy distances.

$$fdis(\mathbf{u}_x, \mathbf{u}_y) = \frac{\sum_{i=1}^N fdis(x_i, y_i)}{N} \quad (11)$$

Formula (10) works poorly for the hybrid user model because it considers only the feature with the minimum distance and ignores other features. According to fuzzy set and concept distances, we need a local fuzzy distance metric, $fdis(x_i, y_i)$, which satisfies the following conditions:

- A. Zero value for the same feature values.
- B. Zero value for different feature values in the same fuzzy set with the same membership values.
- C. Minimized distance between any two feature values that belong to the same fuzzy set and have close membership values.
- D. Maximized distance between any two feature values that belong to two different fuzzy sets.

Condition (A) is a fundamental requirement for any distance function. To clarify condition (B), let's assume that we have two users who are 40 and 35 years old, respectively. Both users have a membership value of 1 for the "adult" category (Figure 2). The distance between them is 5, but they are similar users in terms of fuzzy sets. To make the distance between two users zero, we need another term that gives zero value for this and similar cases. What really makes these two users similar is their equal membership values in one fuzzy set. We then define a corresponding fuzzy distance function that satisfies all four above-mentioned conditions.

Let \mathbf{a} and \mathbf{b} be the membership vectors corresponding to two crisp values a and b for a given feature with l fuzzy sets. The fuzzy distance between a and b is defined as

$$fdis(a, b) = dis(\mathbf{a}, \mathbf{b}) \times dif(a, b) \quad (12)$$

where $dif(a, b)$ is simply the difference operator, and \mathbf{a} and \mathbf{b} are vectors of size l , and $dis(\mathbf{a}, \mathbf{b})$ is any vector metric distance.

In this work, the Euclidean distance is used to calculate $dis(\mathbf{a}, \mathbf{b})$:

$$dis(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{j=1}^l (a_j - b_j)^2} \quad (13)$$

where a_j is the membership value of feature a for its fuzzy set j .

Example

Let's assume we need to calculate the fuzzy distance between two users who have ages:

- a) 35 and 40
- b) 45 and 60
- c) 18 and 23

Case a: $\mathbf{a} = \langle 0,1,0 \rangle, \mathbf{b} = \langle 0,1,0 \rangle$. Then

$$dis(\mathbf{a}, \mathbf{b}) = \sqrt{(0-0)^2 + (1-1)^2 + (0-0)^2} = 0$$

$$dif(35,40) = 40 - 35 = 5$$

$$fdis(35,40) = 0 \times 5 = 0 \quad (\text{similar users})$$

Case b: $\mathbf{a} = \langle 0,1,0 \rangle, \mathbf{b} = \langle 0,0,1 \rangle$. Then

$$dis(\mathbf{a}, \mathbf{b}) = \sqrt{(0-0)^2 + (1-0)^2 + (0-1)^2} = \sqrt{2}$$

$$dif(60,45) = 60 - 45 = 15$$

$$fdis(60,45) = \sqrt{2} \times 15 \quad (\text{opposite users})$$

Case c: $\mathbf{a} = \langle 1,0,0 \rangle, \mathbf{b} = \langle 0.8,0.2,0 \rangle$. Then

$$dis(\mathbf{a}, \mathbf{b}) = \sqrt{(0.8-1)^2 + (0.2-0)^2 + (0-0)^2} = 0.283$$

$$dif(23,18) = 23 - 18 = 5$$

$$fdis(23,18) = 0.283 \times 5 \quad (\text{close users})$$

The example results show that formula (12) satisfies all four features for a necessary local function of fuzzy distance. Therefore, for the fuzzy approach, the fuzzy distance function between two users can be aggregated using formula (11).

3.4 Neighbor selection

After calculating similarity values, the system ranks users according to their similarity to the active user to obtain a set of neighbors for them. The size of the neighbor set can be fixed by choosing the first N users or variable by selecting users whose similarity exceeds a certain threshold. This work distinguishes between the set of neighbors and the set of actual recommendations. The output of the neighbor set is the same as mentioned earlier (distance function), and a priority set is used to refine it.

3.5 Predictions and recommendations

At this stage, the recommendation system assigns a predicted rating to all items that the set of neighbors sees, rather than the active user. The predicted rating $pr_{x,k}$ indicates the expected interest of item S_k to user u_x and is usually computed as the sum of the ratings of u_x environment for the same item S_k :

$$pr_{x,k} = \frac{\sum_{u_y \in N_x} r_{y,k}}{N_x} \quad (14)$$

where N_x denotes the set of neighbors for u_x , who rated item S_k .

Some examples of aggregation functions:

$$pr_{x,k} = k \sum_{u_y \in N_x} sim(\mathbf{u}_x, \mathbf{u}_y) \times r_{y,k} \quad (15)$$

$$pr_{x,k} = m_x + k \sum_{u_y \in N_x} \text{sim}(\mathbf{u}_x, \mathbf{u}_y) \times (r_{y,k} - m_y) \quad (16)$$

The multiplier k serves as a normalizing coefficient and is typically chosen as $k = \frac{1}{\sum_{u_y \in N_x} |\text{sim}(\mathbf{u}_x, \mathbf{u}_y)|}$ and m_y - is the average rating of user u_y .

$$m_y = \frac{1}{|S_y|} \sum_{s_k \in NS_y} r_{y,k} \quad (17)$$

The weighted sum (6) is the most commonly used aggregation function for predicting ratings. Since users typically use rating scales differently, this prediction formula compensates for variations in the rating scale. This allows maintaining predicted ratings for a given user to be close to the average rating of this active user.

Based on the predicted ratings of items that have not yet been rated, seen by the neighbors of the active user, the recommendation system sorts them in descending order according to their predicted ratings to form a prediction list for the active user u_a .

$$\text{PredSet}(u_a) = \{s_k | s_k \in S_i \subset N_a, s_k \notin S_a\} \quad (18)$$

The rank of item s_k in the prediction list u_a , $\text{Rank}_{u_a}(s_j)$ is the position of item s_k in the prediction list for active user u_a . Accordingly, we can define the recommendation list $\text{RecmSet}(u_a)$ for the active user u_a as the set of items with the highest rating N_r in $\text{PredSet}(u_a)$, which is given by:

$$\text{RecmSet}(u_a) = \{s_k | s_k \in \text{PredSet}(u_a), \text{Rank}_{u_a}(s_j) \leq N_r\} \quad (19)$$

It is expected that objects with the highest rating will be the most predominant, so the user is likely to explore objects in an ordered list, starting from the top, hoping to find interesting objects.

4. Experiments

The experiments were conducted using the Python programming language in the Jupyter Notebook environment as a standalone interface for the analyst. The software product is primarily intended for determining the best locations based on machine learning models.

Computation hardware:

- OS Microsoft Windows 10
- Intel Core i5 7300HQ 2.5 GHz – 3.5 GHz
- 16 GB of RAM
- SSD storage drive
- Graphics card: Nvidia Geforce 1050Ti

Dataset description:

- Utilized the original MovieLens dataset comprising 100,000 ratings by 943 users for 1682 movies.
- Ratings categorized from 1 (poor) to 5 (excellent).
- Each user rated a minimum of 20 movies.
- Demographic data (age, gender, occupation, zip code) available for all users.
- Movie information includes title, release date, video release date, and genre (e.g., Action, Comedy, Drama).

Experiment design:

1. User selection criteria:
 - Included users who rated at least 60 movies.

- Split the users into two groups:
 - a) Group 1 (20 movies for building the user model).
 - b) Group 2 (40 movies for testing).
 - Resulted in 497 eligible users providing 84,596 ratings.
2. Random split generation:
 - Created five random splits of training and active users.
 - Each split involved selecting 50 active users and utilizing the remaining 447 users as training users.
 - These splits were labeled as split-1, split-2, ..., split-5 for subsequent cross-validation.
 3. Cross-validation procedure:
 - Conducted five-fold cross-validation, repeating experiments five times, once with each split.
 - Each split served as a distinct training and testing dataset.
 4. Training and testing phases:
 - Training phase - utilized the set of training users (447 users) to find neighbors for the active user.
 - Testing phase - divided ratings of each active user randomly into two sets:
 - a) Training ratings (34%)
 - b) Test ratings (66%)
- Training ratings used to model the user, while test ratings remained unseen for prediction evaluation.

5. Results

In this experiment, we are running a recommendation system using fuzzy distance and comparing its results with classical systems (Pearson correlation and cosine similarity). The size of the neighbor set is kept at 30 for all experiments. In this experiment, we are running the system over the entire database of training users, even if it takes a long time. The difference between gender (profession) values is either 0 if both users have the same gender (profession), or 1 otherwise. This is consistent with our reasoning for establishing opposing values, as far as possible. In addition, a certain normalization is used for age values to ensure that they fall within the same GII range, i.e., [0, 5]. Each age value is multiplied by $(5/\text{MAX})$, where MAX is the oldest user in the system and no younger than 60 years old. The system selects movies from the set of test ratings of the active user one by one. After that, it predicts ratings for them based on the set of all neighbors who rated the same movie. Once the predicted ratings are obtained, the system compares them with the actual ratings provided by the active user. Figures 4-8 show the percentage of correct predictions obtained for fifty active users. Each graph shows the percentage of the number of ratings that the system correctly predicted, out of the total number of available test ratings for the active user.

Table 2
MAE and Coverage – Split 1

	FD	Cosine	Pearson
MAE	0.737490	0.762880	0.763961
Coverage	0.989313	0.982093	0.974973

Table 3
Comparison of prediction results – Split 1

	Greater	Same	Smaller
FD with Cosine	22	6	22
FD with Pearson	29	5	16

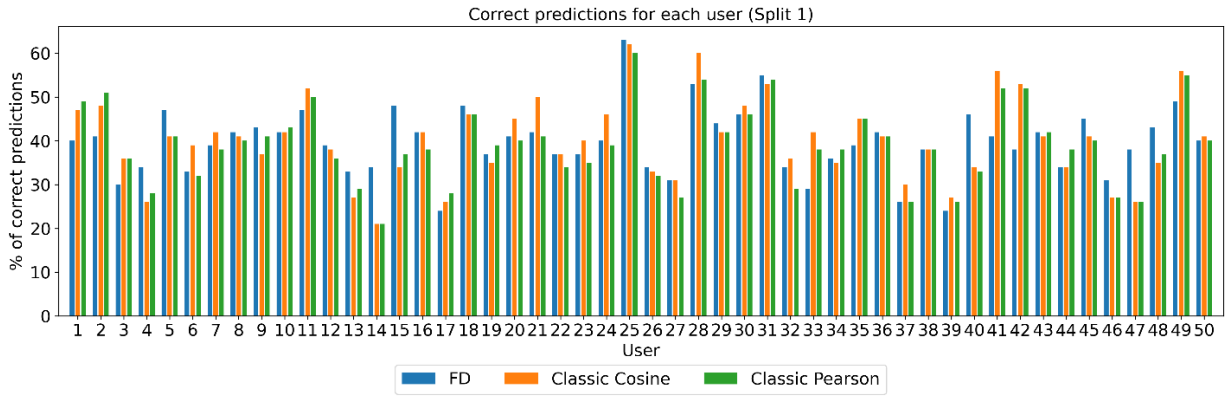


Figure 4: Split 1

Table 4
MAE and Coverage – Split 2

	FD	Cosine	Pearson
MAE	0.760839	0.789274	0.787979
Coverage	0.984859	0.982352	0.976602

Table 5
Comparison of prediction results – Split 2

	Greater	Same	Smaller
FD with Cosine	26	1	23
FD with Pearson	25	5	20

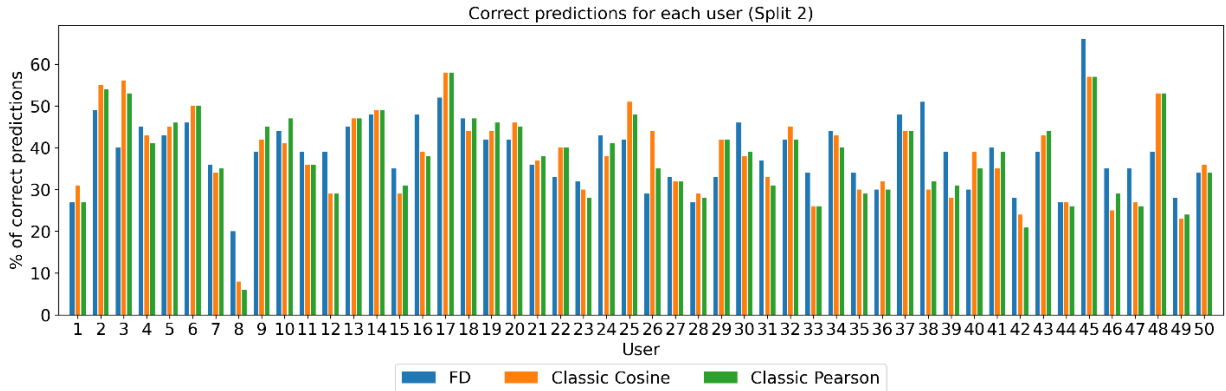


Figure 5: Split 2

Table 6
MAE and Coverage – Split 3

	FD	Cosine	Pearson
MAE	0.743868	0.785782	0.788918
Coverage	0.980292	0.978136	0.971512

Table 7
Comparison of prediction results – Split 3

	Greater	Same	Smaller
FD with Cosine	26	4	20
FD with Pearson	28	2	20

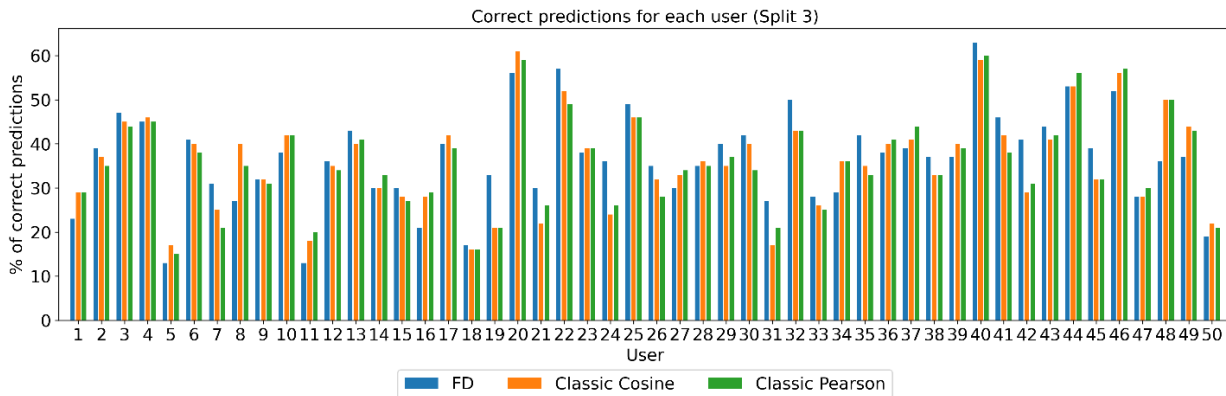


Figure 6: Split 3

Table 8
MAE and Coverage – Split 4

	FD	Cosine	Pearson
MAE	0.787537	0.837564	0.832438
Coverage	0.987336	0.976205	0.970355

Table 9
Comparison of prediction results – Split 4

	Greater	Same	Smaller
FD with Cosine	29	5	16
FD with Pearson	25	10	15

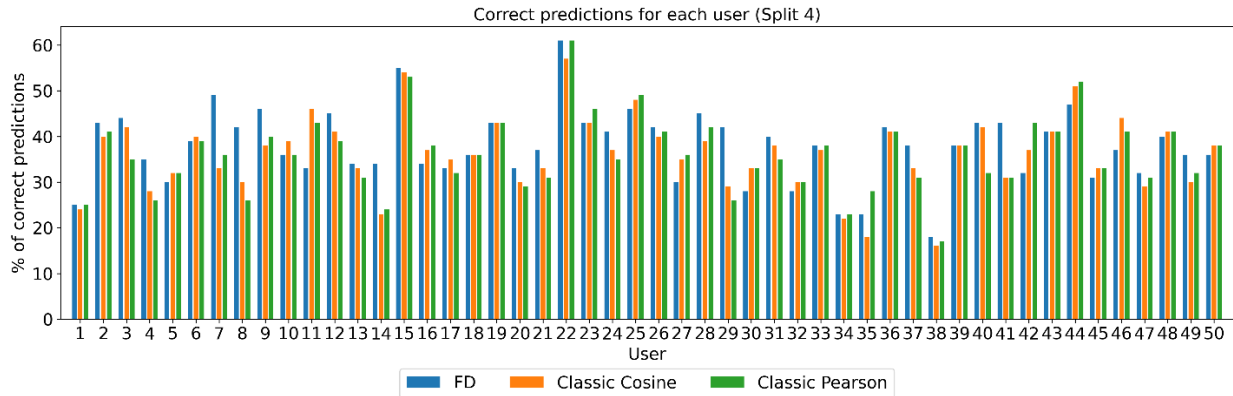


Figure 7: Split 4

Table 10
MAE and Coverage – Split 5

	FD	Cosine	Pearson
MAE	0.751824	0.765969	0.768245
Coverage	0.975372	0.976775	0.968327

Table 11
Comparison of prediction results – Split 5

	Greater	Same	Smaller
FD with Cosine	24	3	23
FD with Pearson	25	5	20

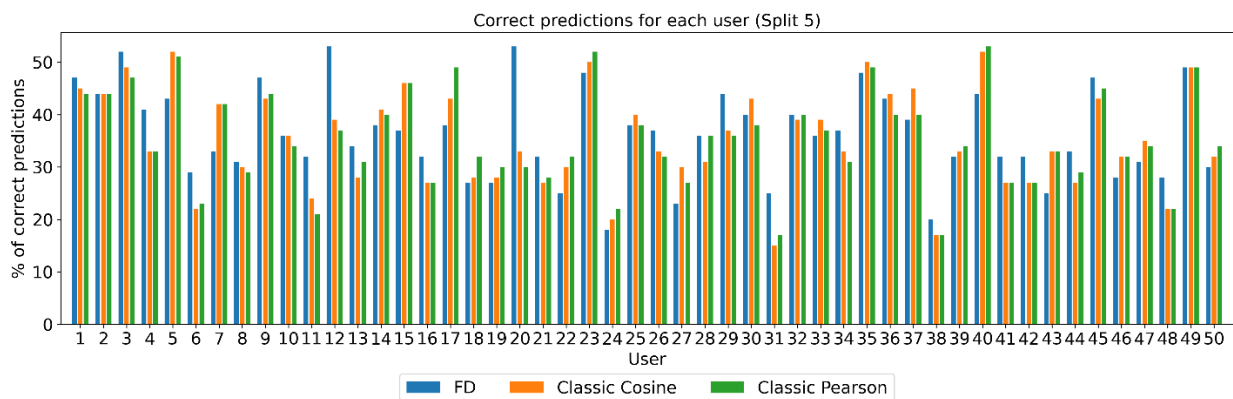


Figure 8: Split 5

6. Discussions

The results of the experiment conducted for each random split of 50 active users and 447 training instances are shown in figures 4-8 and tables 2-11, which depict:

- a table with the results of calculating the Mean Absolute Error (MAE), which is a measure of the accuracy of the recommendation system, and Coverage [14], which is a measure of the percentage of items for which the recommendation system can provide predictions;
- a table comparing the developed algorithm with classical algorithms (Pearson correlation and cosine similarity), where the comparison measure is the percentage of correctly predicted ratings for movies from the test dataset for each of the 50 active users. The table shows the number of users belonging to a certain group, where Greater group has a higher percentage of correctly predicted ratings compared to classical methods, Same group has the same percentage, and smaller group has a lower percentage;

- a graph showing the percentage of correctly predicted ratings using the implemented algorithms for each user.

Based on the obtained data, the following conclusions can be drawn:

- the Mean Absolute Error (MAE) of the developed algorithm is better (smaller) compared to classical approaches, indicating that the deviation of predictions generated by the recommendation system from the true ratings specified by the active user of the recommendation system has decreased;
- the percentage of items for which the recommendation system can provide predictions (Coverage) [15] remained at the same level, and in some cases even increased;
- in all 5 runs, the percentage of correctly predicted ratings was higher for most users.

Higher prediction values obviously illustrate that a better set of corresponding neighboring users has been found, thus increasing the accuracy of the recommendation system.

7. Conclusions

In this study, we proposed and evaluated a recommendation system that leverages fuzzy logic for both user model formation and similarity computation. The research outcomes indicate several key findings:

Enhanced user modeling: By incorporating fuzzy logic into the user modeling process, we were able to capture nuanced user preferences beyond simple ratings. The integration of fuzzy sets for demographic attributes and genre interest indicator resulted in a more accurate representation of user tastes.

Improved similarity computation: The introduction of fuzzy distance metrics facilitated a more robust comparison between user models, considering the partial degree of membership in fuzzy sets. This approach addressed the limitations of traditional distance functions, particularly in handling diverse and imprecise user features.

Superior recommendation accuracy: Experimental results demonstrated that the recommendation system utilizing fuzzy logic outperformed classical approaches, such as Pearson correlation and cosine similarity. The system achieved lower Mean Absolute Error (MAE) and higher prediction accuracy, indicating its effectiveness in providing personalized recommendations.

Stable coverage: Despite the introduction of fuzzy logic, the recommendation system maintained stable coverage, ensuring that a wide range of items could be recommended to users. This suggests that the proposed approach strikes a balance between accuracy and coverage, essential for practical recommendation systems.

In summary, the integration of fuzzy logic in user modeling and distance computation proved to be a promising approach for enhancing recommendation systems' performance.

8. References

- [1] D. Roy, M. Dutta, A systematic review and research perspective on recommender systems, *Journal of Big Data*, Vol. 9, Issue 59, 2022. doi:10.1186/s40537-022-00592-5.
- [2] P.Dahiya, N. Duhan, Comparative Analysis of Various Collaborative Filtering Algorithms, *International Journal of Computer Sciences and Engineering*, Vol. 7, Issue 8 (2019) 347-351. doi:10.26438/ijcse/v7i8.347351.
- [3] E. Çano, M. Morisio, Hybrid Recommender Systems: A Systematic Literature Review, *Intelligent Data Analysis*, Vol. 21, Issue 6, 2017, pp. 1487-1524. doi:10.3233/IDA-163209.
- [4] M. Fauzi, S. Putra, A. A. Stephanie, I. S. Edbert, D. Suhartono, Hybrid Approaches for Customer Segmentation and Product Recommendation, *Proceedings of the International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, 2023, pp. 324-329. doi:10.1109/ICIMCIS60089.2023.10348619.

- [5] A. Vall, M. Dorfer, H. Eghbal-zadeh, M. Schedl, K. Burjorjee, G. Widmer, Feature-combination hybrid recommender systems for automated music playlist continuation, *User Modeling and User-Adapted Interaction*, Vol. 29 (2019) 527–572. doi:10.1007/s11257-018-9215-8.
- [6] S. Fremal, F. Lecron, Weighting Strategies for a Recommender System Using Item Clustering Based on Genres, *Expert Systems with Applications*, Vol. 77 (2017) 105-113. doi:10.1016/j.eswa.2017.01.031.
- [7] P.E. Sytnikova, M.O. Hrebeniuk, Recommendation system based on a compact hybrid user model, *Automated control systems and automation devices*, Vol. 179 (2023) 32-42. doi:10.20837/0135-1710.2023.179.032.
- [8] F. Fkih, Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison, *Journal of King Saud University - Computer and Information Sciences*, Vol. 34, Issue 9 (2022) 7646-7669. doi:10.1016/j.jksuci.2021.09.014.
- [9] H. Zitouni, On Solving Cold Start Problem in Recommender Systems Using Web of Data, *Proceedings of the International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, Vol. 4, 2022, pp. 1-8. doi:10.1109/PAIS56586.2022.9946899.
- [10] P. Kumari, G. Kaur, P. Singh, A. Kumar, Movie Recommendation System for Cold-Start Problem Using User's Demographic Data, *Proceedings of the International Conference on Electrical, Computer and Energy Technologies (ICECET)*, ISSN 0973-6107, 2023, pp. 1-5. doi:10.1109/ICECET58911.2023.10389506.
- [11] R. Y.Toledo, L. Martinez, Fuzzy Tools in Recommender Systems: A Survey, *International Journal of Computational Intelligence Systems*, Vol. 10, (2017) 776-803. doi:10.2991/ijcis.2017.10.1.52.
- [12] M. Panwar, A. Sharma, O. P. Mahela, B. Khan, Fuzzy Inference System Based Intelligent Food Recommender System, *Proceedings of the Asian Conference on Innovation in Technology (ASIANCON)*, Vol. 3, 2023, pp. 1-6. doi:10.1109/ASIANCON58793.2023.10270048.
- [13] F. Kaviani, Recommender System in Social Networks using Fuzzy Logic, *Proceedings of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, Vol. 3, 2023, pp. 1-7. doi:10.1109/ICECCME57830.2023.10253120.
- [14] H. Koohi, K. Kiani, User based Collaborative Filtering using fuzzy C-means, *Measurement*, Vol. 91 (2016) 134-139. doi:10.1016/j.measurement.2016.05.058.
- [15] K. Najmani, L. Ajallouda, El H. Benlahmar, N. Sael, A. Zellou, Offline and Online Evaluation for Recommender Systems, *Proceedings of the International Conference on Intelligent Systems and Computer Vision (ISCV)*, 2022, pp. 1-5. doi:10.1109/ISCV54655.2022.9806059.