International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 6
ISSN: 2088-5334

A Healthcare Recommender System Framework

Kher-Ning Ooi [a], Su-Cheng Haw [a,*], Kok-Why Ng [a]

[a] *Faculty of Computing and Informatics, Multimedia University, Jalan Multimedia, 63100 Cyberjaya, Malaysia*
Corresponding author: [*]sucheng@mmu.edu.my

***Abstract*—After the pandemic hit every part of the world, healthcare awareness is slowly rising among every human being, especially leaders of each country. Due to a shortage of manpower in the healthcare industry, patients tend to search the internet for some self-diagnoses. This way is extremely dangerous as patients might end up using the wrong treatment such as taking the wrong medication to treat their sickness since there are so many different remedies posted on the internet without valid recognition from the healthcare professionals. To aid in overcoming this problem, this research will be building a Healthcare Recommender System. The goal of a Healthcare Recommender System (HRS) aims to supply its user (patient) with medical information that is meant to be highly relevant and tailored to an individual's need. Hence, this paper gives an overview of various recommender systems, datasets employed, and evaluation metrics used in the healthcare system. In addition, we propose the framework for the HRS to capture user input on their condition and recommend the next course of action. The steps involved in our recommender system includes choosing the dataset and techniques, data cleaning and preprocessing, building the recommender system, training the recommender engine, and finally performing the prediction. We generate the accuracy of prediction and analyze with some results. From the experimental results, Cosine Similarity has the highest accuracy compared to Jaccard Similarity and Euclidean Distance.**

***Keywords*—Recommender system; healthcare recommender system; content-based recommender; cosine similarity; jaccard similarity; euclidean distance.**

Manuscript received 12 Dec. 2022; revised 20 Apr. 2023; accepted 19 Aug. 2023. Date of publication 31 Dec. 2023.

I. INTRODUCTION

Since the COVID-19 pandemic hit the world in the second quarter of 2019, the healthcare industry has played an utmost important role [1], [2]. Even in urban areas, the need for healthcare in every part of the world is always emphasized. There are many solutions to improve the world's healthcare industry, such as building more healthcare buildings, hiring more healthcare workers, and introducing more healthcare courses for interested people to pursue. Places with large spaces can build more healthcare buildings, but the healthcare industry will still be nothing without healthcare workers. During the COVID-19 pandemic, the insufficient healthcare workers treating patients caused chaos in almost every hospital. Many patients were admitted to the hospital, requiring many healthcare workers to work overtime. On top of this, many healthcare workers became sick because of sleep deprivation, leading to a low immune system and a shortage of healthcare workers.

Nowadays, almost every human owns a gadget, especially a smartphone or computer. Technologies have expanded more quickly than before, which aids the revolution of the healthcare system. Due to a workforce shortage in the healthcare industry, patients tend to search the internet for self-diagnosis. This way is hazardous as patients might end up using the wrong treatment, such as taking the wrong medication to treat their sickness since there are so many different remedies posted on the internet without valid recognition from healthcare professionals. Considering this, a Healthcare Recommender System (HRS) is introduced [3], [4]. Generally, a recommendation system is an algorithm that guides customers to suitable products or services they most want. Simply put, a recommender system allows people to choose based on their interests, needs, or preferences in various scenarios.

People cannot get valuable information for enhancing their well-being because of the enormous amount of healthcare data available internationally. Using HRS can help with information overload, as extracting useful, personalized information from large amounts of data can be challenging [5], [6], [7]. Both end users and healthcare professionals can use these systems to make more informed judgments about the health of their patients [8], [9]. The HRS can inspire and involve users to provide better options and practical knowledge. The more users enter their data about the disease, the more accurately the system can recommend which healthcare provider suits them best [10], [11]. In this case, it will be more efficient for both parties since the world that we are living in now emphasizes efficiency and accuracy. To do so, the accuracy and overall performance of the HRS are crucial, and the techniques used to build it should be chosen carefully.

This paper used a profile based HRS component to store a Patient Health Record (PHR) owner's medical history. An HRS calculates a set of potentially relevant items of interest to a user. These items can be shown to the user while online examining the PHR because they come from reliable repositories. The system can offer the user some potential treatments based on the dataset. The HRS will be constructed using the Content-Based Filtering method, and the scores of the identified methods will be generated and compared.

Fig. 1 shows a recommender system with different techniques to make user predictions. Choosing the proper recommender system techniques is crucial to obtaining a precise prediction model for users, especially in the healthcare industry associated with a user's health and life.

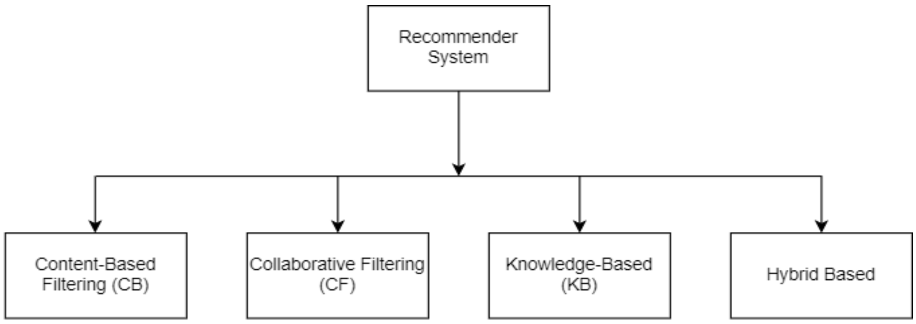


Fig. 1 Grouping of Recommender System

Content-based (CB) filtering involves utilizing characteristics of items to suggest other items similar to what the user has shown a preference for based on their prior actions or explicitly provided feedback [12], [13]. CB recommendation systems assess previous user item encounters to ascertain how similar users and items are [14]. This similarity information is then utilized to create a list of suggested items. CB models tend to produce good results when many user-item interactions are accessible. For example, a patient was recommended to read a health article about vitamin C from Pharmacy A's website. The patient clicked on the link, read it, and left a good comment with 5-star rating. The patient then visited a different Pharmacy's website for another vitamin C article. This data will be collected, and other pharmacy websites' vitamin C articles will be recommended to patients who are also top-rated by other patients.

The CB recommender system needs significant item feature data to build a user preference model. The fact that the item feature information is minimal and contains redundant or missing values proves it is extremely inconsistent, resulting in poor suggestions and inadequate learning of user preference profiles. We can infer that any CB recommender model relies heavily on retrieving effective item feature data. The information about the item's features is primarily textual and occasionally inconsistent. We risk having an insufficient feature representation if we vectorize this data using a method like Term Frequency and Inverse Document Frequency (TF-IDF) [15], [16]. In short, the CB filtering technique collects data from users and analyses the data to recommend other similar data to users. Hence, users' data is crucial to receive the right recommendation.

Using a technique known as collaborative filtering (CF), users can reject goods based on the opinions of other users who share their interests [17], [18], [19]. CF searches through a large amount of user data to locate a smaller group of users with similar preferences [20]. It combines the products people enjoy with other factors to get a sorted list of recommendations. Jane, for instance, enjoys reading the Percy Jackson and Harry Potter book series. Using CF techniques, it will suggest Jane books in the same genre to people who enjoy the same book, such as the Fantastic Beast book series.

Finding comparable people or items is the first step in developing a system that automatically recommends products to users based on their preferences. One of the main advantages of collaborative filtering techniques is that they can make recommendations across genres. It is only determined by whether a user gives a product an explicit or implicit rating. For example, if two users give the same ratings to ten movies, even though they are very different in age. To address the third issue, there are numerous methods for calculating error, some of which are applicable to applications other than CF recommenders. These methods can also be used to test the accuracy of predictions.

Calculating the Root Mean Square Error (RMSE), which involves forecasting ratings for a test dataset of user-item pairs for which the rating values are already known, is one method for evaluating the accuracy. The difference between the actual and anticipated values would constitute the error. The RMSE can be calculated by squaring all of the test set error values, finding the average (or mean), and then taking the square root of that average. Another method to gauge accuracy is Mean Absolute Error (MAE), which determines the error amount by determining its absolute value and then averaging all error values. There are three main types of CF techniques: memory-based, model-based, and hybrid. Memory-based techniques require a user's rating to compare users or goods and suggest them with undiscovered ones.

User-based and item-based collaborative filtering algorithms are the two main categories. By identifying users who have viewed or rated similar content, we can leverage their ratings to propose new content using user-based CF algorithms. This approach can occasionally result in a "cold start," which occurs when there is insufficient data on the user's preferences and nothing to compare. On the other hand, the item-based approach uses a similar idea but starts with a specified item. Using the selections of other users, we can then identify comparable things. Item-based approaches do not have the cold-start issue because their similarity matrices tend to be smaller, and a single item is sufficient to recommend more comparable items, which lowers the cost of locating neighbors. One drawback of this approach is that, compared to user-based CF, the recommendations would be less diverse.

Model-based CF consists of creating a model from the rating dataset. For instance, we can utilize a portion of a dataset as a model instead of the entire dataset each time to generate recommendations. It will be more effective in this manner. We can construct the model using several methods, such as improving memory-based algorithms, solving linear algebra problems, and solving probability problems. This method has the drawbacks of being rigid, and the forecasts' accuracy depends on how the model is constructed.

The matrix factorization method will be required to address this issue if our data is lacking or sparse. Moreover, Singular Value Decomposition (SVD), which incorporates a second diagonal matrix that encodes the weights or strengths, is a well-liked matrix factorization technique used in model-based CF. We can utilize the RMSE to assess the accuracy of our findings.

Knowledge-based (KB) recommender systems base their recommendations not just on the user's search history but also on their past ratings [21] [22]. Fig. 2 shows that KB recommender system technique is based on users' preferences according to the existing background data that the system has and gives recommendations based on the constraints given by the user.

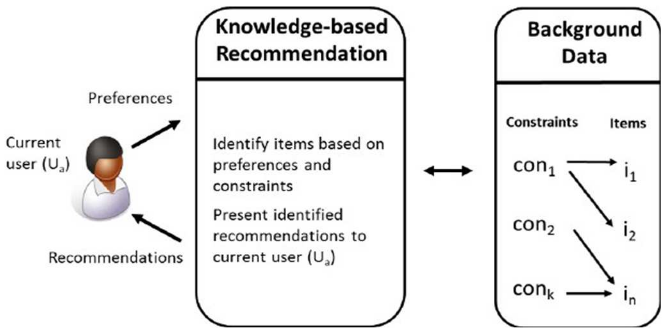


Fig. 2 Knowledge Based Technique

As an illustration, the user searched for a car to be purchased. Then, regardless of size or cost, the system will search the domain knowledge for anything linked to vehicles. Once the algorithm has identified anything connected to vehicles, it combines all its findings into a single database query. Now, since the user may want to be more precise about the outcomes, the user may want to add some specific details, which in a more technological context means tightening and loosening settings. This action generates a new query, and the procedure must be carried out several times until the user locates a suitable item.

Configuring our similarity metrics is essential for creating a strong knowledge base. Finding results that do not exactly match the search query is done using similarity measures. It is critical to comprehend the relative significance of the feature as well as the utility function used to compare two feature values. The parameters can be explicitly defined by a subject-matter expert or discovered from user feedback. Additionally, a decent evaluation approach makes it easier for consumers to locate pertinent findings quickly and consistently. Dynamic criticizing is a more sophisticated method for returning the most pertinent options depending on the recent results. For instance, if a user is looking for a cheap car but has already browsed the model with the lowest features, no further action can be taken to lower the price and satisfy all the user's requirements. Thus, dynamic evaluation is useful in this situation. It considers all potential modifications to the user's query and displays the most prevalent patterns, like "used second-hand automobiles, cheaper price." By doing this, the algorithm looks for patterns across all results and gives people more options.

KB can also be tailored in a variety of ways to each user. The utility and similarity functions for case- and constraint-based recommender systems can be customized by leveraging user feedback to prioritize variables. Dynamic criticisms can also be tailored by considering the support for user history adjustments. For instance, we may monitor users' restrictions for their quotations and make new constraints and recommendations based on popular selections. In some recommender systems, KB can be beneficial. When enough ratings are gathered, they assist in overcoming the cold start issue and transition to collaborative filtering or content-based systems. KB recommenders can be valuable tools in complex or infrequently used item spaces.

Both content-based filtering and collaborative filtering strategies have their drawbacks. Their shortcomings can be solved using hybrid-based strategies [23], [24], [25]. To improve the performance and accuracy of recommender systems and take advantage of their strengths while balancing out their flaws, four main recommendation strategies are used to build hybrids: CF, CB, demographic, and KB. Many apps utilize a HB recommender algorithm because collaborative filtering is a common user problem, and content-based filtering techniques are a novel user problem [26]. Weighted hybrid, mixed hybrid, switching hybrid, feature-combination hybrid, cascade hybrid, feature-augmented hybrid, and meta-level hybrid are the different categories of HB based on how they operate. Fig. 3 shows a general understanding of HB technique by combining inputs from CF and CB recommender systems using a combiner, producing the recommendation for the user.

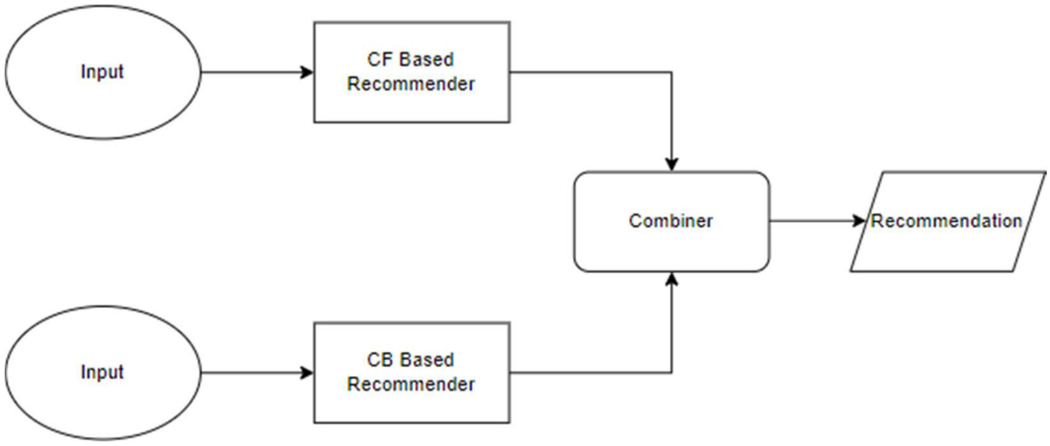


Fig. 3 Hybrid-based Technique Flowchart

Weighted hybrid builds a recommendation list using several recommender results and ratings from each applied strategy using a linear formula, such as P-tango. A CF recommender and a CB recommender comprise this operation. They are first given equal weight. Their weights change over time depending on the predictions or not. The benefits of the recommender system are easily utilized during the suggestion process by using weighted hybrids.

Instead of getting one recommendation for each item, mixed hybrids combine the findings of several recommendation algorithms. Each item is associated with some recommendations generated by various recommendation algorithms. In this operation, a person's performance does not necessarily affect the success of the entire region.

Switching hybrid systems exchanges recommendation techniques based on a methodology that considers the rating that the recommender can produce. Moving the new user issue of the CB recommender to the CF recommendation system, for example, this process can assist in avoiding difficulties particular to one technique. This system will benefit by being aware of the advantages and disadvantages of each of its component recommenders. In contrast, switching hybrids typically makes the recommendation process more challenging because the switching criterion needs to be established.

In order to construct a sequence of preferences between various items, cascade hybridization uses an iterative refinement process. Recommendations from a different method may enhance the other approach. The first recommendation strategy compiles a preliminary list of concepts, refined by the second. Due to the iterative process's coarse-to-fine nature, this method is highly efficient and noise-tolerant. Feature augmentation technique requires more capability from the recommender systems and the ratings and other data generated by the prior recommender.

The meta-level technique uses the internal model of one recommendation approach as input for another. The developed model is commonly, in every case, more enlightening than a solitary grade. Meta-level hybrids can solve the sparsity problem of collaborative filtering procedures, such as creating content-based user profiles using instant-based learning that is then compared cooperatively by using the full model created by the first technique as input for the second technique. Table 1 summarizes the advantages and disadvantages of each recommender system technique.

TABLE I
ADVANTAGES AND DISADVANTAGES OF RECOMMENDER SYSTEM TECHNIQUES

Advantages	Disadvantages
Content-Based	
1. The model does not require other information from other users to make recommendations. 2. The model can identify user's individual selections and recommend users based on other user's preferences. 3. Does not suffer from cold-start problems.	1. Highly inconsistent, such as when the information on an item's features is relatively limited and contains redundant or missing data. 2. Utilizing users' personal and social data leads to no privacy. 3. When we utilize the system to forecast items that are not yet visible, recommendation quality drops. 4. Requires a good amount of item features.
Collaborative Filtering	
1. Without relying on manual engineering features, embeddings can be learned automatically. 2. Does not need the features of the items to be given. 3. Domain knowledge is not required.	1. The system normally cannot embed an item if it is not seen during training, making it impossible to query the model with this item. (Cold-Start problem). 2. Suffers from data sparsity. 3. Uses only user behavior for recommending items.
Knowledge-Based	
1. Help in overcoming cold-start problems. 2. It can be personalized to individual users in several ways. 3. Uses dynamic critiquing.	1. Identifying potential irregularities in the systems, such as circular relationships and redundant rules. 2. Requires an accurate large range of data to perform accurately. 3. It needs to provide the system with high-quality data and information contained in it.
Hybrid Based	
1. Used to overcome the limitations of CB, CF, and KB techniques. 2. Reduces the downside of using individual models.	1. High computational complexity. 2. Requires a large database to keep data metrics updated.

Advantages	Disadvantages
3. Results to more powerful and personalized recommendations for users.	

Han et al. [27] suggested a collaborative filtering recommender system for matching primary care physicians with patients. The researchers used the CF method to carry out their research. The predictive accuracy of their method was higher than the heuristic and CF baseline approaches without the formulated trust measure. Weighted Approximate-Rank Pairwise (WARP) loss was used to train the build model as a learning-to-rank task because the research observed actual patient visits, and most patients had visited fewer than three primary care doctors. When the CF model is trained using stochastic gradient descent and weighted sampling, the trust measure will scale the learning rate toward interactions associated with high levels of patient trust, resulting in improved convergence. Splitting the data over time for temporal cross-validation was done, and the model's performance was compared to that of a heuristic baseline model. With the support of the trust measure, the presented results were encouraging, with the proposed models consistently outperforming the heuristic baseline model by significant margins.

Arshad et al. [28] investigated the state of knowledge management (KM) in Pakistan's public healthcare institutions. They proposed a component-based KM Model that combines an effective KM process framework with all necessary components of a successful KM system. This model aims to improve the interoperability and integration of healthcare information systems, thus promoting better insurance and ensuring the optimal usage of KM in the public healthcare sector. The research employed an "Explanatory Sequential Mixed Methodology Design", which thoroughly examines research issues through two distinct, interactive phases. Quantitative and qualitative surveys were administered to individuals from the prominent public healthcare sector in Pakistan using questionnaires. These public sector healthcare organizations aim to increase their utilization of the KM process framework and component-based KM models to enhance governance, save time and effort in knowledge management, and achieve this through process improvements and technological advancements.

By expediting the general practitioner appointment process, Han et al. [29] hoped to assist a healthcare provider in transforming their primary care health service into a patient-centered service. This was accomplished by implementing a hybrid recommender system that streamlines the procedure for matching patients with primary care physicians. The researchers collected large data, including consultation records, to understand patients' belief in their family physicians. The researchers blended patient and doctor metadata to personalize the doctor recommendations, especially for patients with limited prior interactions with family doctors. A hybrid matrix factorization model was created to depict patients and medical professionals as sequential fusions of variables that are not directly measured describing their traits and relations. This technology offers patients individualized doctor recommendations when used in conjunction with a selection of models based on rules procedure. The proposed hybrid approach was found to be more accurate than a heuristic baseline and a conventional collaborative filtering recommender system. The inclusion of the trust metric improved the performance of both the hybrid and CF systems even further.

Using the Restricted Boltzmann Machine (RBM)-Convolutional Neural Network (CNN) deep learning method, Sahoo et al. [30] proposed intelligent HRS. This work sheds light on how big data analytics can be leveraged to construct an effective health recommender engine. The health recommender engine makes use of Matrix factorization, SVD, Variable Weighted BSVD (WBSD), and deep learning strategies, among other privacy-preserving recommender systems. This paradigm makes it possible to define neighborhood and locality for both hidden and visible units. The RBM's sub- windows of the visible matrix could represent distinct image patches. The weights are dispersed among the hidden unit clusters, and the hidden-visible connections in the RBM are local. Since it is easy to measure and identify, they used the RMSE method in this instance so that we can rapidly calculate the recommender system's quality factor. When two error evaluations, such as RMSE and MAE, are considered, the suggested RBM-CNN method offers higher accuracy than all other methods. For selecting a hospital for a specific patient, the combination of an RBM with CNN in a deep learning environment improves the recommendation accuracy.

Jabeen et al. [31] offered an IoT-based effective community-based recommender system that makes recommendations about the physical and dietary plans and detects the type of heart disease. In an IoT-based setting, the data is transmitted to the server. Then, a cardiovascular disease prediction system is implemented, which can recognize the condition and group it into one of eight classes. The system gathers information from patients about many characteristics to determine the condition. The algorithm then offers the patient the best guidance out of a large pool of suggestions gathered from cardiologists. The performance is evaluated using Precision and Recall. The degree of precision will reflect how well the diseases are categorized. The recall will gauge how well the classifier can identify diseases. The absolute difference between a cardiologist's real suggestions and those made by a recommender system is identified by MAE. The efficacy of the suggested heart disease prediction model is assessed by running it twice. The model was run the first time using the specified feature selection technique, and the second time without it. This procedure aids in assessing the effectiveness of the feature selection method. The technique works well for patients from outlying areas where a skilled cardiologist is typically unavailable. This proposed work might help a young, inexperienced cardiologist make an immediate medical choice.

Ren et al. [32] created the Hybrid Collaborative Filtering Model for Healthcare, or HCFMH to suggest search phrases to physicians. On data set variations with various levels of data sparsity, the HCFMH model beat the baseline approaches. This means that compared to current state-of-the-art RS approaches, a specially designed algorithm for the healthcare industry yields non-trivial speed improvements. For example, an HR@5 of 0.4317 indicates that there is a more than 43% chance that the term a user will search for will appear among the top-five suggested items. To produce recommendations, their approaches in this research rely on relationships and co-occurrence between search phrases and ICD codes. The pros of this research work is that it can help in time and effort efficiency by identifying items proactively. On top of that, this research will aid in providing useful reminders to clinicians about the information that might have been significant that they may have missed. The limitation of these developed methods by these researchers is them facing a "coldstart" problem. If a patient's encounter data is missing from the system, the researchers cannot produce recommendations for them.

Lambay & Mohideen [33] contributes to the study of information about big data analytics in the healthcare industry. Recommender systems are typically created using machine learning. An intelligence-based approach is explored in different phases to build a health recommender system which includes phases like information collection, learning and prediction. Different recommender system techniques are used for different machine learning techniques. In the data visitation phase, a collaborative filtering technique is used as it aids in quickly accessing healthcare-related information that makes it a patient-centered approach. Of course, the disadvantage to this technique is having the cold-start and scalability problem. Since patients' data are needed to recommend an accurate solution for them, PrivacyPreserving Collaborative Filtering (PPCF) is needed which prevents privacy loss by using the concept of Arbitrary Distributed Data (ADD). One upside about this technique is that the performance is not that great. The method employed to make healthcare recommendations is a hybrid one. Multiple Kernel Learning (MKL) and Adaptive Neuro Fuzzy Inference System (ANFIS) are combined to create the hybrid technique. The accuracy of each technique in this study is checked using the MAE (Mean Absolute Error) value.

An automatic healthcare system is presented by Rustam et al. [34] that can successfully take the place of a doctor at the first stage of diagnosis and help save time by suggesting the necessary measures. Using patient symptoms as input, this study develops an automatic system that diagnoses a disease and recommends appropriate preventative measures. Two modules make up the suggested method. Using the dataset of clinical symptoms, the machine learning algorithms are taught in module 1. Categorical data conversion into text so that the data can be connected after speech data conversion into text in Module 2. Preprocessed data is divided 80:20 between training and testing sets. Using the TF-IDF, features from the testing and training sets are extracted. A microphone is used by Module 2 on the patient's side to record patient voice data. To make sure the conversion of audio to text data is done correctly, the speech recognizer's performance is also assessed. The usefulness of two feature extraction techniques—TF-IDF and BoW—for textual data is tested using the data. These models' performance is enhanced by employing the grid search approach to identify the best hyperparameters. The accuracy of the suggested method is assessed and contrasted using accuracy, precision, recall, and F1-score. Experiments' findings show that text data often displays greater accuracy than category data. When classifiers have access to a large feature vector of textual data, their accuracy increases. A 99.9% prediction accuracy was achieved, and equivalent outcomes for precision, recall, and F1 score were also attained. Real-time scalability and good performance characterize the offered approach.

An IoT device-based medical recommender system to detect and cure chronic diseases was presented by Nanehkaran et al. [35]. The current methodology used the dataset of digitized patient health records housed in the PhysioNet data repository. The current dataset contains patient health records that have been documented in accordance with the disorders that have been identified and the doctor's diagnosis. The dataset's disease type, as determined by the nearest neighbor classification approach, is used to train disease-related symptoms. The suggested method will be assessed using the provided suggestions in accordance with the accuracy, sensitivity, and accuracy standards. Afterwards, the proposed method will be contrasted with other approaches already employed in this area. K-NN classification model is used in this recommender system. According to experimental findings, the suggested method performs well in terms of the evaluation criteria. The proposed method's high accuracy level suggests it has a great capacity for making effective treatment recommendations. The proposed method's inability to access actual hospital data is a significant flaw. Medical facilities find it difficult to quickly disclose genuine information to researchers due to patient privacy issues.

Shambour et al. [14] produced a hybrid content-based multi-criteria CF strategy to help patients find the best doctors according to each user preferences. They proposed three modules: the hybrid prediction module, the item-based content module, and the item-based CF module. In contrast to traditional item-based CF similarity algorithms, the Bhattacharyya coefficient is effective at extracting global information from sparse rating datasets, which is why it is used as a similarity measure for global similarity. The researchers examined the local similarities between the objects using the Cosine similarity metric. The Jaccard coefficient is used to determine how many users gave the same scores to both items regarding structural similarity. Their proposed method also includes the item reputation score. in the item-based content module. The content similarity compares category representations of physicians rated by patients to recommend new physicians who have not seen them before. Under specific circumstances, the final predicted rating is determined by the hybrid prediction module employing the switching hybridization strategy. In contrast to the standard item-based CF-based recommendation algorithms that are currently in use, the experimental results on a real healthcare ratings dataset demonstrate that the proposed method can provide suggestions that are extremely reliable in terms of predicted accuracy and coverage even in the case of extremely sparse data.

II. MATERIALS AND METHOD

A. Theoretical Framework

The prototype will be built upon the chosen dataset and techniques. The chosen dataset will first be cleaned and undergo some data preprocessing before implementing it into the prototype. Then, a recommender system model will be built and trained. Model evaluation will also be implemented at the end of building the prototype. The evaluation metrics will be visualized in a bar chart. A graphical user interface will also be implemented for the prototype for better visualization and system interaction for users. Fig. 4 shows the flowchart of the prototype.

B. The Dataset

The dataset used for model training in the prototype is "mtsamples.csv" dataset, taken from Kaggle.com [36]. Medical data are quite difficult to find as this research touches on the medical area and Health Insurance Portability and Accountability Act (HIPAA) privacy requirements. "mtsamples.csv" dataset is the best that could be found and used for this project. The "mtsamples.csv" dataset consists of 5 columns and 4999 rows. Fig. 5 shows the partial view of the dataset.

The columns are "description", "medical_specialty", "sample_name", "transcription" and "keywords".

- The "description" column describes the description of transcription.
- The "medical_specialty" describes the medical specialty classification of transcription. The "sample_name" describes the transcription title.
- The "transcription" column describes the sample medical transcriptions.
- The "keywords" columns describe the relevant keywords from transcription.

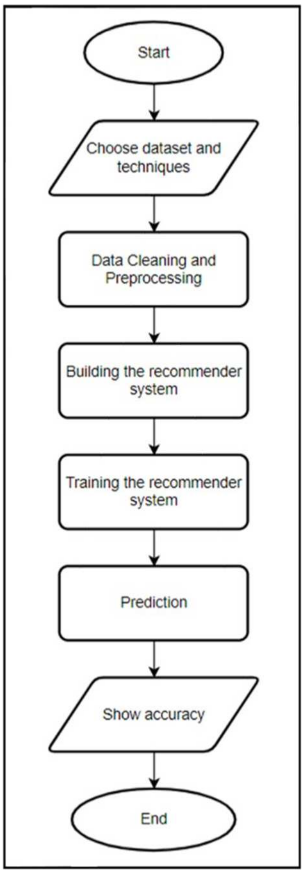


Fig. 4 Flowchart of the prototype

description	medical_specialty	sample_name	transcription	keywords
A 23-year-old white fe	Allergy / Immunology	Allergic Rhinitis	SUBJECTIVE:, This 23-year-old white fe	allergy / immunology, allergic rhini
Consult for laparoscop	Bariatrics	Laparoscopic Gastric Byp	PAST MEDICAL HISTORY:, He has difficu	bariatrics, laparoscopic gastric bypa
Consult for laparoscop	Bariatrics	Laparoscopic Gastric Byp	HISTORY OF PRESENT ILLNESS: , I have s	bariatrics, laparoscopic gastric bypa
2-D M-Mode. Doppler	Cardiovascular / Pulmo	2-D Echocardiogram - 1	2-D M-MODE: , ,1. Left atrial enlargem	cardiovascular / pulmonary, 2-d m-
2-D Echocardiogram	Cardiovascular / Pulmo	2-D Echocardiogram - 2	1. The left ventricular cavity size and w	cardiovascular / pulmonary, 2-d, dc
Morbid obesity. Lapa	Bariatrics	Laparoscopic Gastric Byp	PREOPERATIVE DIAGNOSIS: , Morbid ol	bariatrics, gastric bypass, eea anast

Fig. 5 Partial view of the dataset.

C. Data Cleaning

In the cleaning process, some unwanted columns were deleted which are "Unnamed: 0", "sample_name" and "transcription" column. Fig. 6 shows the outcome of the table after dropping the unwanted columns.

```
df.drop(['Unnamed: 0', 'transcription', 'sample_name'], axis=1,inplace=True)
df.head()
```

	description	medical_specialty	keywords
0	A 23-year-old white female presents with comp...	Allergy / Immunology	allergy / immunology, allergic rhinitis, aller...
1	Consult for laparoscopic gastric bypass.	Bariatrics	bariatrics, laparoscopic gastric bypass, weigh...
2	Consult for laparoscopic gastric bypass.	Bariatrics	bariatrics, laparoscopic gastric bypass, heart...
3	2-D M-Mode. Doppler.	Cardiovascular / Pulmonary	cardiovascular / pulmonary, 2-d m-mode, dopple...
4	2-D Echocardiogram	Cardiovascular / Pulmonary	cardiovascular / pulmonary, 2-d, doppler, echo...

Fig. 6 Dataset output after deleted columns

Moving on, the dataset will be checked for null values. If any null values are present in the "medical_specialty" and "keywords" column, the whole row will be deleted. Fig. 7 shows the number of null values present in the dataset's column. There are 33 null values present in the "transcription" column while the "keywords" column has 1068 null values. Hence, the row that contains null values in "keywords" column will be deleted. As for the "medical_specialty" column, it does not have any null values.

```
data.isnull().sum()
```

```
Unnamed: 0              0
description             0
medical_specialty       0
sample_name             0
transcription          33
keywords             1068
dtype: int64
```

Fig. 7 Number of null values present in the dataset's column

Subsequently, the dataset will be checked for duplicated rows as well. If there are any duplicated rows in this dataset, one of the duplicated rows will be deleted. Fig. 8 shows the number of duplicated data in the dataset which is 0. On account of this, the dataset has no duplicated rows to be deleted.

```
data.duplicated().sum()

0
```

Fig. 8 Number of duplicated rows in the dataset

Next off, the dataset will be checked for whitespaces as this might cause the system to not function correctly if these whitespaces are not removed properly from the dataset. Fig. 9 shows the checking of the whitespace present in each column and removing them. Fig. 10 shows the checking of the whitespace present in each column after removing them just in case there are any miss outs.

```
for column in data.columns:
    print("Column: ", column)
    print("Whitespace: ", data[column].str.contains('\s').sum())
    # remove whitespace from the column
    data[column] = data[column].str.strip()

Column:  description
Whitespace:  4991
Column:  medical_specialty
Whitespace:  2204
Column:  sample_name
Whitespace:  4777
Column:  transcription
Whitespace:  4960
Column:  keywords
Whitespace:  3842
```

Fig. 9 Checking of whitespace in each column

Following collection, the data is frequently stored in a matrix, where the rows represent users, the columns represent things, and the cells represent user ratings. This matrix is frequently referred to as a "user-item matrix" or "ratings matrix." Because not all objects have ratings, not all users have rated all the items, and most of the cells in the user-item matrix are empty, it is known as a sparse matrix. The user-item matrix is the most important element in developing a model-based collaborative filtering system. The matrix must be preprocessed to address missing values, outliers, and other data quality issues. Additionally, it is important to standardize the ratings if the data was compiled from multiple sources.

```
for column in data.columns:
    print("Column: ", column)
    # check if column has any leading or trailing whitespaces
    if (data[column] != data[column].str.strip()).any():
        print("Whitespace found in column")
    else:
        print("No whitespace found in column")

Column:  description
No whitespace found in column
Column:  medical_specialty
No whitespace found in column
Column:  sample_name
No whitespace found in column
Column:  transcription
Whitespace found in column
Column:  keywords
Whitespace found in column
```

Fig. 10 Check for whitespaces in each column

Making a model of user preferences is the next step. One of the most often used techniques in model-based collaborative filtering is matrix factorization. Matrix factorization is the process of identifying two smaller matrices—a user-feature matrix and a feature-item matrix—that, when multiplied together, match the original user-item matrix. These two smaller matrices can be used to predict ratings that are missing. The goal of the factorization is to identify two matrices U and V that have an error between UVT, their product, and the original user-item matrix R. The most frequently applied method for this is Alternating Least Squares (ALS).

The information in the original user-item matrix can be "compressed" into fewer traits through matrix factorization. The user and the item are represented by a vector of features for each item. These features identify the underlying patterns in the data. Users may receive suggestions utilizing the model once it has been constructed. We combine the anticipated rating for each item with the user's feature vector Ui and the item feature vector Vi to provide a suggestion for the user. The product with the highest projected rating is then suggested to the user. The model should be updated as new information becomes available. For instance, the model must be modified to take into account evolving consumer preferences. By enhancing the item matrix while keeping the user matrix constant, the model can be updated when utilizing ALS.

The advantage of model-based collaborative filtering is that it can suggest products to users who haven't yet rated anything to items that haven't yet gotten any user ratings. It's critical to remember that it can be computationally expensive, especially when working with large datasets, and that it can only be utilized with certain types of feedback data. It's crucial to remember that the caliber of the data, the features used to train the model, and the factorization optimization procedure all significantly impact how accurate model-based collaborative filtering is. As a result, it's critical to divide the data into training and testing sets and evaluate the model's performance using metrics like RMSE, MAE, Precision, Recall, and F1-score.

D. Recommender Engine

After cleaning and preprocessing the dataset, the dataset is now ready to be used for our prototype. So, in this prototype we will be using CB recommendation technique. After exploring the data, building the predictive model will be next. The predictive model will allow users to key in keywords according to their symptoms. From here, the system will check the keywords with the database and recommend a medical specialist for the user. Users can prefer to register and use our system's recommendation tool, or they may use it as a guest. If the user is a guest, they will not be able to see their recommended history of our system. In the other way round, if a registered user uses our recommendation tool, they can check on their recommended medical specialist history together with their symptoms or keywords entered to the system previously.

Hence, the predictive model will deal with the keywords that the user keyed into the system. Thus, TF-IDF method and Cosine Similarity will be used. TF-IDF involves tokenizing the text data into individual words, and then calculating the TF-IDF weight for each word. The TF-IDF weight reflects how important a word is to the meaning of the document, with words that frequently appear in the document but not in many other documents considered more important. Once the items have been represented using TF-IDF, the next step is to calculate their similarity. Cosine Similarity is used to compare the TF-IDF representations of the keywords entered by user and find those that are most similar with the keywords in our database. The similarity score ranges between -1 and 1, where 1 indicates that the vectors are identical and -1 indicates that they are completely dissimilar. By comparing the TF-IDF representations of the keywords using Cosine Similarity, the healthcare recommender system can find and recommend those that are most like the keywords with a medical specialty for the user. Finally, the recommender system can be integrated into a user interface, such as deploying it into Streamlit, so that users can easily access and interact with the recommendations.

E. User Interface

The prototype uses VSCode and Streamlit to build. The prototype can be run by typing the following command in the command prompt "streamlit run ProjectDemo.py". The command prompt will direct the user to the streamlit webpage on a browser. Fig. 11 shows that users should understand that this recommender system is not to help them to find a cure for the symptoms entered. Users should be consulting their healthcare provider prior to making any decisions related to their health. The predicted medical specialty should not be used as a substitute for professional medical judgement.

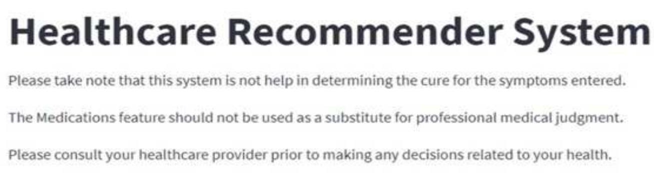


Fig. 11 Healthcare Recommender system launching page.

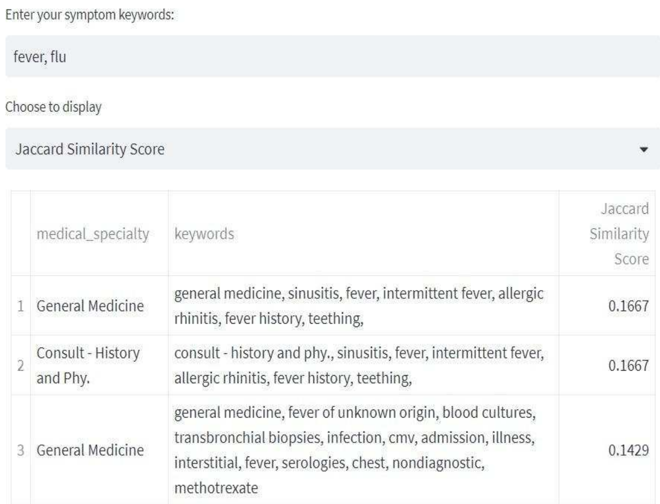
Fig. 12 prompts user to enter their symptom keywords into the system. Users can enter as many keywords as they want to and choose which method, they want the prediction to be conducted in.

Fig. 12 Enter symptom keywords and choose the prediction method.

III. Results and Discussion

Evaluation metrics is an essential tool for measuring the performance of machine learning models and algorithms. They provide a quantitative measure of how well a model or algorithm achieves a specific task or goal. Evaluation metrics can be used to compare different models or algorithms, identify areas for improvement, and track progress over time. A wide variety of evaluation metrics are available, depending on the task or goal of the model or algorithm. Some common evaluation metrics include accuracy, precision, recall, F1 score, and AUC-ROC. Overall, evaluation metrics play a critical role in developing, testing, and improving machine learning models and algorithms.

Fig. 13 shows the predicted output from the entered symptom keywords from users based on Cosine similarity. The user entered 'fever' and 'flu', two symptom keywords. After the user enters the keywords, the user is required to choose which method do they want the prediction to be conducted in. The system will only show the top 3 highest predicted results for users. From here, the user can understand which medical specialty they should consult according to the score displayed. The user can key in as many keywords as they want.


Enter your symptom keywords:

fever, flu

Choose to display

Jaccard Similarity Score

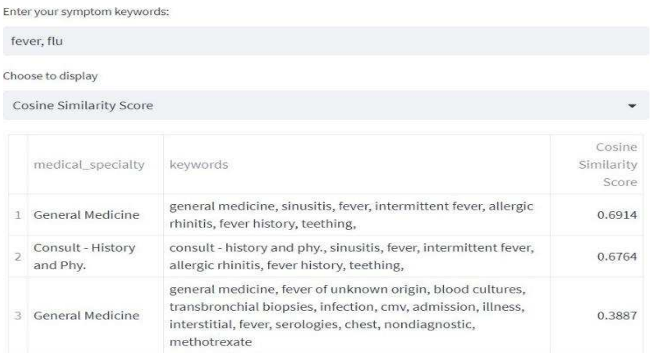
	medical_specialty	keywords	Jaccard Similarity Score
1	General Medicine	general medicine, sinusitis, fever, intermittent fever, allergic rhinitis, fever history, teething,	0.1667
2	Consult - History and Phy.	consult - history and phy., sinusitis, fever, intermittent fever, allergic rhinitis, fever history, teething,	0.1667
3	General Medicine	general medicine, fever of unknown origin, blood cultures, transbronchial biopsies, infection, cmv, admission, illness, interstitial, fever, serologies, chest, nondiagnostic, methotrexate	0.1429

Fig. 13 Predicted output of Cosine Similarity Method

Cosine similarity is a measure of similarity between two non-zero vectors in an inner product space. The cosine of the angle between two vectors is measured. The dot product of the two vectors divided by the product of the Euclidean lengths of the two vectors is used to define the cosine similarity. The result of the cosine similarity algorithm spans from -1 to 1, with -1 denoting orthogonal vectors and 1 denoting identical vector (i.e., vectors pointing in the same direction) (i.e., the vectors are at a 90-degree angle to each other). The degree of similarity between the two vectors is indicated by a value between -1 and 1, with larger values indicating greater similarity. Cosine similarity is useful because it is relatively efficient to compute, invariant to the scale of the input vectors, and considers the vectors' magnitude.

In addition, Fig. 13 also shows the cosine similarity scores of each medical specialty. "General Medicine" and "Consult-History and Phy." are considered very high (0.6914 and 0.6764 respectively), which suggests that these two fields are very similar to each other. On the other hand, the similarity score of the last recommendation which is "General Medicine" is much lower (0.3887), which suggests that these two fields are less suggested. There will be repeated predictions of medical specialty which is normal and unconcerned because the data has different sets of keywords. Hence, the repetition of medical specialty is predicted with different similarity scores. Since this project is using TF-IDF, this vectorization method checks the frequency and the importance of each word present in a keyword column. If there are many keywords in the 'keywords' column that contain the word 'fever' and 'flu' it could mean that the word 'fever' and 'flu' are important. Thus, the score will be higher.

Jaccard similarity is a measure used to quantify the similarity or dissimilarity between two sets. It is based on the size of the intersection of the sets divided by the size of their union. The Jaccard similarity coefficient, also known as the Jaccard index, is a value between 0 and 1, where 0 represents no similarity and 1 represents complete similarity. Fig 14. shows the prediction of Jaccard similarity score for 'fever' and 'flu' symptom keywords. All three of the predictions are relatively low, which are nowhere close to 1. Hence, we can state that this method for prediction is not ideal to be used compared to the Cosine Similarity score given in Fig 13.


Enter your symptom keywords:

fever, flu

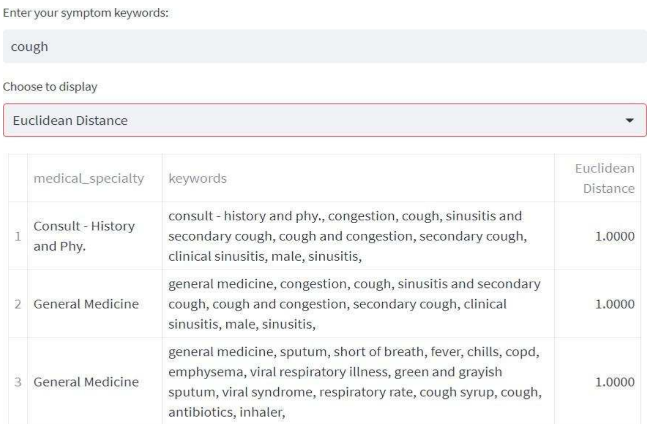
Choose to display

Cosine Similarity Score

	medical_specialty	keywords	Cosine Similarity Score
1	General Medicine	general medicine, sinusitis, fever, intermittent fever, allergic rhinitis, fever history, teething,	0.6914
2	Consult - History and Phy.	consult - history and phy., sinusitis, fever, intermittent fever, allergic rhinitis, fever history, teething,	0.6764
3	General Medicine	general medicine, fever of unknown origin, blood cultures, transbronchial biopsies, infection, cmv, admission, illness, interstitial, fever, serologies, chest, nondiagnostic, methotrexate	0.3887

Fig. 14 Jaccard Similarity Score for 'fever' and 'flu' symptom

In content-based recommendation systems, the Euclidean distance method is a technique used to measure the similarity or dissimilarity between items based on their content features. By calculating the Euclidean distance between the target item and all other items in the system, a list of most similar items can be generated, and these items can be recommended to the user. The smaller the Euclidean distance, the more similar the items are, while a larger distance indicates greater dissimilarity. Fig. 15 shows the result of 'fever' and 'flu' symptom keywords by using Euclidean Distance prediction method. All three predictions given show 1. A Euclidean Distance of 1 suggests that the items are as dissimilar as possible and have no commonality in their feature values. They are located at the opposite ends of the feature space. Hence, we can say that the results are not suggested to be taken into consideration.


Enter your symptom keywords:

cough

Choose to display

Euclidean Distance

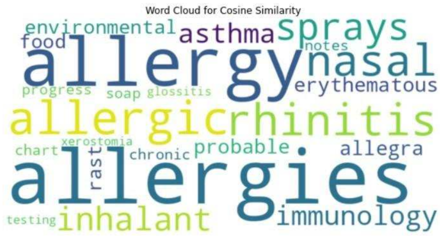
	medical_specialty	keywords	Euclidean Distance
1	Consult - History and Phy.	consult - history and phy., congestion, cough, sinusitis and secondary cough, cough and congestion, secondary cough, clinical sinusitis, male, sinusitis,	1.0000
2	General Medicine	general medicine, congestion, cough, sinusitis and secondary cough, cough and congestion, secondary cough, clinical sinusitis, male, sinusitis,	1.0000
3	General Medicine	general medicine, sputum, short of breath, fever, chills, copd, emphysema, viral respiratory illness, green and grayish sputum, viral syndrome, respiratory rate, cough syrup, cough, antibiotics, inhaler,	1.0000

Fig. 15 Euclidean Distance Score for 'cough' symptom

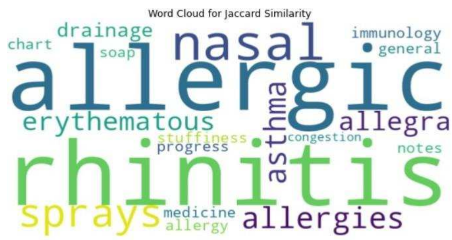
Users can also click the "View History" button to view their predicted history output. The historical output will be visualized in the form of the word cloud. The commonly recommended medical specialty word appears larger with a darker font color, while less frequent or less significant words appear smaller and lighter. This lets users quickly identify the most frequently mentioned or significant words in the text. Fig. 16 shows the visualization of Word Cloud in Cosine Similarity, Jaccard Similarity and Euclidean Distance predictions respectively.

The summary of the evaluation results is depicted in Table II. From the analysis, Cosine Similarity outperforms Jaccard Similarity and Euclidean Distance. The field of healthcare recommendation systems has seen a lot of progress in recent years, as evidenced by the range of approaches described in the related works above. Collaborative filtering techniques have been utilized in various ways, including the use of both item-item and user-user collaborative filtering.

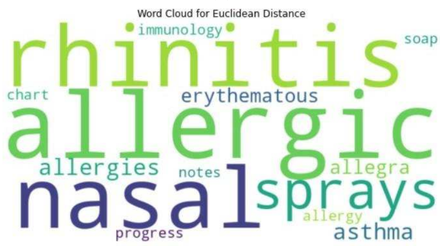
Researchers have also proposed hybrid approaches, which combine multiple techniques to improve the accuracy and coverage of recommendation systems. Additionally, privacy-preserving collaborative filtering has been introduced to protect the privacy of healthcare data while still generating accurate recommendations. The use of sentiment analysis, topic modelling, and matrix factorization has also been suggested to enhance recommendations' personalization. Finally, the application of IoT devices and machine learning techniques to medical recommender systems has shown great promise, although concerns remain regarding the accessibility of hospital data due to privacy concerns. Overall, these approaches show great potential for improving the quality of healthcare services and personalized patient care, and further research in this area is likely to yield significant benefits.



(a)



(b)



(c)

Fig. 16 Word Cloud visualization for (a) Cosine Similarity, (b) Jaccard Similarity, and (c) Euclidean Distance

TABLE II
SUMMARY OF EVALUATION RESULTS

Method Used	**Keywords entered are exactly the same as database score**	**Keywords entered are not same as the database score**
Cosine Similarity	1.0000	0.6933
Jaccard Similarity	1.0000	0.1429
Euclidean Distance	0.0000	1.4142

IV. CONCLUSION

In this paper, various recommender system techniques are introduced and discussed. The summary of these techniques is also presented in a table format for better quick comparison and easy reading that includes each technique's advantages and disadvantages. Some related works on recommender systems are discussed in this paper, and a table summary of related works is presented. According to the related works, we can see that different techniques work differently in different recommender systems. Researchers have used many different evaluation metrics to evaluate the accuracy of their models' accuracy; the most common method is RMSE. In the future, we shall continue with more experimental evaluations to evaluate the performance of our recommender engine compared to state-of-the-art related works. In addition, we will also incorporate user feedback and preferences into the recommendation system to improve accuracy.

REFERENCES

[1] M. Etemadi *et al.*, "A systematic review of healthcare recommender systems: Open issues, challenges, and techniques," *Expert Systems with Applications*, vol. 213. 2023. doi: 10.1016/j.eswa.2022.118823.

[2] V. Sharma and S. Singh Samant, "Health Recommendation System by Using Deep Learning and Fuzzy Technique," *SSRN Electronic Journal*, 2022, doi: 10.2139/ssrn.4157328.

[3] A. del Rio, J. Jimenez, R. Medina-García, C. Lozano-Hernández, F. Alvarez, and J. Serrano, "Improving Quality of Life in Chronic Patients: A Pilot Study on the Effectiveness of a Health Recommender System and Its Usability," *Applied Sciences (Switzerland)*, vol. 13, no. 10, 2023, doi: 10.3390/app13105850.

[4] M. Torres-Ruiz, R. Quintero, G. Guzman, and K. T. Chui, "Healthcare Recommender System Based on Medical Specialties, Patient Profiles, and Geospatial Information," *Sustainability (Switzerland)*, vol. 15, no. 1, 2023, doi: 10.3390/su15010499.

[5] R. Saraswathi, D. Pavithra, S. Gokuldev, I. D. Jebakumar, and K. Tharageswari, "Designing the Prediction Protocol based on Sparrow Search Optimization Algorithm with Vibrational Auto Encoder in E-Healthcare," in *Proceedings - 5th International Conference on Smart Systems and Inventive Technology, ICSSIT 2023*, 2023. doi: 10.1109/ICSSIT55814.2023.10061100.

[6] A. A. Zaveri, R. Mashood, S. Shehmir, M. Parveen, N. Sami, and M. Nazar, "AIRA: An Intelligent Recommendation Agent Application for Movies," *Journal of Informatics and Web Engineering*, vol. 2, no. 2, 2023, doi: 10.33093/jiwe.2023.2.2.6.

[7] S. C. Haw, O. Tahir Yinka, T. T. V. Yap, and S. Subramaniam, "Improving the data access control using blockchain for healthcare domain," *F1000Res*, vol. 10, 2021, doi: 10.12688/f1000research.72890.2.

[8] W.-X. Ong, S.-B. Ho, and C.-H. Tan, "Enhancing Migraine Management System through Weather Forecasting for a Better Daily Life," *Journal of Informatics and Web Engineering*, vol. 2, no. 2, 2023, doi: 10.33093/jiwe.2023.2.2.15.

[9] J.-P. Cheng and S.-C. Haw, "Mental Health Problems Prediction Using Machine Learning Techniques," *International Journal on Robotics, Automation and Sciences*, vol. 5, no. 2, pp. 59–72, Sep. 2023, doi: 10.33093/ijoras.2023.5.2.7.

[10] A. Poulose, A. P. Valappil, and J. Sebastian, "Medication recommender system for healthcare solutions," *Journal of Information and Optimization Sciences*, vol. 43, no. 5, 2022, doi: 10.1080/02522667.2022.2094545.

[11] A. K. Sangaiah, S. Rezaei, A. Javadpour, and W. Zhang, "Explainable AI in big data intelligence of community detection for digitalization e-healthcare services," *Appl Soft Comput*, vol. 136, 2023, doi: 10.1016/j.asoc.2023.110119.

[12] A. Chandra, A. Ahmed, S. Kumar, P. Chand, M. D. Borah, and Z. Hussain, "Content-Based Recommender System for Similar Products in E-Commerce," in *Lecture Notes in Electrical Engineering*, 2022. doi: 10.1007/978-981-19-0019-8_46.

[13] R. J. Kuo and H. R. Cheng, "A content-based recommender system with consideration of repeat purchase behavior," *Appl Soft Comput*, vol. 127, 2022, doi: 10.1016/j.asoc.2022.109361.

[14] Q. Y. Shambour, M. M. Al-Zyoud, A. H. Hussein, and Q. M. Kharma, "A doctor recommender system based on collaborative and content filtering," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 1, 2023, doi: 10.11591/ijece.v13i1.pp884-893.

[15] K. Rrmoku, B. Selimi, and L. Ahmedi, "Provenance and social network analysis for recommender systems: a literature review," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 5. 2022. doi: 10.11591/ijece.v12i5.pp5383-5392.

[16] M. M. Bendouch, F. Frasincar, and T. Robal, "Addressing Scalability Issues in Semantics-Driven Recommender Systems," in *ACM International Conference Proceeding Series*, 2021. doi: 10.1145/3486622.3493963.

[17] G. Behera and N. Nain, "Trade-Off Between Memory and Model-Based Collaborative Filtering Recommender System," 2022. doi: 10.1007/978-981-16-5747-4_12.

[18] K. K. Jena *et al.*, "E-Learning Course Recommender System Using Collaborative Filtering Models," *Electronics (Switzerland)*, vol. 12, no. 1, 2023, doi: 10.3390/electronics12010157.

[19] A. Jeejoe, V. Harishiv, P. Venkatesh, and S. K. B. Sangeetha, "Building a Recommender System Using Collaborative Filtering Algorithms and Analyzing its Performance," in *IoT, Cloud and Data Science*, 2023. doi: 10.4028/p-1h18ig.

[20] G. George and A. M. Lal, "Hy-MOM: Hybrid Recommender System Framework Using Memory-Based and Model-Based Collaborative Filtering Framework," *Cybernetics and Information Technologies*, vol. 22, no. 1, 2022, doi: 10.2478/cait-2022-0009.

[21] R. K. Mishra, J. A. A. Jothi, S. Urolagin, and K. Irani, "Knowledge based topic retrieval for recommendations and tourism promotions," *International Journal of Information Management Data Insights*, vol. 3, no. 1, 2023, doi: 10.1016/j.jjimei.2022.100145.

[22] M. Gao, J. Y. Li, C. H. Chen, Y. Li, J. Zhang, and Z. H. Zhan, "Enhanced Multi-Task Learning and Knowledge Graph-Based Recommender System," *IEEE Trans Knowl Data Eng*, 2023, doi: 10.1109/TKDE.2023.3251897.

[23] A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen, *Knowledge-Based Configuration: From Research to Business Cases*. 2014. doi: 10.1016/C2011-0-69705-4.

[24] Y. F. Lim, S. C. Haw, K. W. Ng, and E. A. Anaam, "Hybrid-based Recommender System for Online Shopping: A Review," *Journal of Engineering Technology and Applied Physics*, vol. 5, no. 1, 2023, doi: 10.33093/jetap.2023.5.1.3.

[25] Abdullah, A. Hussain, S. Ali, H. C. Kim, M. Sain, and S. Aich, "Hybrid Based Model Face Shape Classification Using Ensemble Method for Hairstyle Recommender System," in *Lecture Notes in Networks and Systems*, 2022. doi: 10.1007/978-981-16-9480-6_7.

[26] S. C. Haw, L. J. Chew, and S. Subramaniam, "A hybrid recommender system based on data enrichment on the ontology modelling," *F1000Res*, vol. 10, 2021, doi: 10.12688/f1000research.73060.1.

[27] Q. Han, I. Martinez De Rituerto De Troya, M. Ji, M. Gaur, and L. Zejnilovic, "A Collaborative Filtering Recommender System in Primary Care: Towards a Trusting Patient-Doctor Relationship," in *Proceedings - 2018 IEEE International Conference on Healthcare Informatics, ICHI 2018*, 2018. doi: 10.1109/ICHI.2018.00062.

[28] A. Arshad, M. F. Bin Noordin, R. Bint Othman, and W. Mehmood, "Designing Healthcare Knowledge Management Systems – Development of an Integrated Component-Based KM Model for Effective Governance," 2018. doi: 10.1109/ict4m.2018.00065.

[29] Q. Han, M. Ji, I. Martinez De Rituerto De Troya, M. Gaur, and L. Zejnilovic, "A hybrid recommender system for patient-doctor matchmaking in primary care," in *Proceedings - 2018 IEEE 5th*

International Conference on Data Science and Advanced Analytics, DSAA 2018, 2019. doi: 10.1109/DSAA.2018.00062.

[30] A. K. Sahoo, C. Pradhan, R. K. Barik, and H. Dubey, "DeepReco: Deep learning based health recommender system using collaborative filtering," *Computation*, vol. 7, no. 2, 2019, doi: 10.3390/computation7020025.

[31] F. Jabeen *et al.*, "An IoT based efficient hybrid recommender system for cardiovascular disease," *Peer Peer Netw Appl*, vol. 12, no. 5, 2019, doi: 10.1007/s12083-019-00733-3.

[32] Z. Ren, B. Peng, T. K. Schleyer, and X. Ning, "Hybrid collaborative filtering methods for recommending search terms to clinicians," *J Biomed Inform*, vol. 113, 2021, doi: 10.1016/j.jbi.2020.103635.

[33] M. A. Lambay and S. Pakkir Mohideen, "Big Data Analytics for Healthcare Recommendation Systems," in *2020 International Conference on System, Computation, Automation and Networking, ICSCAN 2020*, 2020. doi: 10.1109/ICSCAN49426.2020.9262304.

[34] F. Rustam *et al.*, "Automated disease diagnosis and precaution recommender system using supervised machine learning," *Multimed Tools Appl*, vol. 81, no. 22, 2022, doi: 10.1007/s11042-022-12897-x.

[35] Y. A. Nanehkaran *et al.*, "Diagnosis of Chronic Diseases Based on Patients' Health Records in IoT Healthcare Using the Recommender System," *Wirel Commun Mob Comput*, vol. 2022, 2022, doi: 10.1155/2022/5663001.

[36] T. Boyle, Medical transcriptions. Kaggle. https://www.kaggle.com/datasets/tboyle10/medicaltranscriptions