Journal of Theoretical and Applied Information Technology
15th April 2020. Vol.98. No 07

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# A STRUCTURED FRAMEWORK FOR BUILDING RECOMMENDER SYSTEM

**MOHAMED GRIDA[1], LAMIAA FAYED[2], MOHAMED HASSAN[3]**

[1] Assistant professor of Industrial Engineering, Faculty of Engineering, Zagazig University, Egypt.

[2] Assistant lecturer, Faculty of Computers and Informatics, Zagazig University, Egypt.

[3] Head of Information System Department, Faculty of Computers and Informatics, Zagazig University, Egypt.

E-mail: [1]mogrida@zu.eg, [2]lamiaa.fayed@yahoo.com, [3]mohmonir372@yahoo.com .

**ABSTRACT**

With the increase of information on the internet, more and more electronic data are appearing. Recommender systems were developed to help customers find related items or personalize services. Several online companies apply recommender systems to build up the relationship with users and enhance marketing and sales. Researchers and managers approve that the recommender system offers great opportunities in various domains. Thus, successful development of recommender systems for real-world applications are significant. The most widely used algorithms for recommender systems are categorized into the traditional recommendation and deep-based recommendation algorithms and hybrid recommendation approaches. There is a vital necessity to understand the recommendation system development way. So, this paper presents a structured framework that helps researchers and practical experts recognizing the development phases of the recommender system. The proposed framework is validated through a case study. Furthermore, this paper introduces a general classification scheme for all current recommendation approaches. A summary of historical past recommender system models provided in a way that facilitates understanding their target.

***Keywords:*** *Recommender System (RS), Deep Learning (DL), Structured Framework, Sparsity, Cold Start, Scalability.*

## 1. INTRODUCTION

With the massive growth of information on the web, the recommender system has become a significant web service. The recommender system assists in overcoming the information and product overload problem. The most successful web sites use recommender systems to improve their relationship with users. Recommender systems (RS) assist the user in finding related information. They aim to provide personalized services to users[1].

In other words, the Recommender system was introduced as a computer program that generates suggestions about new items for a particular user or customer[2]. A recommendation can be products for sale, suggest a movie to watch, match articles to users, and recommend ads on pages, displaying posts on social network news feeds or advising people. The recommendation process is based on predicting the association between the user and the item[3]. This association can be generalized to upcoming user behaviors. Mostly, the association can be categorized as user-to-user associations—the relationship between different users' actions on the same item and item to item association —the relationship between a user's actions on different items. These types of associations are the difficult primary task to make recommendations. Association problem has been handled in various ways in literature based on the kind of information available about items, users and interaction between them.

Over the years, several approaches were introduced for extracting user-user or item-item associations and making a recommendation. Recommendation approaches can be classified into two categories traditional recommendation and deep-based recommendation approaches. Each approach has its advantages and limitations. So, hybrid methods were proposed to alleviate these drawbacks and enhance performance.

Journal of Theoretical and Applied Information Technology
15th April 2020. Vol.98. No 07

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Recommendation techniques are still evolving. Various survey researches on recommender systems were published. However, these studies can be either an analysis of traditional recommendation techniques with their advantages and drawbacks ([1],[4],[5] )or review different models that were developed for a particular domain like tourism [6] and e-commerce [7]. Moreover, a comprehensive survey paper [2] was introduced to analyze traditional recommender system applications which are very significant for researchers and real-world applications. In addition to several recent surveys has been investigated deep-based recommender systems in detail like ([8],[9]).While [10] has been provided a comprehensive review of deep learning-based recommendation approaches and solutions for the challenges of recommender systems. The above-mentioned papers present comprehensive surveys of recommender system. Furthermore, they handle recommendation approaches challenges. However, there is still confusing of how to develop recommender system and the structured way of building.

**The main contributions of this paper are:**

1) We frame different questions at different stages to build a recommender system. The proposed framework provides a useful guide for researchers and practicing professionals on how to build a recommender system.
2) This paper gives a better view of current recommendation approaches where we propose a general classification scheme to categorize the entire existing approaches into three different perspectives (traditional recommendation approaches, deep-based recommendation approaches, and hybrid recommendation approaches). Furthermore, the most dominate representation algorithms are presented.
3) This paper summarizes all data types that may be incorporated in a recommender system to infer user preferences. Furthermore, it presents a comprehensive review of user profile models as it is the backbone of the entire process of the recommender system and has a significant impact on recommendation accuracy.

This paper organized as follows: Section 2 introduces recommender system algorithms. Section 3 presents the proposed recommender systems framework. Section 4 explains the case study. Section 5 present insights and the research findings. Finally, we give our conclusion in section6.

## 2. RECOMMENDATION APPROACHES

Over the years, several approaches introduced for building recommender systems. Recommendation approaches classified into traditional recommender approaches, deep recommender approaches, and hybrid recommender approaches. Figure 1 illustrates the classification of the most widely applied methods. Due to the accuracy and scalability limitations of the traditional recommendation approaches, deep recommendation approaches are proposed to achieve high performance and improve recommendation quality. The hybrid approach integrates deep learning models with a traditional recommendation to tackle recommendation challenges. The next sub-sections briefly investigate the main characteristics of each type.

### 2.1 Traditional Recommendation Approaches

This section explains traditional recommendation approaches. Moreover, the primary representative models from the literature is introduced. Additionally, The main limitations are discussed.

#### 2.1.1 Collaborative filtering approach

The collaborative filtering approach (CF) provides a recommendation based on a user's evaluation of items and evaluations of other users that have similar behavior [2]. Users can explicitly evaluate items using rating scores or implicitly inferred from a user's behavior. Enough rating data available to provide more accurate recommendations [4]. CF predict user's opinion about unseen items using the aggregation function of similar users' rating, e.g., the weighted sum or weighted average. Therefore, CF ranks items and recommend active user with top N-items. CF approach is the most frequently used in large e-commerce sites such as Amazon, Netflix, and Last.fm[8]. Collaborative filtering have two main algorithms: memory-based [13] and model-based ([14],[15],[16] ).

- **Memory-based**

The memory-based algorithm [11] applies statistical methods to discover a set of users that share common preferences to an active user and then recommend items that they like. The memory-based algorithm is easy to implement, scalable for extensive data set, deal with the dynamic database, and it also provides feedback to the users to inform them what features are applied to perform

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

recommendation. However, it suffers from an inaccurate recommendation for sparse data. Rather than, it mainly depends on the user's rating, and the entire database is processed for every time it makes a prediction.

The most famous representative algorithms for memory-based are similarity-based and matrix factorization. **For similarity-based algorithms**, users, and items represented as vectors. A similarity-based vector model is utilized to determine similar users to the active user. Cosine similarity, Pearson correlation, and slope one are the most widely used metrics or algorithms ([12],[13]). Another most popular memory-based algorithm is the nearest neighborhood method [14] to recommend interesting items to a user [15]. The nearest neighbor method has three steps: firstly finding the K-nearest neighbor for the active user. Several similarity measures are used to calculate the similarity between each pair of users or items, i.e., users that share common items have a high similarity score. Secondly, predicting a user's evaluation of nonrating items based on a weighted average combination of the rating of defined neighbors for active users or items. Finally, create a list of top N items with a high prediction score.

**Matrix factorization approaches** utilized in various directions: extract the latent relationship between users and items that help predicting user's ratings on items([16], [17],[18]) or shrink the dimensionality of user-items space([19],[20],[21]). Matrix factorization is the most successful recognition method of latent factor models [22]. Moreover, Matrix factorization builds a predictive model providing a recommendation by inferring latent features from user-item interaction (rating). Recently, it is considered a superior CF algorithm [23] due to accurate recommendations and the ability to handle sparsity problems. However, a predictive model is costly to build.

- **Model-based**

The model-based algorithms**(**[24]**,**[25]**,**[26]**)** apply machine learning and data mining techniques to build an offline model. This model is built and trained by estimating predictions for online CF tasks. It is better to address sparsity and scalability problems and improve prediction performance. However, it is costly to build the model. Representative algorithms categorized into supervised learning and unsupervised learning. **For supervised learning,** the Bayesian network is a popular probabilistic algorithm for learning model-based recommender systems [27]. It handles the recommendation process as a classification where each item represented by a node and its rate correspond to the state ([28],[29]). The Bayesian network is more suitable for applications that the user's preferences change slowly

**For unsupervised learning,** The clustering approach aims to group users based on items that are rating or grouping items based on the users rating them. Prediction of unknown ratings can be either by averaging the rating of the users that belong to the cluster or averaging across the groups. The clustering approach represents each user as partial participation in various classes. Clustering, in some situations, may lead to worse accuracy than the CF approach [28]. It also can be used in the preprocessing phase to narrow user-items space. However, it may reduce recommendation accuracy. So clustering, in most cases, is used as a complementary technique to other methods. The clustering model was applied in several recommender systems ([30],[31],[32]).

#### 2.1.1.1 Collaborative filtering challenges

- **Scalability problem:**

This problem arises with the increasing data set. The computation time will increase by searching for similar users and items and providing a recommendation. Several solutions have been investigated in the literature to alleviate this problem[33].

- **Sparsity problem:**

This problem occurs when there are several empty entries in the user-item matrix. It causes insufficient ratings to make predictions and recommendations. in the literature, several solutions have been proposed to cover this problem ([34],[35],[36],[37],[33],[38]).

- **Gray-sheep user problem:**

This issue occurs when the user has a unique preference. Consequently, it was difficult to interpret their choices, to which user groups he belongs. This issue affects recommendation accuracy and coverage. Several solutions have been presented in the literature to alleviate this problem[31].

### 2.1.2 Cold Start problem:

This problem arises due to insufficient rating and has a significant effect on

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

recommendation accuracy. It can be categorized into three types: new user, a new item, or a new community. There is a significant relationship between cold start and data sparsity. Several proposed solutions introduced in the literature depend on either extend CF with additional external information ([39],[40],[41],[42],[35]) or proposing a new similarity metric ([43],[1]). Furthermore, various reviews published to introduce cold start solutions using traditional RS([33], [44]), and deep learning models [10].

### 2.1.3 Content-based filtering approach

The content-based filtering (CB) approach makes a recommendation of items that are similar to items that users preferred in the past. It also generates recommendations by matching user preference with items contents. It depends only on item contents without considering other users' preferences. So, item characteristics are important where more descriptive items lead to highly accurate recommendations [15]. CB approach compares user interests with items characteristics. Therefore, user interests and the item should have the same presentation composed of the same set of attributes and keywords. The content-based approach has three main components:

1. **The content analyzer** acts as a preprocessing phase to represent the content of the item in a high-level representation to be compared efficiently. This representation form the input for profile learner and filtering component. Item representation methods are ranged from the vector space model to ways that integrate ontologies or encyclopedia knowledge. For instance, Cataldo, M., et al. developed a content-based recommendation (CBRS) framework, which uses textual features extracted from Wikipedia to learn user profiles based on Word Embedding [45].
2. **Profile learner** takes user preferences (likes/dislike data) and generalizes it to build a user profile. A profile can be learned using Bayesian classifiers, clustering, decision trees, and artificial neural networks. A user profile is updated frequently through user feedback (e.g., binary feedback, rating, and comments). The learning process implemented periodically based on new feedback. So, the user profile will be adapted continuously to consider dynamic user preferences.
3. **The filtering component** matches user profiles with item representation to recommend users with their interested items. The result can be a binary or real value computed using similarity metrics like cosine similarity.

Representative algorithms for content filtering approach are categorized to similarity-based, supervised learning and unsupervised learning. ***For the similarity-based***, both user-profiles and items represented as weighted term vectors. Predictions of a user's interest in a particular item can be derived by calculating the similarity between user and item vectors. The items with high prediction value are recommended to the user. Various similarities are introduced in the literature, such as cosine similarity, Singular Value Decomposition (SVD), Latent Semantic Analysis (LSA), Filtered Feature Augmentation. ***The supervised learning* algorithm** learns a prediction function that expects the user favorites based on the content features. Several unsupervised learning algorithms are introduced, such as nearest-neighbors [46], decision trees, rule-based classifiers, Bayesian networks[47], artificial neural networks, support vector machines. ***Unsupervised learning***: find the hidden relationships in unlabeled items. The most common method for unsupervised learning is clustering algorithms.

#### 2.1.3.1 Content-based approach challenges

1. CB algorithm is limited to the content associated with the items
2. **Cold start problem:** CB algorithm should have an accurate description of user preferences to be able to make a recommendation. In the case of new users, the algorithm has no information about their preference.
3. **Overspecialization:** Content-based Approach suggests items that are very similar to items that a user has liked before. Therefore, the user does not get different items that he may like (diversity and novelty).
4. It is very time-consuming.

### 2.1.4 Hybrid approach

Hybrid approaches combine two or more recommendation techniques to cover the problems of individual approaches. Furthermore, it supports improving recommendation accuracy and performance. Burke, R., Classified hybridization methods into seven classes[48]:

1. Weighted: a predicted value of the recommended item is a combination of all

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

predicted value obtained from all recommendation algorithms.
2. Switching: each recommendation algorithm is implemented in certain situations. So, the system will gain the strengths of each algorithm. However, there is a complexity of determining switching criteria.
3. Mixed: The recommendation list is a mix of recommended items from all algorithms.
4. Feature combination: The prediction process relies on a combination of features from different recommendation techniques.
5. Meta-level: another applies the model generated by one algorithm.
6. Feature augmentation: The recommendation output of one technique considered as input to another method.
7. Feature cascade: Another refines the recommendation output of one technique.

Various studies recently introduced deep learning to develop a recommender system. A deep recommender system is based on collaborative or content-based models. It can be designed basically upon the DL algorithm or integrated tightly or loosely with other traditional RS models. Various literature surveys discussed deep learning-based recommender systems ([8],[10][8], [52], [53]). Big data and computational power are the main two factors distinguishing deep learning and present it as a state-of-the-art machine learning technique. The main reasons that encourage applying deep learning

Various studies have introduced hybrid approaches to improve performance. For instance, Barragáns-Martínez, A. B., et al. introduce mix hybridization between content and collaborative algorithm to suggest TV programs[49]. Carrer-Neto, W., et al. presents a hybrid approach between knowledge and social networks-based algorithms[50]. M. A. Ghazanfar, M. A. and Prugel-Bennett, A. proposed a switching hybridization approach to tackle gray-sheep user problems and improve recommendation accuracy[31]. Celdrán, A. H., et al. presented a hybrid recommender algorithm that combines users' locations and preferences and the content of the items located close to such users[51].

## 2.2 Deep recommendation approaches

in recommender system applications listed as follow:
1. Deep learning achieves high impact performance in many forms and improves recommendation quality.
2. Deep learning provides a better understanding of the user's interests and items features in addition to a high-level representation of items characteristics and user-item interactions.
3. DL can solve complex problems and provide state-of-art results.

*Table 2. 1 Strength points of deep learning models in RS.*

| DL technique | Strengths in RS |
|---|---|
| **AE** | • Rating prediction and extraction of latent features.<br>• Dimensionality reduction.<br>• Helpful for covering sparsity and cold-start problems. |
| **CNN** | • Extracting local and global latent features.<br>• Has an impact on performance in processing images, text, and audio. |
| **RNN** | • Suitable for handling the temporal dynamics of ratings and evolving preference over time.<br>• Helpful for processing sequential features in the recommender system. |
| **RBM** | • Handling scalability for large scale data set through reducing user-item preferences dimensionality. |
| **DSSM** | • It maps documents to feature vectors in a latent semantic representation |

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

### 2.2.1 Recommendation based on single deep learning

Some of the deep RS discussed below according to the applied deep learning algorithm. The strengths of each deep model are summarized in table1.

- **Auto-encoder (AE)**

AE is an unsupervised learning approach that utilizes backpropagation to reconstruct the output value from its input data. Auto-encoder is considered as a nonlinear feature extraction technique without using labels of the class[54]. It performs better information representation instead of doing classification. Auto-encoder has various types as Denoising auto-encoder (DAE), sparse auto-encoder, Contractive auto-encoder, variational auto-encoder. Recently, multiple articles have been introduced auto-encoder in the recommender system. Auto-encoder applied in the recommender system into two directions: (1) Predicting rating in the reconstruction layer. (2) Extracting salient features representation in the middle layer. Sedhain, S., et al. applied auto-encoder to develop a collaborative filtering model called autorec [55]. It gets observed user partial vector and item partial vector, projects it at the hidden layer to get low dimension latent space then reconstructs the input in the output layer. The reconstruction part predicts unrated items. An extension of autorec is proposed in [56]. However, it is based on stacked denoising auto- encoder and deals with sparse inputs. Wu. Y., et al. [57] Introduce Collaborative Denoising Auto-Encoder (CDAE) to predict top-N recommendation. The model gets the implicit user's preference as input in a binary form, one of the users like an item and zero for others. This input is corrupted using Gaussian noise. The model is trained on the corrupted input to learn correlations between the user's item preferences.

- **Convolutional neural network (CNN)**

CNN is a feed-forward neural network that is effective for processing data with grid structure topology. It has convolutional and pooling operations, which significantly improve the efficiency of extracting global and local features [8]. CNN is utilized in a recommender system for latent feature extraction from image, text, and audio. for example, Nguyen, H. T. et al. Applied CNN to personalize the tag recommender system where convolution and pooling layers are employed to extract visual features from images[58].

- **Recurrent neural network(RNN)**

RNN is different from a feed-forward neural network where RNN contains memories and loops to model sequential data by sending data over time-steps[52]. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) network are two variants of RNN that have been proposed to handle the vanishing gradient problem. Due to the ability of RNN to integrate temporal and dynamic user preferences, it the in session-based recommender systems, which makes a recommendation based on short term preferences. For instance, Hidasi. B., et al. introduced GRU to build a session-based system based on a history of visited web page and the order of visit in order to enhance recommendation accuracy[59]. Furthermore, Wu. S., et al. develop a session-based recommendation for e-commerce based on user's click histories to predict what the user will buy next[60].

In another direction, RNN is used to study the evolution and co-evolution of items and user's features over time. For instance, Devooght, R. and Bersini. H. uses RNN to study the development of user interests in the recommendation phase keeping a sequence prediction problem[61]. Dai. H. and Wang. Y. Also study users and items latent features effects besides the coevolution of them over time[62].

- **Deep semantic similarity model**

DSSM is a deep neural network modeling approach that projects the input to semantic space. DSSM often have two neural networks. The left network for a user and the other for an item. The input represents user profile and item features. DSSM transforms the input to semantic vectors in a shared semantic space then computes semantic similarity using cosine and rank items according to similarity score to a user profile. DSSM is used to develop a recommender system based on the content model. It usually used to get a top-n recommendation. Huang. P., et al. apply DSSM to develop a deep recommender system in a web search to suggest top-N documents for a given query[63]. A deep-semantic similarity-based personalized recommendation (DSPR) model has been proposed in [64] to present tag-aware recommendation by mapping both tag-based user and item profiles to common tag space. The similarity between user and items are maximized. Elkahky. A., et al. introduced a multi-view deep learning model by learning user and item features from various domains[65].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- **Restricted Boltzmann machine (**RBM)

RBM [66] is an artificial neural model for unsupervised learning. Deep RBM is a stacked RBM in order to enable learning more complex representations of data. To the best of our knowledge, The first deep recommender system is based on a restricted Boltzmann machine that proposed in [67]. Restricted Boltzmann machine is applied in a recommender system to serve various targets: (1) providing low-dimensionality of user's interests. (2) Modeling correlation between users or items. Deng. S., et al. applied RBM to extract latent features from user-profiles and rated items[68]. Moreover, Georgiev. K. introduces a hybrid model of user-based RBM-CF and item-based RBM-CF where hidden layers are connected to an item visible layer and user-visible layer in order to combine the correlation between the users or items[69].

### 2.2.2 The deep composite deep learning model

A deep composite recommender approach combines two or more deep learning models. Different deep algorithms can complement each other and create a more powerful deep composite model. A deep composite model provides a more powerful tool for modeling the features of the user and item. Due to model temporal user preferences challenge, Song, Y. et al. integrates RNN to DSSM, where items static contents are represented at the left network and two-sub networks for long-term static and short-term temporal user interests[70]. It also aims to improve recommendation quality. Besides, Wang, H. et al. introduces a recommender system based on a composite deep learning model that integrates RNN and denoising auto-encoder to model sequence text in item content information[71]. Gao. J. et al. integrates CNN in DSSM hidden layers to develop an interest-aware RS[72]. This composite model makes a significant improvement compared to utilize DSSM solely. Zhang, F. et al. combines CNN and stacked denoising auto-encoder to learn the item's visual representations[73].

## 2.3 Hybrid recommendation approach

The hybrid approach integrates traditional recommender approaches with a deep recommender approach. This approach classified into tightly integrated or loosely integrated based on how tightly approaches are integrated. The former, tightly integrated approach learns and optimizes parameters simultaneously where two integrated approaches are mutually affected each other and allow two- way interaction between them. Otherwise, loosely integration, the parameters are learned separately. Tightly integration can be done at once but it should be designed and optimized carefully. On the contrary, loosely integration can be combined easily but it needs more training steps.

**For the tight integration approach**, deep learning and traditional recommender model are learned simultaneously. There are various studies attempt to combine auto-encoder and collaborative filtering algorithms. The general framework to unify this integration investigated in [37]. It models the mappings between the latent factors used in CF and the latent layers in deep models

Wang. H. et al. introduced a hierarchical Bayesian model called collaborative deep learning (CDL)[38]. CDL integrates stacked denoising autoencoder (SDAE) with probabilistic matrix factorization. CDL is a tightly coupled method for a recommender system that permits two-way interaction between two components. CDL employs generalized Bayesian SDAE for learning feature representation.

Additionally, Wang. H., et al. proposed a tightly integrated model for tag recommendation which is similar to CDL except that authors apply relational information matrix instead of PMF[74]. Rather than Li, X. and J. She, J. replace SDAE deep learning model in CDL with variational auto-encoder which can learn probabilistic variables of content information and combine images and video data sources[75]. Another extension of CDL introduced in [37] where marginalized denoising auto-encoder was applied. It incorporates content information of items, and it is more scalable than CDL.

Similarly, Zhang. S.,et al. introduced a hybrid approach (ConvFM) that integrates the convolution neural network with probabilistic matrix factorization [8]. CNN differentiated by providing high-level representation and accurate contextual information of items through word embedding and convolutional kernels. Furthermore, Zheng, L., et al. join a convolution neural network with matrix factorization[76]. The model consists of two parallel CNN connected at the final layer with matrix factorization to extract user-item interactions for predicting rate. It handles sparsity and provides an excellent semantic representation of review text.

**For the loose integration approach**, the deep learning model is applied to learn latent feature representation and then provides them with traditional recommender systems. For instance,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Barragans-Martinez, B., et al. integrates contractive auto-encoder with SVD++[77]. A contractive auto-encoder captures several input variations, and it can learn the implicit feedback to improve accuracy. Kim, D., et al. proposed a hybrid recommender system that combines stacked denoising auto-encoder and timeSVD++ [78]. It is an attempt to solve the item cold-start problem.

Some articles have tried to combine CNN with a traditional recommender system loosely. For example, Abdillah. O. and Adriani. M. propose a hybrid recommender approach to suggest restaurant mainly based on visual information like food images and restaurant furnishings[79]. CNN is utilized to extract visual features and high-level representation of text descriptions. Then, it feeds them to probabilistic matrix factorization. Lika, B., et al. introduced a hybrid approach for the e-learning recommender system[80]. Researchers firstly extract item features from item content such as introduction and content of materials and then incorporate them into weight matrix factorization. Generally, modeling items content participate in alleviating the cold-start item.

## 3. Recommendation Systems Framework

We present here a structured framework that consists of four RS phases that help a developer how to build a recommender system as shown in Figure2. Each stage has several forms to apply. All of that described as following:

### 3.1 Data Collection Phase

Data collection refers to gather user preferences. There are two forms of data collection, explicitly by asking users to rate an item or implicitly by tracking users' behavior. Explicit data provide an accurate recommendation. However, it is time-consuming and very tedious for the user. While implicit suffers from a cold start challenge where the user hasn't enough history.

#### 3.1.1 Data type

Figure3 summarizes all data types that may be incorporated in a recommender system to infer user preferences. These data collected from various data sources, including online users' behavior, extracted from log files, their own generated content or their social interactions data from the social network. It classified as explicit or implicit regarding how to get user preferences

User preference can be defined explicitly through rating or voting. In [81], initial preferences are provided about movies by asking users what they like about genres of movies. Hence, preferences are updated by explicit user vote or implicitly inferred, i.e., if the user has watched the movies entirely then, he must have liked it. Additionally, users can provide their preferences through user feedback like review ([82],[76], [83]).

Demographic data refers to personal characteristics such as age, gender, education, abilities, and experience. Users with the same demographic data have the same behavior and preferences. Lika, B., et al. Infer user preference for new users through the identification of the other users who have the same demographic data[84].

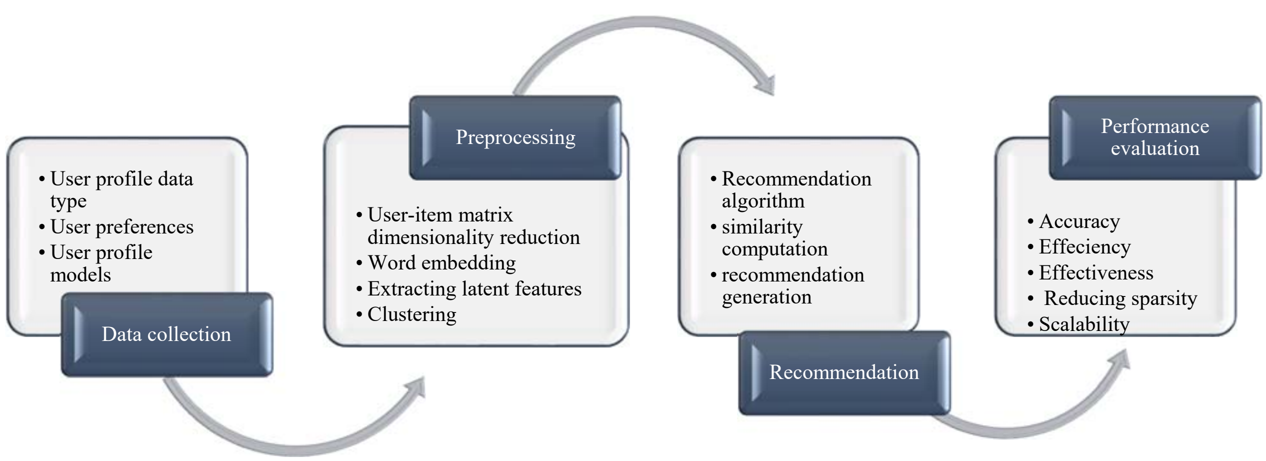


*Figure 2. Main Phases of Recommender System Framework.*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Furthermore, User preferences can be learned from the user context. The user's context is defined as any information that can be used to characterize the user's situation, such as time, geospatial data, or group of related people such as friends or family members. For instance, Kawashima, H., et al. implicitly infer user degree of interest for each product according to the distance between users and products [81]. The position of products and users was determined using RFID. In [85] implicitly estimates user preference for a brand store by averaging time spent in a store, frequency of entering the store and promotional offers factors. New position approach called Received Signals Strength (RSS) pattern mining position method. In [86], Time-aware filtering technique was applied to get users' preferences by considering the change of preferences over time.

Recently, User preferences are learned from social interaction with other users, such as online friending, posts, comments, and tags. In [87] captures user interest by analyzing user tag information. The research considers the frequency, duration, and recency of social tags to face dynamic changes of the user's attention with time. In [88] incorporates social network information such as user's friendships and rating record (tags) to predict the missed user's preference.

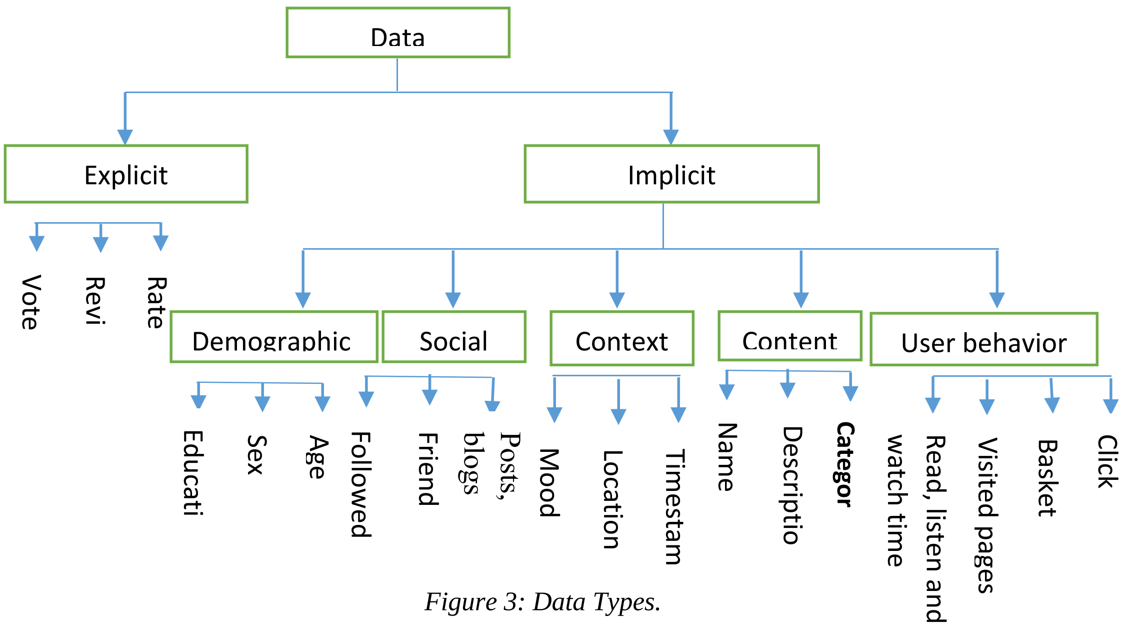


*Figure 3: Data Types.*

Other studies attempt to combine two or more of the data mentioned above sources to obtain accurate user preferences and intentions. For instance, Costa-Montenegro, E., et al. implicitly detect preferred mobile applications based on a combination of historical consumed applications, use patterns, and previous tagging of applications and history of ratings[15].

### 3.1.2 User profile

User profiles are a data structure used to store the user's characteristics and data. They are considered a key component of most recommendation systems (RS). A user profile can incorporate all types of data described above. Various studies have introduced user profile modeling because it is the backbone of the entire process of the recommender system. There are different user profile models in the literature. Each has its advantages and disadvantages. Most frequently used models reviewed below:

### A. Keyword Profile Model

The user profile represented as a set of preferred keywords or categories that can be extracted or directly provided by users. Keyword weight is a numerical representation of a user's level of interest and reflects how important it is. A model generalizes preference that may lead to an inaccurate recommendation. However, it maintains privacy and does not navigate user personal characteristics. It is a simple representation model. Personalized search engines are the most suitable applications for this model. Diederich, J., et al. created the keyword user

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

profile using tags associated with objects[89]. A recommender system is developed in the research domain to suggest publications, keywords, and persons, based on a proposed tag based-profile. Lengsfeld. C.S. and Shoureshi. R.A. present a web page recommender system based on a keyword user profile[90]. Keywords are extracted from a web page that the user was browsed. Shmueli-Scheuer, M. et al. represent user profiles through the extracted keywords of the documents associated with the user over time[91]. Bhattacharyya, P., et al. construct keyword user profile to present exciting fields in the online social network[92]. Similar users are determined according to the proposed model that matches keywords based on a semantic relationship.

### B. Vector Profile Model

This approach models user profile as a vector of numeric values that represent a user's degree of interest corresponding to each attribute of the item. It also can add other user's characteristics as demographic information. This model considered the most widely used. It provides an accurate recommendation based on the excellent description of items. Kim, J.K. and Y. H. Cho, Y.H. defined the customer profile as a vector of ratings corresponding to products[93]. Ratings are inferred implicitly from shopping process activities like several click-throughs, basket placement, and purchase. Yang, W. S. and Hwang, S.Y. proposed a travel recommender system in mobile that creates the user's profiles as a vector of ratings of their visited attractions[94]. Barragans Martinez, B., et al. created a user profile for movie preferences using a rating vector that can be provided by the user or implicitly inferred from viewing history[81].

### C. Semantic Profile Model

Ontology is the way to introduce semantics in a user profile. An ontology represents the user profile as a set of concepts within a domain and the relationships between these concepts. Ontology is emerging as a natural choice for the next generation user profiles due to powerful knowledge representation and associated inference mechanisms [95]. Some studies [96], [97] model a generic user profile that presents the primary static user's information. While others[98] combine personal and contextual information such as time and location. Other studies learn user profiles from the domain context [95]. The semantic profile model applied in various domains, including personalized web search engines [99], information retrieval[100], tourism [101] and e-commerce ([86], [102]).

### D. Uncertainty Profile Model

Uncertainty models developed to handle imprecise preference information stored in the profile. Implicit rating inferred from online user's behavior. It is considered an uncertain value. The user profile has a confidence degree associated with each rate and recognized as a weighting factor in the exploitation stage. IF-sets rules [103]and fuzzy ontology [104] used to model vague and imprecise preferences to help to recommend a product that fits the best to user expectations. The main advantage of this model is to reason about incomplete and uncertain knowledge of the user's behavior.

### E. Probabilistic Profile Model

The probabilistic user profiling model presents user preferences as a probability. User preferences are given as a probability that can be learned or manually entered for objects [105]. They learned dynamically based on the user's relevance feedback on the object by using "I like" or "I dislike". Yin, H., et al. modeled a user profile using a location-aware probabilistic generative model, LA-LDA, considering the three important location-based observations[106].

### F. Histogram profile Model

Histogram profile model introduces the profile as a histogram of relative frequencies where the information denoted as an arrangement of independent samples of predefined categorized data (a probability mass function) to protect privacy. It represents the general concepts associated with weight, frequencies of user's use. For instance, in the news recommender system, a predefined set of topics is determined then the user profile is modeled by the histogram of the distribution of user's clicks on each news topic category[107]. Furthermore, for a mobility user profile where the user profile is represented as the probability distribution of each location to the set of visited locations [108]. This type of user profile model usually introduced in content-based recommender systems [109] and recently, researchers propose histogram profile model in when preserving privacy in recommendation system([110], [111],[112],[113]).

## 3.2 Pre-Processing Phase

The pre-processing step has several forms in the recommender system. It can be in the form of dimensionality reduction, clustering, and content representation. For dimensionality reduction,

Journal of Theoretical and Applied Information Technology
15th April 2020. Vol.98. No 07


ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Recommendation systems use generalization approaches to reduce user-item matrix dimensionality space. Dimensionality reduction results are so directly applicable to the computation of the predicted value. Generalization approaches are like Singular Value Decomposition (SVD) ([20], [21], [49])and Principal Component Analysis (PCA)[114]. These approaches are now considered as an approach to RS design, rather than a preprocessing technique.

The clustering approach can be used in the preprocessing phase to narrow user-items space ([30],[115]). Additionally, the preprocessing step is beneficial for analyze item content to extract features like concepts, n-grams, or keywords using feature extraction approaches and information retrieval algorithms[116]. Moreover, a word embedding technique can also be incorporated to map texts into a low-dimensional semantic space keeping the word sequences information[82].

### 3.3 Recommendation Phase

User profile and item content form the input to the recommendation phase. This phase contains three steps: recommendation algorithm selection, similarity computation, and recommendation generation described as following.

#### 3.3.1 Recommendation algorithms selection

Recommendation approaches are classified into two classes: traditional and deep-based recommendation algorithms. These algorithms are investigated in detail in section2 above.

#### 3.3.2 Similarity computation

The quality of recommendation obtained from the CF approach depends on the similarity between users or items. There are some ways to determine the similarity between users or items. The most popular metrics are Pearson correlation, cosine, adjusted cosine, constrained correlation (CCORR), Mean Squared Differences and Euclidean. However, there are new measures that can improve these metric results especially in cold start problems and increase prediction quality [1]. For example, Jaccard Mean Squared Differences (JMSD). the most common similarity metrics are described as follows.

- **Pearson Correlation**

Pearson correlation metric calculates the similarity between users based on the overlap of rating items between an active user and the neighbors. Pearson correlation provides the best prediction and recommendation result over the other metrics in a collaborative filtering approach[5] and [1]. Equation (1) represents the formula for the Pearson correlation between user x and neighbor y, where $S_{xy}$ denotes the set of co-rated items between x and y, $r_{\bar{x}}$ is the average rating of the user x and $r_{\bar{y}}$ is the average rating of the user y.

$$sim(x,y) = \frac{\sum_{s\in S_{xy}}(r_{x,s}-r_{\bar{x}})(r_{y,s}-r_{\bar{y}})}{\sqrt{\sum_{s\in S_{xy}}(r_{x,s}-r_{\bar{x}})^2}\sqrt{\sum_{s\in S_{xy}}(r_{y,s}-r_{\bar{y}})^2}}$$
(1)

- **Cosine-based similarity**

Users or items represented as vectors in N-dimensional space, e.g., user space. The similarity between two users is computed by the cosine of the angle between two vectors. Equation (2) refers to the formula of cosine between user x and y.

$$sim(x,y) = \cos(x,y) = \frac{x.y}{\|x\|\|y\|}$$
(2)

- **Conditional Probability-based Similarity**

A measure that is based on the conditional probability of liking (or rating) an item given that the user already showed his interest for another item. If an item i has a good chance of being purchased after an item j was purchased then i and j are similar. The similarity is given by equation (3) where α is a factor dependent on the problem.

$$sim(i,j) = p(i|j) \times \alpha \quad (3)$$

#### 3.3.3 Recommendation Generation

There mainly exists two types of recommendation generation based on the output forms: rating prediction and ranking prediction (top-n recommendation).

- **Rating prediction**

The recommender system can build a model using one of the similarity measures described above. It predicts the rating for any user-item pair can be estimated by aggregation function. For example, $user_1$ evaluates $item_2$ and $item_3$, and we want to predict $user_1$ would prefer $item_1$ or not i.e., $r_{11}$. An unknown rating is usually predicted using an aggregate of the ratings of a similar user who like the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

same items ($user_2$ and $user_3$). Some examples of aggregation function [4]:

a) $r_{c,s} = \frac{1}{N}\sum_{c' \in C} r_{c',s}$ (4)

b) $r_{c,s} = k \sum_{c' \in C} sim(c,c') \times r_{c',s}$ (5)

c) $r_{c,s} = \bar{r_c} + k \sum_{c' \in C} sim(c,c') \times \left(r_{c',s} - \bar{r_{c'}}\right)$, (6)

Where $r_{c,s}$ is the unknown rating for user c and item s, N most similar users for the same item s, multiplier k serves as a normalizing factor and is usually selected as

$k = \frac{1}{\sum_{c' \in C}|sim(c,c')|}$, and where the average rating of user c,$\bar{r_c}$ is defined as

$\bar{r_c} = (1/|S_c|)\sum_{s \in S_c} r_{c,s}$ , where

$S_c = \{s \in S | r_{c,s} \neq \emptyset\}$

- **Ranking prediction**

Ranking prediction aims to predict the top n items and produces a ranked list according to its similarity with a user profile or item features or both of them. There are several ranking procedures, such as pairwise[117] and point-wise[118] [119].

### 3.4 Performance Evaluation Phase

In the literature, researchers use various evaluation metrics for evaluating the performance of the developed recommender system. Recommender systems should make the suggestions more accurate, efficient, effective, and scalable, and the system should deal with sparse data and dynamic databases. [52] categorizes evaluation metrics based on the output form which is either rating or ranking prediction. For rating prediction, we can use Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). For ranking prediction, information retrieval metrics are applied such as recall, precision, and F1 score. All recommendation evaluation metrics discussed below:

- **Accuracy**

This is measured by how close the result of a recommendation matches a user's preference [1]. There are two measures for evaluating the accuracy of a recommender system, statistical accuracy metric **(**One of the widely used metrics is the Mean Absolute Error (**MAE**), the lower the value of MAE the more accurate the result is.) and decision support accuracy**:** Mean Squared Error (***MSE***), Root Mean Squared Error (***RMSE***).

$$\boldsymbol{MSE} = \frac{1}{n}\sum_{i=1}^{n}(p_i - r_i)^2 \quad (7)$$

$$\boldsymbol{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - r_i)^2} \quad (8)$$

$$\boldsymbol{MAE} = \frac{1}{n}\sum_{i=1}^{n}|p_i - r_i| \quad (9)$$

n: unrated items used for the test.
$P_i$ the prediction on the ith test instance.
$r_i$ the corresponding rating value is given by the user.

- **The Coverage metric:** a metric that estimates a percentage of items that the recommender system can able to predict rating[52].

$$\boldsymbol{Coverage} = 100 \times \frac{u}{U}, \quad (10)$$

where U potential users n items number, u: users for whom the recommender system was able to generate a recommendation lists

- **The Stability quality metric:** a metric that reflects if the recommender system is stable over time[4].

$$\boldsymbol{Stability} = \frac{1}{P_2}\sum_{i \in P_2}\left|P_{2,i} - P_{1,i}\right|, \quad (11)$$

where P1: old prediction, P2: a new prediction

- **The Novelty quality metric:** a metric that estimates the difference level between the recommended items and interesting items.

$$\boldsymbol{Novelty} = \sum_{i \in L}\frac{\log_2 P_i}{n} \text{ where } P_i = \frac{n - rank_i}{n-1} \quad (12)$$

*Where,* L recommendation list.

- **The Diversity metric:** a metric that measures the degree of the difference among suggested items.

$$\boldsymbol{Diversity} = \frac{a}{c}\sum_{i=1}^{c}\frac{1}{n}\sum_{j=1}^{n} i_j \quad (13)$$

Where C content vector related to each item having a length c.

- **Efficiency**

Memory and Computation time are two important metrics that evaluate the efficiency of a recommender system.

- **Effectiveness(quality) for top-n recommendation**

Precision and recall metrics are used to evaluate the quality of the recommendation set. Precision is the ratio of the recommended relevant items over all recommended items. A recall is the ratio of the recommended relevant items over all related items. F-measure is a weighted average of precision and recall. Receiver Operating Characteristic (ROC) is a graphical technique that uses two metrics: TPR (True Positive Rate) and FPR (False Positive Rate)

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (14)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (15)$$

$$\text{F1}_{\text{Score}} = \frac{2\times\text{precision}\times\text{Recall}}{\text{Precision+Recall}} \quad (16)$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP+FN}} \quad (17)$$

$$\text{FBR} = \frac{\text{FP}}{\text{FP+TN}} \quad (18)$$

Where, TP: true positive an interesting item is recommended to the user
TN: True Negative: an uninteresting item is not recommended to the user
FN: False Negative: an interesting item is not recommended to the user
FP: False Positive: an uninteresting item is recommended to the user.

- **Scalability**

Scalability is the ability to manage the vast increase in the number of users and items.

- **Handle Sparsity**

Sparsity means that most of the users only rated a small portion of the total available items. The sparsity challenge may make the recommender algorithm very unreliable because of insufficient information processed. Because of that system should cover this challenge to obtain good performance.

- **Real-Time**

Real-time measures how the recommender system generates new recommendations when the user has new behaviors immediately.

- **Deal with Dynamic database**

Dynamic database means new items or users arrive once a model is built [120].

## 4. Case study

We build an effective real recommender system application for 2B Egypt Company to help its customers find products to purchase. It practically follows the proposed framework. In data collection phase, user profile model is created by monitoring daily user activities while using a mobile device to collect the user's interests. RS employs content-based recommendation approach that comparing the user profile with the description of the items.

The recommendation phase consists of three steps: content analysis, similarity computation, and recommendation generation. Firstly, the content analyzer defines the underlying set of topics in the content of items to be compared with the user profile efficiently. Secondly, the similarity between the item/user profiles computed. Finally, recommendations generated where we decide to recommend an item or not. So, Items with high similarity with the user's interests are carefully chosen and presented to the user. Precision and recall metrics are used to evaluate the quality of the generated recommendation set.

## 5. Insights and Discussions

In this section, we discuss our set of findings and inferences and present insights based on the overall analysis on recommender systems development phases.

- Deep learning models are widely employed for latent features extraction from items content and auxiliary information and then incorporated into the recommendation process. Autoencoder and CNN models are the most commonly used for extracting latent features to improve recommendation accuracy and handle sparse data.
- The recommender system should incorporate two or more of the data types from various data sources to obtain accurate user preferences and intentions.
- The user profile is considered as a vital component of the entire recommendation process. The most commonly used is the vector profile that modeled as vector of numeric values that represent ratings corresponding to products. Ratings can be collected explicitly or inferred implicitly from shopping process activities.
- User profile incorporates sensitive information about the user, such as demographic characteristics, physical location, and preferences. Histogram profile models user's data as an arrangement of independent samples of predefined categorized data to protect user privacy.
- All recommendation approaches focus on enhancing rating prediction accuracy. Mean-squared error (MSE), mean absolute error (MAE), and RMSE for prediction accuracy.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Precision, recall, F1-measure, and receiver operating characteristic (ROC) for quality for top-n recommendation.

- Generalization approaches and deep learning model are also used as a preprocessing approach in recommender systems. Their purpose is to handle large-scale datasets by reducing dimensionality.
- We should think about the performance and quality metric and try to put specific performance priorities like scalability instead of accuracy. Therefore, a technique that combines features from many approaches should have an effective recommendation.

## 6. CONCLUSION

Recommender systems are powerful systems that give added value to business and corporation. They are relatively recent technology and they will only keep improving in the future. However, all of current articles provide a comprehensive review of current recommendation approaches, there is no article introduce structured framework for building recommendation system. Our research aims to frame all the questions into a set of structured phases. In conclusion, for the other researcher would use the proposed framework when trying to build a new recommender system. The overall performance of the recommendation system depends on a combination of the methods used in data collection, preparation and recommendation phases. The proposed framework was validated the presented case study.

## 7. REFERENCES

[1] A. G. J. Bobadilla , F. Ortega, A. Hernando, "Recommender systems survey," *Knowledge-Based Syst.*, vol. 46, pp. 109–132, 2013.

[2] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decis. Support Syst.*, vol. 74, pp. 12–32, 2015.

[3] C. Du, C. Li, Y. Zheng, J. Zhu, and B. Zhang, "Collaborative Filtering with User-Item Co-Autoregressive Models," *Thirty-Second AAAI Conf. Artif. Intell.*, pp. 2175–2182, 2018.

[4] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005.

[5] D. Almazro, G. Shahatah, L. Albdulkarim, M. Kherees, R. Martinez, and W. Nzoukou, "A Survey Paper on Recommender Systems," *Comput. Res. Repos. - CORR*, vol. abs/1006.5, no. 5, pp. 129–151, 2010.

[6] J. Borràs, A. Moreno, and A. Valls, "Intelligent tourism recommender systems : a survey," *Expert Syst. Appl.*, 2014.

[7] K. Wei, J. Huang, and S. Fu, "A survey of E-commerce recommender systems," *Proceedings - ICSSSM'07: 2007 International Conference on Service Systems and Service Management*, no. 2006, 2007.

[8] S. Zhang, L. Yao, and A. Sun, "Deep Learning based Recommender System: A Survey and New Perspectives," vol. 1, no. 1, pp. 1–35, 2017.

[9] B. T. Betru and C. A. Onana, "Deep Learning Methods on Recommender System : A Survey of State-of-the-art," vol. 162, no. 10, pp. 17–22, 2017.

[10] Z. Batmaz, A. Yurekli, A. Bilge, and C. Kaleli, "A review on deep learning for recommender systems : challenges and remedies," *Artif. Intell. Rev.*, 2018.

[11] A. Bellogín, P. Castells, and I. Cantador, "Improving memory-based collaborative filtering by neighbour selection based on user preference overlap," *Open Res. Areas Inf. Retr.*, no. 2, pp. 145–148, 2013.

[12] F. Ricci, L. Rokach, B. Shapira, P. B. Kantor, and F. Ricci, *Recommender Systems Handbook 123*. 2011.

[13] M. Xu, S. Berkovsky, S. Ardon, S. Triukose, A. Mahanti, and I. Koprinska, "Catch-up TV Recommendations : Show Old Favourites and Find New Ones," pp. 285–293, 2013.

[14] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "an algorithmic framework for performing collaborative filtering," 1999.

[15] E. Costa-Montenegro, A. B. Barrag??ns-Mart??nez, and M. Rey-L??pez, "Which App? A recommender system of applications in markets: Implementation of the service for monitoring users' interaction," *Expert Syst. Appl.*, vol. 39, no. 10, pp. 9367–9375, 2012.

[16] B. M. Sarwar, G. Karypis, J. a Konstan, and J. T. Riedl, "Application of Dimensionality Reduction in Recommender System -- A