IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 2, 2021, accepted October 8, 2021, date of publication November 8, 2021, date of current version November 12, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3124939*

# Integration of Linked Open Data in Collaborative Group Recommender Systems

**ROSMAMALMI MAT NAWI, SHAHRUL AZMAN MOHD NOAH, (Member, IEEE), AND LAILATUL QADRI ZAKARIA**

Center for Artificial Intelligence Technology, The National University of Malaysia, Bangi 43600, Malaysia

Corresponding author: Shahrul Azman Mohd Noah (shahrul@ukm.edu.my)

This work was supported in part by the Ministry of Higher Education under Grant FRGS/1/2020/ICT02/UKM/01/1.

**ABSTRACT** A group recommender system (GRS) is a system that collectively recommends items to a group of users based on their preferences. The GRS and the individual RS challenge lies in a very small and incompleteness of user-item ratings. Such incompleteness resulted in the data sparsity problem. The issues of data sparsity in a group negatively affect the quality of recommendations to the group. It occurs due to the inefficient formation of groups, which usually involves individuals with sparse data in their user profiles. Most of the current studies focus on this issue after the formation of groups. However, this study focused before the group formation, based on the intuition that it will be more efficient if the data sparsity at the individual level is addressed before the group formation process takes place. Therefore, applying the approach through Linked Open Data (LOD) technology is proposed to ensure that the data sparsity issues can be overcome before the group formation process is implemented. We proposed a GRS-LOD model. The experimental evaluations relating to the prediction accuracy and recommendation relevancy of the proposed model were implemented on three aspects: comparison with the basic approach or baselines; comparison with the current approaches, and comparison in terms of group size and aggregation strategies. The aggregation strategies used were the Average (AV), Most Pleasure (MP), Average without Misery (AVM), and Least Misery (LM). The metrics for prediction accuracy were based on the RMSE and MAE, whereas for relevancy, precision, recall, and F1-score were considered. The results show that the prediction accuracy and relevancy of the developed model's recommendations is better than the baseline study by adapting the Average (AV) strategy with the individual profile aggregation approach. Meanwhile, for the evaluation in terms of group size, the results show larger group size exhibits better prediction accuracy for the four used aggregation strategies. On the other hand, in terms of recommendation relevancy, the result shows that relevancy decreases with the increase in group size for the MP, AV and AVM strategies.

**INDEX TERMS** Group recommender system, linked open data, clustering, k-nearest neighbour.

## I. INTRODUCTION

A new search paradigm is emerging, in which users' perspectives on information searching have shifted from searching for information to receiving information. One of the new approaches for receiving information is via recommender systems (RSs). RSs have become an important tool for addressing information overload problems and proved to be successful in many classical domains such as movies, books, and music. Usually, such recommendations are made to individuals by adapting to the user's characteristics and preferences. Few recommendation techniques are currently in existence. However, the collaborative filtering techniques, which filter out items that a user might like according to ratings by similar users are most widely used.

RSs for groups of users are gaining attention as several information needs arise from the group and social activities, such as listening to music, watching movies, traveling, and attending sporting events [1]. Furthermore, as stated by Felfernig *et al.* [2], compared to conventional RSs, relatively a group recommender system (GRSs) is still a new field with few successful commercial applications being reported. Research that focuses on the recommendation to a group of users although is still limited [3], but the recent trend has seen a great demand for such applications of recommendations.

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

The approaches of group recommendation generally follow a three-step process. It first starts with the formation of groups based on the identification of similar users (represented by the user profiles) as group members. It then proceeds with the group modeling which mainly concerns representing the features and characteristics of the groups. Aggregation of the preferences of group members is of importance at this step. The three-step procedure concludes with the prediction of unrated items for the group.

A common problem faced by RSs is the small number of ratings provided by users to items [4], [5]. This situation refers to the issue of data sparsity. Grouping or clustering is one form of classification affected by the subject of data sparsity [6]. While various efforts have been put forward to overcome the data sparsity issues in RSs, its impact on GRSs is still a major concern. GRSs still suffers from providing reliable recommendations when the data is sparse. As stated by Wang *et al.* [7], generating useful recommendations will be more difficult when a lot of data is not rated. In fact, according to Boratto and Carta [8], the results of their study found that the data sparsity problem greatly influences the process of clustering and thus, may have negative consequences in the formation of a group in GRSs. Effective and reliable clustering methods will assist in generating recommendations more efficiently for the group of users.

Current research works on GRS addressed the data sparsity issues after the formation of groups [9]; meaning that the data sparsity problem is addressed between groups. However, we hypothesized that it will be more effective if the data sparsity problem is addressed before the formation of groups, which is at the individual level of users. While many approaches for ameliorating the data sparsity problems have been proposed, such as the recursive filtering approaches [10] and the data imputation approaches [11] little work has been put forward for GRSs.

The Linked Open Data (LOD) initiative has been quite successful in terms of publishing and interlinking data on the Web. On top of the huge amount of interconnected data, measuring relatedness between resources and identifying their relatedness could be used for various applications such as the RSs. However, the exploitation of LOD for collaborative recommender systems is still very few, let alone GRS. Thus, exploiting LOD technology on the profiling of users in such a way that the sparseness data can be minimized and improved the formation of groups in GRS is desirable.

LOD can obtain item similarities by linking to publicly accessible external sources, such as DBpedia. It brings semantic relationships to existing datasets and enriches item information. This augmented information can then be exploited to accumulate new and relevant information in the group formation process. Furthermore, research has shown that using LOD to find other relevant items not represented in the dataset yielded positive results [12].

In this paper, we proposed to enhance the effectiveness of GRSs by ameliorating the sparseness of user-item ratings through the exploitation of the knowledge structure represented in LOD. In this case, DBpedia which is the most famous knowledge source in the LOD cloud is being considered and used during the step of forming a group. To our knowledge, there aren't any studies that integrate LOD in a group-based recommender system, specifically to overcome the issue of sparsity during group formation.

The contributions of this paper are summarized as follows:

1) We use the LOD technology in GRSs to ameliorate the data sparsity problems, thus introducing the GRS-LOD model.
2) We illustrate that the use of LOD during the formation of groups improves the quality of the recommendations among groups.
3) We showed that group sizes and aggregation strategies have an impact on the effectiveness of GRSs.

The remainder of the paper is organized as follows. Section 2 presents related work on group recommendation approaches. Section 3 defines the details of the approach of the framework model. While section 4 describes the evaluation criteria for the experiments. We discuss the result and outcome in section 5. And finally, we conclude the paper in Section 6.

## II. RELATED STUDIES

We discuss some related studies of LOD-enable RS as well as for GRS in this section.

### A. GROUP RECOMMENDER SYSTEM

Recommender systems are a type of information filtering tool that aims to provide suggestions for items to be of use to the user. Such suggestions can relate to different decision-making processes, such as what users to connect to in a social network, what items to buy, which services to commit, what music to listen to, or what movie to watch [13]. RS can provide different users with various services to meet individual needs [14]. The recommendation methods are usually divided into three categories: collaborative filtering recommendation (CF), content-based recommendation (CB), and hybrid approaches (HAs). CF approaches can be divided into user-based, item-based, and model-based methods [15].

The CF methods use similarity among users to make recommendations. Those methods belong to the CB find items that are mostly similar to the items that the user liked in the past. HAs combine CF and CB methods, which can help avoid certain limitations of CF and CB. The CF technique is one of the most successful techniques used by RS, which filters information by exploiting the recommendation of other similar users [16]. However, CF suffers from the drawbacks related to cold start problems, scalability, and data sparsity [17].

In recent years, a new set of RS arises to cope with services or products that users consume collectively [18]. Domains such as movies, restaurants, and tourisms tend to be used more frequently by more than one user with particular preferences. According to Xu *et al.* [3], recent years has seen an increase in the challenge of providing recommendations to groups. GRS have attracted significant research efforts for their importance in benefiting group members. They suggest items to a group of people engaged in a group activity. For example recommending a movie to several friends [19], similar patients may be recommended for equal treatment in one group for medical diagnosis [23], [24], and recommending TV programs as a social TV for a group of people [22].

Recommendation for a group is more complicated than the individual recommendation due to the need to combine various individual preferences and requirements [26], [27]. The most notable difference is in the aggregation mechanism to represent a group [28], [29]. There are two approaches for aggregation mechanism; profile aggregation and recommendation aggregation [27], [28].

This profile aggregation approach is the most commonly used in GRS. It aggregates the individual profiles of all group members into a single group profile and then making recommendations based on this single group profile. Single group profile represents group preferences [29]. The recommendation aggregation approach, on the other hand, assembled the individual recommendations of each member and combined them to create a recommendation list for the group.

As mentioned earlier, approaches in GRS generally follow a three-step process [30]: (i) Group formation - identification of users with similar preferences as group members; (ii) Group modeling - aggregation of group members' preferences; and (iii) Group Prediction - prediction of unrated items. The initial task in the group recommendation process is to partition users into groups in the most appropriate way. Few studies use established groups [31]. However, in most cases, groups are usually not known in prior [32]. Thus, as mentioned in [33], groups can be in the form of random groups, occasional groups, and automatically detected groups. We briefly describe these groups as follows:

#### 1) ESTABLISHED GROUP

A group of people who have made the conscious decision to be a part of it as they have common long-term goals. These groups are persistent, and users actively join them [34]. GRec_OC [35], which recommends books for online communities, is an example of established groups.

#### 2) RANDOM GROUP

A group of people who share the same environment at the same time but have no explicit interests in common. When members of a random group have differing opinions on a product, this poses a challenge for the group [36]. Several known examples of this group used in the study in [37], [38], and [39].

#### 3) OCCASIONAL GROUP

A group of people who occasionally do something together and members have a common aim at a particular moment —for instance, traveling [43] and watching movies together [28].

#### 4) AUTOMATICALLY DETECTED GROUP

A group that is formed automatically based on user preferences and/or available resources. The goal of automatic group identification is to find intrinsic communities of users [34]. Several works that applied this group type are in [40] and [41].

### B. LINKED OPEN DATA

The concept of LOD is derived from combining linked data (LD) and open data (OD). LOD is based on the idea to realize the large-scale implementation of a lightweight Semantic Web [42]. Through Semantic Web, each 'thing' is given a Uniform Resource Identifier (URI) which is a single global identification system used for giving unique names to anything. Thus, we can distinguish between different things or know that one thing from one dataset is the same as another in a different dataset. To exploit URIs efficiently, the Resource Description Framework (RDF) provides the platform for graph-based representation for data publishing and interchange on the Web.

LOD cloud somehow enables to overcomes the incompleteness that often characterizes single data sources [43]. One of the major results of LOD lies in DBpedia [44], [45]. DBpedia is based on Wikipedia and can extract RDF data from Wikipedia sites and subsequently provide URIs and RDF data on related topics from various fields [46].

### C. LOD-ENABLED RECOMMENDER SYSTEM

The semantic technology approach is now widely applied in numerous fields and domains. Considerations on how to take advantage of LOD have been raised since real LOD bases became available [47]. In 2014, the ESWC 2014 Challenge on Linked Open Data-enabled RS was launched. The key aim of the challenge was to create a link between the Semantic Web and the RS groups and demonstrate how LOD and semantic technologies would improve the development of a new kind of knowledge-enabled RS.

Examples of the applications of LOD in RS are presented in [48], [49]. LOD dataset properties could be used for a wide range of purposes, for instance in generating cross-domain recommendations [50], generating effective natural-language recommendation explanations [51], and assessing the semantic similarity measure of two resources of LOD datasets [52], [53]. LOD-enabled RSs may be applied to address few issues exhibited by the conventional RSs as follows:

- to address issues such as limited content analysis or cold-start, for example, by bringing new relevant features to improve item representations [54];
- to deal with increasing data sparsity [55], [56]; and
- to deal with serendipitous item recommendation [57], [12].

The applications of LOD in GRS, however, are still limited with little research focusing on this area. However, we highlighted the following researches which are very much related to our work due to the use of semantic technology in GRS.

Garcia *et al.* [58] developed a domain-independent GRS, which can be used with any ontology-based application domain and various group modeling strategies. Their study also presents four acquisition mechanisms for group modeling. The two basic mechanisms adopted are *Average* and *Average without Misery*. While they develop another two novel incremental mechanisms for preference management; *Incremental Intersection* and *Incremental Collaborative Intersection*. The domain ontology entities (tourism and movie) in their GRS are organized hierarchically, with classification levels becoming more specific towards the bottom. Classes in ontology represent the features, i.e. features in the movie's domain are Comedy, Drama, and Romance.

SMART [59], a system for GRS multidimensional semantics was introduced to improve the query language formulation process. SMART proposed a novel ontological approach to perform group profiling in RS. The ontology offers a rich conceptualization of group profiling in the financial data warehouse area by representing the main concepts and relationships between the multidimensional concepts. Furthermore, they create algorithms based on ontological representation to generate relevant semantic recommendations for the group. The system operates in a two-layer process: a layer of semantic group modeling describing group interests using ontological concepts, and the semantic recommendations which are derived from the built profiling ontology by the recommendations generation layer.

Meanwhile, Stratigi *et al.* [21] proposed a function of semantic similarity between users in the health domain, named *SemS*. Beyond the patient's health information, their study considers additional dimensions, which are the level of education, health literacy, and the patient's psycho-emotional status. The exploitation of these dimensions is used in producing relevant and fair recommendations to a group of patients. The proposed semantic similarity measure assumes that the health information dimensions are captured using *SemS*. Other than that, they used the International Classification of Diseases and Related Health Problems (ICD10) ontology to monitor and identify commonalities between health problems and users.

The aforementioned works illustrate the benefits of exploiting semantic structure to enhance the effectiveness of GRS. To date, although applications of LOD have been addressed in some conventional (individual) RSs, its application in GRS is yet to be extensively explored, and more specifically, group formation. Therefore, it is within the interest of this study to explore the contributions of LOD to the effectiveness of GRSs.

## III. THE PROPOSED LOD-ENABLED GRS MODEL

The proposed LOD-enabled GRS model of which we called GRD-LOD is as illustrated in Fig. 1. The model shown is within the overall of our research framework. The model consists of three main components which are: (i) Application of LOD on GRS, (ii) Group Formation, and (iii) Group Modeling. In implementing the research framework, the MovieLens 1 Million (ML1M) dataset and DBpedia knowledge resources were used. Furthermore, during evaluations, the proposed model was evaluated against a set of baselines using standard evaluation metrics. We describe in the following subsection the three components of the GRS-LOD model.

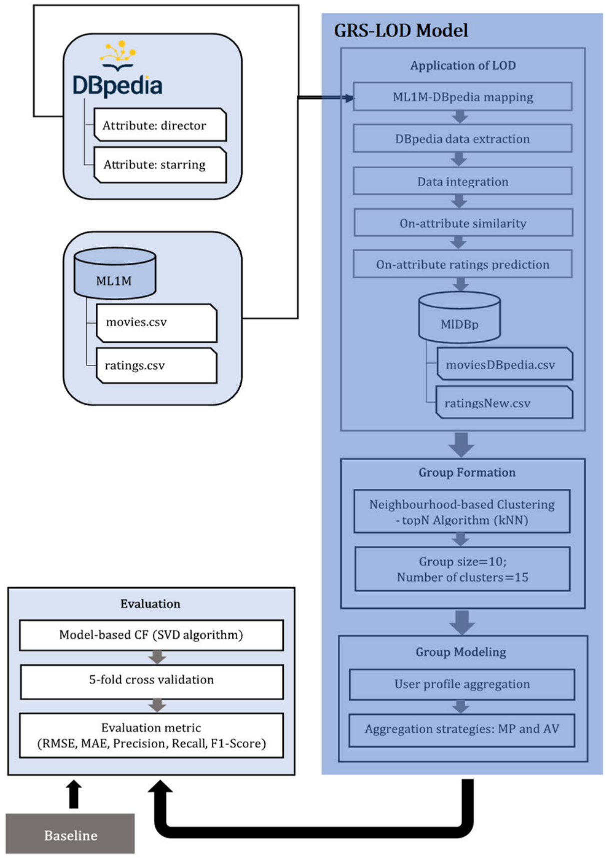


**FIGURE 1.** The research framework of the proposed study.

### A. APPLICATION OF LOD

This section describes the approach used to enhance and enrich the ML1M dataset with DBpedia. The general algorithm for enriching user-item rating matrix with additional LOD data ("director" and "starring" attributes) in the GRS-LOD model is shown in Fig. 2. The algorithm refers to the director attribute. The same approach applies to the starring attribute with only a change in the number of stars starring.

Referring to the Application of LOD process, items that are movies from the ML1M dataset will be mapped to the DBpedia data. In this case, we used the mapping data between the two sources as described in [52]. The matching is based on the title of movies in the ML1M datasets with the rdfs:label property of resources in DBpedia. For example, the movie Sense and Sensibility matches the URI in DBpedia, http://dbpedia.org/resource/Sense_and_Sensibility_(film).

In the proposed approach, matched data from the DBpedia are being traversed and subsequently used to enrich the

**Algorithm I**

```
Input     : Filtered 'director' list; ('director' >= 8)
Output    : New rating prediction
   1   Get a list of 'director'
   2   Get the film directed by the 'director'
          'user_id' and 'rating'; pivot table
   3   Get 'user_id' with the same rating for the same 'director'
       movie
   4   Find items that have not been rated
          Select a movie at random
   5   Predict item
          Five predictions for each item
   6   Add a prediction item to a new rating table, 'ratingsNew'
```

**FIGURE 2.** **The general algorithm of the first component.**

ML1M dataset. A detailed discussion of this approach is provided in [60]. In DBpedia, we use two attributes that are most frequently used in Wikipedia Web pages portraying movies; dbo:starring and dbo:director. These attributes are significant in our work because they influence the selection of films to watch. The finer points of selecting these attributes are also discussed in our prior work [60]. The SPARQL code example as shown in Fig. 3, illustrating the extraction of the director data for the movie Braveheart.

```
"""SELECT DISTINCT ?label ?abstract ?director
WHERE{
  <http://dbpedia.org/resource/Braveheart>rdfs:label ?label.
  OPTIONAL {<http://dbpedia.org/resource/Braveheart> dbo:abstract ?abstract.}
  OPTIONAL {<http://dbpedia.org/resource/Braveheart> dbo:abstract ?director.}
  FILTER (lang(?label)="en")
  FILTER (lang(?abstract)="en")
}
"""
```

**FIGURE 3.** **SPARQL code for 'director' attribute.**

For this study, up to only a maximum of four actors for each movie is being extracted and stored in the enriched ML1M dataset, named MlDBp. While extracting the data associated with the attributes director and starring, directors who have directed at least eight movies and actors who have starred in at least 14 movies are considered and stored in the MlDBp dataset. The data extraction and integration processes correspond to the second and third stages in this process.

Our proposed approach and the LOD-enabled GRS model involved the first level of clustering based on the selected attributes. The first-level of clustering involved the clustering of users based on their similarity of features (i.e., the attributes) selected; dbo:director and dbo:starring as mentioned in the previous section. This is implemented by obtaining on-attribute user similarity within the first level of clustering. Fig. 4 depicts the algorithm for carrying out this stage.

In our model, users are assumed to be similar if they gave identical ratings for the selected attributes of the movies. In this case, the similarity of users is based on the ratings assigned by them for movies with the same actor or director. We assumed ratings $\geq$3.5, indicate that a user likes the movie. Thus, the value of 3.5 was set as the threshold value and user-item data were filtered based on this threshold value. The threshold value is influenced by our experimental work. As we are focusing on obtaining user similarity based on attribute, we are narrowing our lookup for user similarity for each value in a preselected attribute, with a rating of 3.5 or above as the threshold. Therefore, we can create a group of users who, based on that threshold, are the most similar to active users within the attribute. The threshold value 3.5 thus so far provide the best results for our work.

The similarity is based on the assumption: "*If you like a movie with a particular director, you might also like other films directed by the same director.*"

```
Algorithm II
Input: movies_movieDBpedia, users_ratings
Output: matrix_movies
START
    1.  Merge Data;
        mergeAll=mergeData(movieDBpedia,userRat)
    2.  Extract list of films by same director;
        martinScorsese_data =
        mergeAll[mergeAll.director=='Martin
        Scorsese']
    3.  Extract user and rating for films by
        same director;
        martinScorsese_data
        [['movie_id','user_id','rating']]
    4.  Create pivot table;
        matrix_movies =
        martinScorsese_data.pivot()
END
```

**FIGURE 4.** **On-attribute based on first level clustering algorithm.**

The algorithm of the first-level clustering is as illustrated in Fig. 5. We describe the first-level clustering process to find a group of users base on the director attribute. Consider, for instance, the director Martin Scorsese who directed ten movies as shown in Table 1. User data and ratings for each movie directed by Martin Scorsese were obtained via a pivot table. Fig. 6 depicts the pivot table (2918, 10), displaying 2918 user rating data on ten movies directed by Martin Scorsese. Active users were assigned at random based on the pivot table for a particular attribute.

The top-N algorithm with cosine measure is then used to form the cluster. In the first-level clustering, we set $n = 10$, where each group has ten members. Users may be a member of more than one group. Table 2 shows the result of the first-level clustering process for the director Martin Scorsese.

The information gleaned from the first-level clustering process (in this scenario, the Martin Scorsese data) is then used to predict rating. Fig. 7 illustrates the algorithm to carry out this action. Based on the outcome of the first-level of clustering, the Singular Value Decomposition (SVD) prediction algorithm was executed to predict ratings for the unrated items. In the case of this study, for each attribute of director and starring, five predicted ratings were included at random. The process resulted in an enhanced dataset called the MlDBp dataset, as shown in Fig. 1.

For the proposed model, variation in the number of attributes will not change the proposed approach as the

```
Algorithm III
Input: matrix_movies
Output: top10 most similar user
START
    1.  Calculate distances and similarities
        between user sim(u_i,u_k);
        user_sim = 1-
          pairwise_distances(matrix_movies,
          metric="cosine")
    2.  Set active user's most similarity to top10
        users;
    3.  Get user most similar top10;
        def get_user_mostsimilar_top10(user_id,
        topN=10):
          matrix_movies['similarity'] =
            user_sim_df.iloc[user_id-1]
          top_n =
            matrix_movies.sort_values(["similarity"
            ])[0:topN]
          return top_n
    4.  Find similar users;
        def get_user_mostsimilar_top10(5)
END
```

**FIGURE 5.** **User similarity of first level clustering algorithm.**

**TABLE 1.** **List of films for 'director' attribute.**

| director | movie_id | movie_title |
|---|---|---|
| Martin Scorsese | 16 | Casino (1995) |
| | 111 | Taxi Driver (1976) |
| | 412 | Age of Innocence, The (1993) |
| | 1213 | GoodFellas (1990) |
| | 1228 | Raging Bull (1980) |
| | 1343 | Cape Fear (1991) |
| | 1730 | Kundun (1997) |
| | 2022 | Last Temptation of Christ, The (1988) |
| | 2474 | Color of Money, The (1986) |
| | 2976 | Bringing Out the Dead (1999) |

| movie_id | 16 | 111 | 412 | 1213 | 1228 | 1343 | 1730 | 2022 | 2474 | 2976 |
|---|---|---|---|---|---|---|---|---|---|---|
| user_id | | | | | | | | | | |
| 2 | 0.0 | 0.0 | 0.0 | 2.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 5 | 3.0 | 0.0 | 2.0 | 5.0 | 0.0 | 0.0 | 4.0 | 0.0 | 0.0 | 0.0 |
| 8 | 4.0 | 5.0 | 0.0 | 5.0 | 0.0 | 0.0 | 4.0 | 0.0 | 0.0 | 0.0 |
| ... | ... | | | | | | | | | ... |
| 6037 | 0.0 | 5.0 | 0.0 | 3.0 | 0.0 | 2.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 6039 | 0.0 | 4.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 6040 | 0.0 | 5.0 | 0.0 | 4.0 | 5.0 | 0.0 | 0.0 | 50 | 0.0 | 0.0 |

**FIGURE 6.** **An example of a pivot table for the attribute director of Martin Scorsese.**

**TABLE 2.** **The cluster of first-level clustering for attribute of "director = martin scorsese".**

| user_id | movie_id | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 16 | 111 | 412 | 1213 | 1228 | 1343 | 1730 | 2022 | 2474 | 2976 |
| 10 | 5 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 4 |
| 2206 | 0 | 0 | 5 | 0 | 4 | 4 | 0 | 4 | 0 | 0 |
| 934 | 5 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 |
| 933 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2262 | 4 | 3 | 3 | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| 839 | 0 | 3 | 0 | 4 | 5 | 4 | 0 | 0 | 0 | 0 |
| 774 | 5 | 0 | 0 | 5 | 3 | 0 | 0 | 0 | 0 | 3 |
| 689 | 4 | 5 | 0 | 0 | 5 | 3 | 4 | 4 | 0 | 0 |
| 5366 | 3 | 4 | 4 | 3 | 0 | 0 | 3 | 0 | 3 | 0 |
| 3635 | 0 | 0 | 3 | 0 | 3 | 4 | 0 | 3 | 0 | 0 |

approach is generic enough and can adapt with variations in the number of attributes. However, we reasonably believe that increasing the number of attributes in this proposed model may have an effect on the quality of recommendation results. This is due to the fact that the enhanced dataset will have an increase in user ratings. However, for the model we propose, it is contingent upon a few of the parameters being set. For instance, the number of predicted ratings at each attribute. The attributes dbo:director and dbo:starring, on the other hand, are the most important in the movie domain. Experimenting with how the various attributes contribute to GRS performance is fascinating and desirable. As a result, we've designated this as a priority for the foreseeable future.

Based on the newly created MIDBp dataset, the approach then proceeds with group formation, which is discussed in the following section.

```
Algorithm IV
Input: martinScorsese_data
Output: rating prediction r_u,i within on-attribute
START
    1.  Calculate prediction;
        martinScorsese_data =
          Dataset.load_from_df(martinScorsese_data
          )
    2.  Retrieve the trainset;
        trainset=
          martinScorsese_data.full_trainset()
        algo.train(trainset)
    3.  Get rating prediction r_u,i for random user u
        within a cluster of the same similarity of
        directors' attribute;
    4.  Repeat 5 times (n) with different random
        user u of unrated item m;
        for(n=5)
          actual_rating=0
          insert u, m
          algo.predict()
        endfor
END
```

**FIGURE 7.** **Rating prediction algorithm of first-level clustering.**

### B. GROUP FORMATION

The formation of groups which is the second component of the LOD-GRS model, also involved the clustering technique. Groups can be formed intentionally through explicit user definition or automatic system identification [61]. There are four crucial entities involved in the GRS, namely: (i) group members, (ii) group profile, (iii) neighbors, and (iv) recommendation of results. The recommendation for GRS is as shown in (1), where $G$ refers to the target group, $I$ is a set of available items, and $Prediction\ (G, i_k)$ is a utility function for items $i_k$ based on group members $G$.

$$Recommendation(G, I) = \arg \frac{max}{i_k \in I_k} Prediction(G, i_k) \quad (1)$$

GRS can influence interpersonal attraction in groups by emphasizing the similarity between group members. Therefore, it is advantageous to cluster the user for a group based on their similar preferences. The more similar the user preferences are in the group, the better the group recommendations [62]. Thus, the accuracy of group recommendations increases as the similarity between members of the group increases.

$$simU_{u,v} = \cos(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\|^2 \times \|\vec{v}\|^2} \quad (2)$$

**Algorithm V**

```
Input: ratingsNew_df
Output: top10 most similar user
START
    1.  Count and removing duplicates;
        ratingsNew_df[ratingsNew_df.duplicated()]
        ratingsNew=ratingsNew_df.drop_duplicates()
    2.  Create pivot table;
        users_movies_df =
           ratingsNew.reset_index().pivot()
    3.  Calculate distance between user as
        Equation 2;
    4.  Find similar user sim(u_i,u_k) based on the
        distances between user;
        user_sim = 1-
           pairwise_distances(users_movies_df,
           metric="cosine")
    5.  Get user most similar_topN=10;
        def get_user_mostsimilar_top10(user_id,
        topN=10):
          users_movies_df ['similarity'] =
            user_sim_df.iloc[user_id-1]
          top_n =
            users_movies_df.sort_values(["similarit
            y"])[0:topN]
             return top_n
    6.  Find similar users to active user;
        get_user_mostsimilar_top10(1)
    7.  Repeat code of step 6, with changing the
        topN equal to size evaluated
END
```

**FIGURE 8.** **Neighborhood-based clustering algorithm.**

In this study, the kNearest Neighbor (kNN) clustering technique is used. It is proven to be an efficient clustering method to form groups while enabling users to be duplicated in more than one group [63]. The kNN technique groups homogeneous users into automatically detected groups. It is also a widely used and popular technique due to the algorithm's stability and simple to be implemented [64], [67]. The function of neighborhood-based clustering is seen by giving a target user, and the algorithm locates other similar users, often called neighbors and utilizes the neighbors' ratings. In this study, we used the cosine similarity measure of which the equation is as shown in (2), where $\vec{u}\vec{u}$ and $\vec{v}\vec{v}$ are vectors of users. Clusters with similar interests (referring to homogeneous groups) are the best where similar recommendations can be generated. It is necessary to reduce the complexity of decisions and provide recommendations to increase satisfaction among group members [68]. Thus, the best situation is that group members need to have as many similar choices as possible.

We formed the cluster with different group sizes ($n$) of 10, 20, 35, and 50, where $n$ refers to the number of members in a group. Note that users can be members of several groups. Based on the test data, we limit 15 groups only for each group size during the experiment. The output based on the neighborhood-based clustering algorithm (Fig. 8) for the group formation process was analyzed based on the group modeling applied in this study. In this component, we consequently define the clustering phase as a second-level of clustering. Further discussion related to group modeling is presented in the next section.

### C. GROUP MODELLING

As illustrated in Fig. 1, the group modeling corresponds to the third component of the GRS-LOD model. Group modeling combines multiple user models into a group model [23]. In group recommendation, group modeling allows a system to derive a group preference for each item. When it comes to modeling a group, it is essential to note two contexts: the group aggregation approach and the aggregation strategies. As described in [27], [28], the group recommendation approach is divided into aggregating users' profiles and aggregating users' recommendations. In this study, we opt for the profile aggregation approach, which in our case is the aggregation of user preferences.

While for the aggregation strategies, we employ four different strategies, as introduced by Mashtoff [23]: namely the Least Misery (LM), Average (AV), Average Without Misery (AVM), and Most Pleasure (MP) methods. A brief description of each strategy is as follows. Assume that $Grel\,(G, i)$ represents the group preferences for item $i$, $Rel_{ui}$ refers to rating of user $u$ for item $i$, and $G$ represents the group cluster.

#### 1) LEAST MISERY (LM)

The LM is formulated as a group rating with the lowest group member rating, as shown in (3). This strategy is in the category of Border-Line strategy, which according to [69], maybe a good alternative for small groups; however, it has a high probability that it will affect the recommendations for large groups.

$$Grel(G, i) = \min_{u \in G}(Rel_{ui}) \tag{3}$$

#### 2) AVERAGE (AV)

The AV aggregation strategy is more democratic, as all group members are treated equally, and it provides an average score of all group members' ratings. The equation for AV is as shown in (4).

$$Grel\,(G, i) = \frac{\sum_{u \in G} Rel_{ui}}{|G|} \tag{4}$$

#### 3) AVERAGE WITHOUT MISERY (AVM)

In the case of AVM, a threshold plays an important role. Therefore, items with a rating at a certain threshold ($\delta$) will be removed, and the remaining rating priority will be calculated as the average score for that item. In this experiment, we set $\delta = 3$. The AVM is defined as follows.

$$\begin{aligned} Grel(G, i) \;&=\; \min_{u \in G}(Rel_{ui}) \\ Rel_{ui} \;&>=\; \delta \end{aligned} \tag{5}$$

#### 4) MOST PLEASURE (MP)

In this strategy, the rating assigned to an item for a group is equal to the maximum rating given by the group members. The MP's plan presupposes that the group will be pleased with the highest-rated members [70] and refers to the equation below (6).

$$Grel(G, i) = \max_{u \in G}(Rel_{ui}) \tag{6}$$

## IV. EVALUATION SETTING AND METHOD

In evaluating the proposed LOD-enabled GRS, we implemented the SVD algorithm for prediction and employed the five-fold cross-validation approach. Fifteen groups were formed using the ML1M dataset. The evaluation was performed in three different contexts.

1) Comparison of the proposed GRS-LOD against the baseline. The baseline referred to the GRS without the use of LOD.
2) Comparison of the proposed GRS-LOD model against existing works of GRS. Three existing works were considered [8], [71], and [72].
3) Assessing the effects of different group sizes and different strategies of the proposed GRS-LOD model. Diverse group sizes are considered to assess the effect of group size on the group recommendation results. We hypothesized that varying group sizes would impact group satisfaction.

### A. THE DATASET

We used the MovieLens 1M (ML1M) dataset to evaluate the effectiveness of the proposed model. The ML1M dataset includes 6040 users and 3952 movies with 1,000,209 ratings. While on the other hand, DBpedia as LOD dataset was applied to retrieve additional information for the items (refer to films in this study). As mentioned earlier, two attributes were chosen, the dbo:director and dbo:starring attributes.

### B. THE METRICS

To assess the overall performance of the proposed model, we employ two types of metrics: the error metrics for measuring the prediction accuracy of the proposed model; and the relevancy metrics to assess the relevancy of the recommended items to groups.

The prediction accuracy is evaluated using the Root Mean Square Error (RMSE) (7) and Mean Absolute Error (MAE) (8) metrics which are widely used in RSs to measure the difference between predicted scores and actual user ratings[73]. Here, $n$ represents the number of predicted ratings, while $r_{ui}$ and $\hat{r}_{ui}$ respectively refer to the actual and predicted ratings of user $u$ on item $i$. They are defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum\nolimits_{u=1}^{n} \left|r_{ui} - \hat{r}_{ui}\right|^2} \tag{7}$$

$$MAE = \frac{1}{n}\sum\nolimits_{u=1}^{n} \left|r_{ui} - \hat{r}_{ui}\right| \tag{8}$$

RMSE and MAE mainly concern with the accuracy of prediction and they are undisputedly the most popular measure for evaluating recommender systems [74]. However, RMSE and MAE do not reflect the actual user experience. Thus, according to McLaughlin and Herlocker [75] precision and recall reflect the actual user experience better than RMSE and MAE do because, in most cases, users received ranked lists from a recommender instead of predictions for ratings of specific items. Both precision (9) and recall (10) are computed as fractions of $hits_u$ which is the number of correctly recommended relevant items for user $u$. The equations for precision ($P$) and recall ($R$) are as follows respectively, where $recset_u$ is the recommended items for user $u$ and $testset_u$ refers to hits owing to the testing set size. We also consider the F1-score (11) which is the harmonic means between both $P$ and $R$.

$$P_u = \frac{|hits_u|}{|recset_u|} \tag{9}$$

$$R_u = \frac{|hits_u|}{|testset_u|} \tag{10}$$

$$F1 - score = 2 \cdot \frac{P_u \cdot R_u}{P_u + R_u} \tag{11}$$

Precision and recall are binary metrics that are used to assess models with binary output. As a result, we require a method to convert our numerical rating problem from 1 to 5 into a binary problem (relevant and not relevant items). In this experiment, we assume that any true rating greater than 3.5 corresponds to a relevant item and that any true rating less than 3.5 is irrelevant. A relevant item for a particular user-item pair indicates that this item is a good recommendation for the user in question.

## V. RESULTS AND DISCUSSION

### A. EVALUATION OF GRS-LOD MODEL WITH THE BASELINE

In this experiment comparison of the GRS-LOD model is made against the GRS model without the use of LOD. We refer to the GRS model without LOD as the baseline. The experiments involved 15 groups, where each group consists of 10 members (users). The AV and MP were the aggregation strategies used for this experiment as both have been widely adopted in many research works [26], [62]. The results of the experiments are shown in Table 3.

**TABLE 3.** GRS-LOD and baseline evaluation score comparison.

| | Baseline | | GRS-LOD | |
|---|---|---|---|---|
| | MP | AV | MP | AV |
| Mean RMSE | 1.0246 | 0.9178 | 1.0141 | 0.8983 |
| Mean MAE | 0.8266 | 0.7117 | 0.8108 | 0.6881 |
| Precision ($k$=5,$th$=3.5) | 0.9200 | 0.9333 | 0.9467 | 0.9600 |
| Recall | 0.0497 | 0.0515 | 0.0576 | 0.0621 |
| F1-score | 0.0943 | 0.0976 | 0.1085 | 0.1167 |

A statistical hypothesis test assessed the significant differences between the observed results at 5% of the significance level. In this case, we want to assess whether the improvements of the proposed GRS-LOD are significant or otherwise. Therefore, the paired t-test method is used, which is a form of repeated measures design where the same variable is measured on several occasions for each subject. The results are shown in Table 4.

**TABLE 4.** **Paired t-test findings for baseline and GRS-LOD.**

| | | *P*-value | Mean Diff. | Std.Err Diff. |
|---|---|---|---|---|
| RMSE | AV | .000 | 0.03996 | .00526 |
| | MP | .002 | 0.02609 | .00675 |
| MAE | AV | .000 | 0.04053 | .00484 |
| | MP | .003 | 0.01749 | .00487 |
| Precision | AV | .002 | -.03089 | .00786 |
| | MP | .000 | -.01720 | .00328 |
| Recall | AV | .000 | -.01196 | .00234 |
| | MP | .076 | -.00705 | .00368 |
| F1-score | AV | .000 | -.02096 | .00407 |
| | MP | .073 | -.01245 | .00642 |

Based on the table, the *P*-value is less than 0.05 for all cases except for the MP strategy for the recall and F1-score. Thus, indicating a significant improvement in the GRS-LOD model for all measurement metrics with the AV strategy. On the other hand, the result also shows that the proposed MP strategy does not significantly improve the GRS-LOD for the recall and F1-score. As shown in Table 3 and Fig. 9, based on the values of RMSE and MAE, the GRS-LOD shows better prediction accuracy for both strategies. However, comparing between the strategies, the AV strategy gives a more accurate prediction than the MP strategy for both the baseline and the GRS-LOD model.

Similar results are also shown in terms of relevant recommendations. The precision, recall, and F1-score show that the GRS-LOD model performs better than the baseline and with the AV strategy achieved better results than the MP strategy (Fig. 10). Improvement by approximately 2.9% for the precision has been achieved for both strategies for the precision. The recall measure is low because we only measure at the top-5 of the recommended items. The results, thus, prove that knowledge extracted from the LOD can significantly improve the recommendation effectiveness of GRS [76].

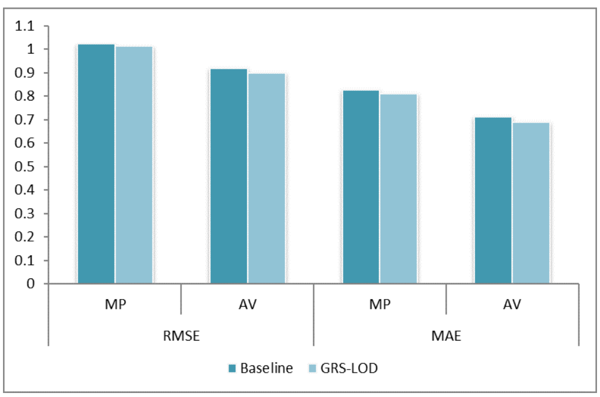


**FIGURE 9.** **Prediction accuracy graph for baseline and GRS-LOD model.**

In terms of aggregation strategy, the AV strategy produces better results than MP. Compared to the baseline, the AV strategy produced 96.0% and 6.21% precision and recall, respectively, for the GRS-LOD model compared to 93.3% and 5.15% for precision and recall, without LOD. It shows an increase of 2.67% in precision, and 1.06% of relevant items were suggested at the top 5. The F1-score shows the harmonic mean of the precision and recall metrics. The AV and MP strategies for the GRS-LOD model respectively show improvement scores of 1.9% and 1.42% compared to the baseline.

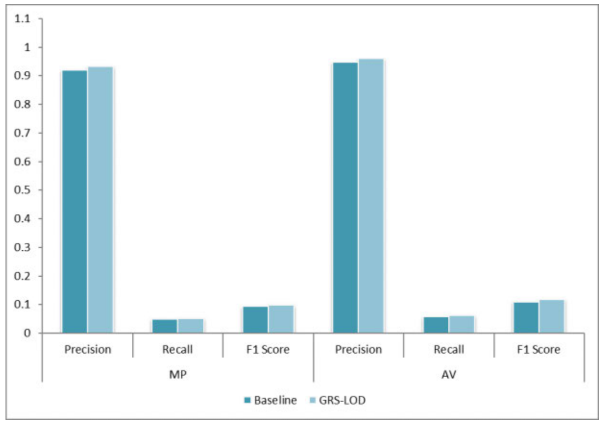


**FIGURE 10.** **Recommendation relevancy graph for baseline and GRS-LOD model.**

In terms of the two aggregation strategies, relying on the highest rating among the group members to represent the group as represented by the MP strategy is less effective than considering all group preferences (represented by the AV strategy). It thus supports the view that if a strategy only considers some of the group's priorities, the accuracy of the proposed results for the group will suffer [8].

Overall, the results indicate that the GRS-LOD can improve the quality of prediction accuracy by reducing RMSE and MAE errors for both AV and MP strategies. It proves that the reduction in sparsity in the user-item matrix based on the GRS-LOD model approach is effective. In this sense, the GRS-LOD model is able to predict unrated data by exploiting and inferring the knowledge from the LOD clouds (i.e., the DBpedia). Thus, it ameliorates the data sparsity issued in GRS. The result of the experiment demonstrates the potential of the proposed approach. The sparsity issues should be overcome before the aggregation process [72]. It is because data sparsity affects the process of the aggregation, including the aggregation strategy used. It thus subsequently affects the clustering process.

Moreover, with the additional knowledge gained from the LOD, the two-level clustering introduced in this study could group more similar users in the same cluster. It then subsequently influenced the recommendation quality to the group. Thus, supporting the hypothesis that the more similar the users in a group, the more effective the group's recommendation [77]. This scenario is also supported in [8], [17], which stated that data sparsity strongly influences the clustering process, and overcoming it will improve the quality of recommendation to a group. As a result, sparsity can be viewed as a critical factor in group formation and subsequently in generating quality recommendations in GRSs.

### B. EVALUATION OF GRS-LOD MODEL WITH THE PREVIOUS APPROACH

In this section, we compare our proposed approach to three existing approaches, which are the approaches proposed by Boratto & Carta [8], Hammond *et al.* [71], and Pujahari & Singh [72]. These approaches were chosen based on four criteria: (i) they are systems for group recommendations; (ii) the use of datasets belonging to the exact domains; (iii) eliminating data sparseness problems was the main aim of the approaches; and (iv) they share identical aggregation strategies.

In terms of group size, we compare the groups' sizes to the groups' sizes as stated in each study. As previously stated, we highlighted two aspects in evaluating the performance of our proposed model: the prediction accuracy and the recommendation relevancy. Thus, the studies by Boratto and Carta [8] and Hammou *et al.* [71] are assessed in terms of prediction accuracy, while the study by Pujahari and Singh [72] is considered for recommendation relevancy. The work of Boratto and Carta [8] applies the KMeans clustering technique to cluster users into groups based on the Predict & Cluster approach. Similarly, the study by Hammou *et al.* [71] also applies the kMeans method by introducing the Archi2 km and Archi2 bkm (Bisecting kMeans).

Based on the findings in Table 5 and Fig. 11, it can be seen that the GRS-LOD model provides better prediction accuracy with RMSE = 0.872 for the group size 20 and RMSE = 0.7816 for the group size of 50 as compared to the other three approaches. The approach proposed in [8] employs the user-based CF technique to predict items that users have not rated to reduce sparsity. User-based CF belongs to the memory-based RS, and this method uses the entire database to generate recommendations. According to Shahab [78], memory-based RS methods do not perform well for large data sets. Besides, the clustering process is also affected by the complexity of each iteration in the rating matrix for the item prediction process.

**TABLE 5. Findings in aspect of prediction accuracy for GRS-LOD model and previous study.**

| | Group Size | |
|---|---|---|
| | 20 | 50 |
| GRS-LOD Model | 0.8720 | 0.7816 |
| Predict&Cluster [8] | 0.9554 | 0.9435 |
| Archi2_bkm [71] | 0.9799 | 0.9785 |
| Archi2_km [71] | 0.9795 | 0.9781 |

Conversely, the researchers in [71] applied distributed technology, XGBoost (Extreme Gradient Boosting), to implement their study through the Apache Spark platform. Among the aspects of the study in [71] is to perceive the effectiveness of distributed technology based on Big Data on-time computing in generating recommendations. However, this aspect is beyond the scope of this study. The GRS-LOD model is more effective in prediction accuracy based on the results in Table 5 due to the insertion of relevant rating data based on data exploited from the DBpedia. Furthermore, the prediction algorithm approaches employed, based on matrix factorization with the SVD model, play a vital part in delivering better prediction accuracy.

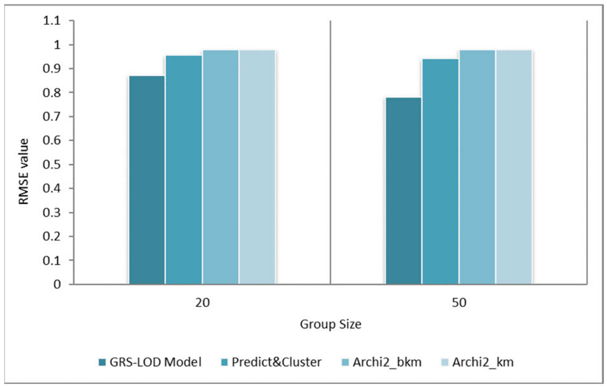


**FIGURE 11. Prediction accuracy comparison of GRS-LOD model and previous study.**

In contrast to the comparison against the study in [8], [71], the study in [72] is being evaluated for comparing the recommendation relevancy aspect. Pujahari & Singh [72] applies 'preference relation' based on matrix factorization to reduce sparsity and acquire unknown relationship preferences for all users. Then it is used in the profile aggregation phase and develops a model known as PR-GA-GRS.

Table 6 shows the comparison results for our GRS-LOD model against the PR-GA-GRS model based on a precision@10 for groups sizes of 10 and 20 using the AV aggregation strategies. The GRS-LOD and PR-GA-GRS models differ substantially in the type of group generated. PR-GA-GRS focuses on random group generation, where group members can have different options and interests. Instead, the GRS-LOD model automatically detects the homogeneous group, defined as similar preferences users in one cluster.

**TABLE 6. Findings in aspect of recommendation relevancy for GRS-LOD model and previous study.**

| | Group Size | |
|---|---|---|
| | 10 | 20 |
| GRS-LOD Model | 0.9067 | 0.9467 |
| PR-GA-GRS [72] | 0.5523 | 0.5346 |

The results in Table 6 and Fig. 12 conclude that the homogeneous clusters generate better relevant recommendations as compared to the randomly formed clusters. It is because groups with similar interests will give similar recommendations based on collaborative filtering techniques.

**TABLE 7.** Evaluation of group size and aggregation strategies based on the aspect of prediction accuracy.

| Group Size ($n$) | Mean RMSE MP | AV | AVM | LM | Mean MAE MP | AV | AVM | LM |
|---|---|---|---|---|---|---|---|---|
| $n$=10 | 1.0303 | 0.9497 | 0.6681 | 1.0867 | 0.8197 | 0.7466 | 0.5485 | 0.8769 |
| $n$=20 | 0.9571 | 0.8717 | 0.6238 | 1.0521 | 0.7489 | 0.6748 | 0.5055 | 0.8462 |
| $n$=35 | 0.9172 | 0.8116 | 0.5970 | 1.0278 | 0.7068 | 0.6193 | 0.4810 | 0.8307 |
| $n$=50 | 0.8881 | 0.7760 | 0.5795 | 1.0179 | 0.6773 | 0.5862 | 0.4620 | 0.8245 |

Recommended items that are relevant to the group member's choices improve group member satisfaction.

In addition, according to Hammou *et al.* [71], the recommendations will be more useful since more users with similar preferences are located in the same cluster. This clarifies that system performance relies not only on the size of the group but also on the similarity between user interests. The findings by Nozari and Kohi [79] also emphasize that the quality of the recommendations is even better if the group members have similar preferences.

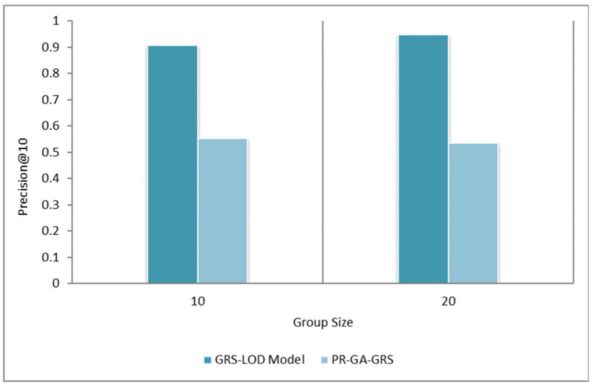


**FIGURE 12.** The recommendation relevancy comparison between the GRS-LOD model and previous study.

In terms of varying group sizes, the results are consistent with the findings in [78], where when group size increases, recommendation effectiveness tends to decline solely for randomly created groups. Among other things, the challenge for a random group is when members of the group have differences of opinion on a particular item [36]. Therefore, the priority given by group members is also inconsistent due to different opinion choices. The findings in [80], [3], and [81] are among those that provide research results that support group formation built with similar member preferences delivering better recommendation results.

### C. EVALUATION OF GROUP SIZE AND AGGREGATION STRATEGIES

The size of a group is an essential element that directly influences both performances using the group modeling approach and group member satisfaction [32], [80]. We experimented with different group sizes and aggregation strategies for this type of evaluation to analyze their impact on the recommendation effectiveness. The group sizes ($n$) experimented with are 10, 20, 35, and 50, and evaluated to four aggregation strategies: MP, AV, AVM, and LM.

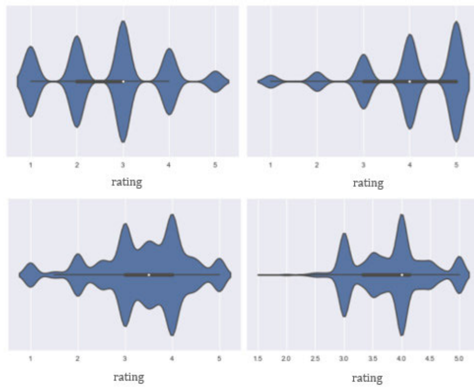


**FIGURE 13.** Violin plot of group rating for four aggregation strategies – group size = 50 (upper left: LM, upper right: MP, lower left: AV, lower right: AVM).

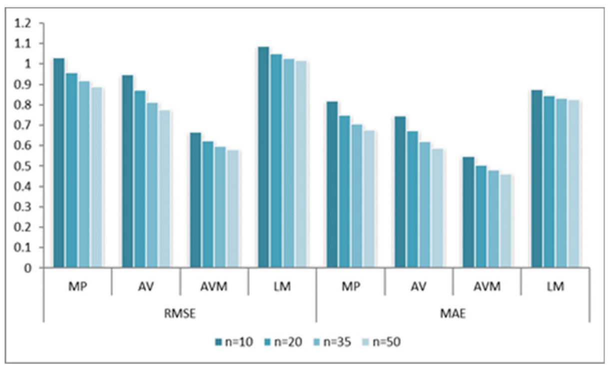


**FIGURE 14.** Comparison of group size and aggregation strategies in prediction accuracy aspect.

Fig. 13 shows the violin plots group ratings after each strategy is implemented for the group size of 50. The width of the violin plot indicates the data density. The rating pattern for the group sizes of 10, 20, and 30 are approximately the same as in Fig. 13. Thus, the rating pattern for each of these aggregation strategies influences the prediction accuracy aspects' evaluation results.

Table 7 and Fig. 14 show the prediction accuracy score of RMSE and MAE metrics in terms of group size and aggregation strategies. It illustrates that the prediction error decreases with the group size for all four strategies indicating that the

**TABLE 8.** Evaluation of group size and aggregation strategies based on the aspect of recommendation relevancy.

| Group Size ($n$) | Precision (k=5, th=3.5) | | | | Recall | | | | F1-score | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MP | AV | AVM | LM | MP | AV | AVM | LM | MP | AV | AVM | LM |
| $n=10$ | 0.9386 | 0.8719 | 0.9033 | 0.6299 | 0.0667 | 0.0384 | 0.0414 | 0.0371 | 0.1245 | 0.0737 | 0.0793 | 0.0701 |
| $n=20$ | 0.9733 | 0.9092 | 0.9413 | 0.6559 | 0.0248 | 0.0276 | 0.0295 | 0.0328 | 0.0485 | 0.0537 | 0.0572 | 0.0625 |
| $n=35$ | 0.9866 | 0.9386 | 0.9466 | 0.6393 | 0.0196 | 0.0240 | 0.0262 | 0.0325 | 0.0386 | 0.0469 | 0.0510 | 0.0620 |
| $n=50$ | 0.9893 | 0.9600 | 0.9439 | 0.6399 | 0.0165 | 0.0221 | 0.0238 | 0.0330 | 0.0326 | 0.0432 | 0.0465 | 0.0629 |

larger the group size, the lower the RMSE and MAE values. In other words, a large group outperforms a small group in terms of prediction accuracy, and it applies to all four of the strategies. This outcome is corroborated by the fact reported in [73], as they claimed that prediction quality improves with increasing group size.

The AVM strategy provides the best prediction accuracy, followed by the AV, MP, and LM. As previously mentioned, we set the rating threshold for AVM strategy at 3. It means that items with a rating of less than three will not be considered. The AVM strategy emphasizes all group members' preferences by taking the average preferences in the group. However, it also accentuates the angle to misery and user satisfaction with the threshold setting. Therefore, the AVM strategy is seen to be effective in improving the quality of prediction. This assessment supports the findings of a study reported in [81], where group members are more satisfied when the recommendations are made via the AV, AVM, and MP strategies.

The AV strategies, without threshold, which takes into account the preferences of all group members, also delivers good prediction accuracy. The average percentage difference between the four group sizes for AVM is 8.86%, which is less than the AV strategy's 17.46%. It explains that the prediction accuracy difference for each AVM strategy size is smaller than the AV strategy. Although the AVM strategy is the highest prediction accuracy based on RMSE metrics, the increase in group size has less impact with a smaller reduction in measurement metrics than the AV strategy.

The LM strategy, however, yields the lowest prediction error accuracy for the four group sizes tested. The LM strategy takes into account the lowest rating priorities in a group. The work by Masthoff [83] reported a less efficient conclusion for the LM strategy as well. It is due to each user preference has a significant impact on small groups [82]. This scenario affects the calculation of the rating prediction error, where the average rating range in the dataset is in the range of [3.0,4.0], as shown in Fig. 13. The same situation also influences the results of the MP strategy considering the maximum preferences in the group.

Subsequently, we offer the findings for this evaluation context, focusing on the recommendation relevancy aspect based on the relevancy metric. Fig. 15, Fig. 16, and Fig. 17 show the precision, recall, and F1-score results, respectively. The same results are also shown in Table 8.

Generally, in terms of relevance, as the size of the group increased, the recommendation relevancy decreased. This can be seen from the values of the F1-score (Fig. 17) that represent the harmonic means between precision and recall. Thus, it contradicts the pattern of prediction accuracy results. For the aggregation strategies of MP, AV, and AVM, the relevance of item recommendation reduces as group size increases. It can also be explained that the relevance value of the top-5 item recommendation is inversely proportional to the growth of the group size. The LM method, on the other hand, demonstrates inconsistent relevancy performance towards larger group sizes.

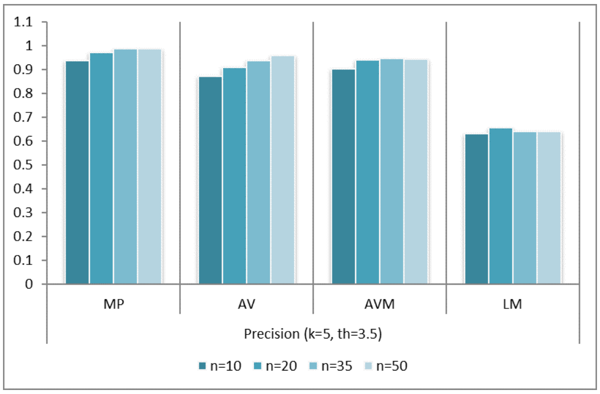


**FIGURE 15.** Comparison of group size and aggregation strategies in recommendation relevancy aspect (Precision).

We narrow down the discussion of the findings in terms of precision and recall. Fig. 15 indicates that the precision of MP and AV strategies improves as group size increases. In contrast, the precision value is inconsistent for the AVM and LM strategies as the group size grows. For AVM strategy, there is an increase in the precision with the increase of group size for sizes 10, 20, and 35, but there is a reduction in the precision value for the size 50. Utilizing the LM strategy, there is an increase for sizes 10 and 20, but the precision value decreases from 35 to 50. Overall, the MP and AVM aggregation strategies perform well, with more than 90% of items recommended to the group being relevant. The study of [86] also uncovered unpredictability in precision values for the group sizes less than 50.

Meanwhile, the recall results, as shown in Fig. 16 show that the percentage of relevant recommendations for all the

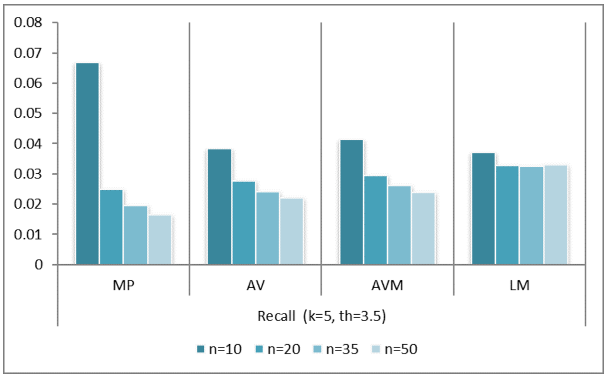


**FIGURE 16.** Comparison of group size and aggregation strategies in recommendation relevancy aspect (Recall).

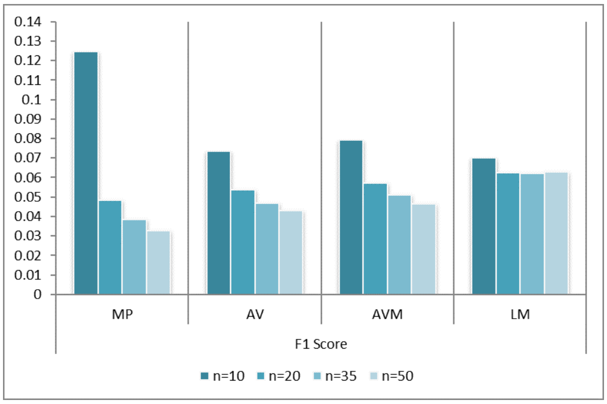


**FIGURE 17.** Comparison of group size and aggregation strategies in recommendation relevancy aspect (F1-score).

strategies is higher for the group size of 10. As a result, we can argue that the F1-scores (Fig. 17) are influenced by the recall values because the larger the group size, the lower the F1-score value for all aggregation strategies except the LM strategy. For the group size of 10, the MP strategy gives the highest F1-score, followed by the AVM, AV, and LM strategies. The increase in group size shows that the LM strategy did not show a significant increase in F1-score, where the values were in the range of 0.062 to 0.0701, with n = 10 shows the highest F1-score. Thus, if the LM strategy is being preferred, a smaller size group is sufficient. The results coincide with [28] views that the LM strategy is suitable for smaller groups.

As a whole, we discovered that, while the prediction accuracy provides better results for the group with more members, the group members' item recommendation is not always relevant. The rationale for this is that more group members result in larger preferences. Therefore a small group has a higher chance of delivering a more relevant recommendation.

## VI. CONCLUSION

In this paper, we presented the results of our investigation on the role of LOD (i.e., DBpedia) in improving the recommendation quality of GRSs. The experiments on the MovieLens 1M dataset showed encouraging results. The DBpedia establishes semantic relationships and enriches the MovieLens 1M dataset items, thus easing the dataset's sparseness. We addressed data sparsity issues before the group formation, assuming that it would be more efficient to address data sparsity at the individual level before the group formation process. Therefore, applying the approach through LOD technology is proposed to ensure that the data sparsity issues can be overcome before the group formation process is implemented.

Thus far, the results obtained are promising and provide evidence that the quality formation of groups resulted in more accurate group recommendations. Overall, it shows that group formation produced through GRS-LOD creating homogeneous user groups based on the automatically detected group makes more effective recommendations. We can conclude that recommendation approaches with features extracted from LOD outperformed non-LOD-based methods based on the experiment we performed in this study.

In terms of future work, group recommendation generation can be focused on other features such as context, trust, and friendship, which may be possible by leveraging the social network. Another potential area for GRS is the inclusion of an explanation facility. By functioning in an explainable manner, the recommendation algorithm not only generates a list of recommendations, but also explanations for the recommendations. In addition, in a future study, we plan to address the issue of a complete cold-start problem in order to increase the accuracy of the group recommendation.

## ACKNOWLEDGMENT

The authors would like to express their gratitude to the Universiti Tun Hussein Onn Malaysia for providing scholarship assistance throughout this study.

## REFERENCES

[1] A. Delic, J. Neidhardt, T. N. Nguyen, and F. Ricci, "An observational user study for group recommender systems in the tourism domain," *Inf. Technol. Tourism*, vol. 19, nos. 1–4, pp. 87–116, Jun. 2018, doi: 10.1007/s40558-018-0106-y.

[2] A. Felfernig, M. Atas, M. Stettinger, T. N. T. Tran, and S. Reiterer, "Group Recommender Applications," in *Group Recommender Systems: An Introduction*, A. Felfernig, L. Boratto, M. Stettinger, and M. Tkalcic, Eds. Cham, Switzerland: Springer, 2018, pp. 75–89.

[3] H. Xu, Y. Ding, J. Sun, K. Zhao, and Y. Chen, "Dynamic group recommendation based on the attention mechanism," *Future Internet*, vol. 11, no. 9, p. 198, Sep. 2019.

[4] N. F. Al-Bakri and S. H. Hashim, "Reducing data sparsity in recommender systems," *Al-Nahrain J. Sci.*, vol. 21, no. 2, pp. 138–147, Jun. 2018, doi: 10.22401/JNUS.21.2.20.

[5] X. Amatriain, A. Jaimes, N. Oliver, and J. M. Pujol, "Data mining methods for recommender systems," in *Recommender Systems Handbook*. Dordrecht, The Netherlands: Springer, 2011, pp. 39–71.

[6] M. Radovanović, A. Nanopoulos, and M. Ivanović, "Hubs in space: Popular nearest neighbors in high-dimensional data," *J. Mach. Learn. Res.*, vol. 11, pp. 2487–2531, Sep. 2010.

[7] W. Wang, G. Zhang, and J. Lu, "Member contribution-based group recommender system," *Decision Support Syst.*, vol. 87, pp. 80–93, Jul. 2016, doi: 10.1016/j.dss.2016.05.002.

[8] L. Boratto and S. Carta, "Using collaborative filtering to overcome the curse of dimensionality when clustering users in a group recommender," in *Proc. 16th Int. Conf. Enterprise Inf. Syst. (ICEIS)*, Apr. 2014, pp. 564–572, doi: 10.5220/0004865005640572.

[9] S. Ghazarian, N. Shabib, and M. A. Nematbakhsh, "Improving sparsity problem in group recommendation," in *Proc. Hypertext Extended Late-Breaking Results, Doctoral Consortium Workshop 25th ACM Hypertext Social Media Conf. (Hypertext)*, no. 3, F. Cena, A. S. da Silva, and C. Trattner, Eds. Santiago, Chile, Sep. 2014, pp. 1–6.

[10] S. Ihm, S. Lee, Y. Park, A. Nasridinov, M. Kim, and S. Park, "A technique of recursive reliability-based missing data imputation for collaborative filtering," *Appl. Sci.*, vol. 11, no. 8, pp. 1–16, 2021.

[11] E. Inan, F. Tekbacak, and C. Ozturk, "Moreopt: A goal programming based movie recommender system," *J. Comput. Sci.*, vol. 28, pp. 43–50, Sep. 2018.

[12] N. I. Y. Saat, S. A. Mohd Noah, and M. Mohd, "Towards serendipity for content–based recommender systems," *Int. J. Adv. Sci., Eng. Inf. Technol.*, vol. 8, nos. 4–2, p. 1762, Sep. 2018, doi: 10.18517/ijaseit.8.4-2.6807.

[13] W. H. El-Ashmawi, A. F. Ali, and A. Slowik, "Hybrid crow search and uniform crossover algorithm-based clustering for top-N recommendation system," *Neural Comput. Appl.*, vol. 33, no. 12, pp. 7145–7164, Jun. 2021, doi: 10.1007/s00521-020-05482-6.

[14] H. Zarzour, F. Maazouzi, M. Soltani, and C. Chemam, "An improved collaborative filtering recommendation algorithm for big data," in *Proc. 6th IFIP Int. Conf. Comput. Intell. Appl.*, Oran, Algeria, May 2018, pp. 660–668, doi: 10.1007/978-3-319-89743-1.

[15] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl, "Application of dimensionality reduction in recommender system—A case study," in *Proc. ACM-SIGKDD Conf. Knowl. Discovery Databases (WEBKDD), Web Mining E-Commerce Challenges Opportunities*. Boston, MA, USA: ACM, 2000.

[16] N. A. Osman, S. A. M. Noah, and M. Darwich, "Contextual sentiment based recommender system to provide recommendation in the electronic products domain," *Int. J. Mach. Learn. Comput.*, vol. 9, no. 4, pp. 425–431, Aug. 2019, doi: 10.18178/ijmlc.2019.9.4.821.

[17] A. Roy, S. Banerjee, C. Bhatt, Y. Badr, and S. Mallik, "Hybrid group recommendation using modified termite colony algorithm: A context towards big data," *J. Inf. Knowl. Manag.*, vol. 17, no. 2, p. 31, 2018, doi: 10.1142/S0219649219920011.

[18] I. Christensen, S. Schiaffino, and M. Armentano, "Social group recommendation in the tourism domain," *J. Intell. Inf. Syst.*, vol. 47, no. 2, pp. 209–231, Oct. 2016, doi: 10.1007/s10844-016-0400-0.

[19] F. Ortega, R. Hurtado, J. Bobadilla, and R. Bojorque, "Recommendation to groups of users using the singularities concept," *IEEE Access*, vol. 6, pp. 39745–39761, 2018, doi: 10.1109/ACCESS.2018.2853107.

[20] N. D. Thanh, M. Ali, and L. H. Son, "A novel clustering algorithm in a neutrosophic recommender system for medical diagnosis," *Cogn. Comput.*, vol. 9, no. 4, pp. 526–544, 2017, doi: 10.1007/s12559-017-9462-8.

[21] M. Stratigi, H. Kondylakis, and K. Stefanidis, "Multidimensional group recommendations in the health domain," *Algorithms*, vol. 13, no. 54, pp. 1–21, 2020, doi: 10.3390/a13030054.

[22] S. Pyo, E. Kim, and M. Kim, "LDA-based unified topic modeling for similar TV user grouping and TV program recommendation," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1476–1490, Aug. 2015.

[23] J. Masthoff, "Group recommender systems: Aggregation, satisfaction and group attributes," in *Recommender Systems Handbook*, Aberdeen, U.K., 2nd ed., F. Ricci, L. Rokach, and B. Shapira, Eds. Boston, MA, USA: Springer, 2015, pp. 743–776.

[24] N. Shabib, J. A. Gulla, and J. Krogstie, "On the intrinsic challenges of group recommendation," in *Proc. 5th ACM RecSys Workshop Recommender Syst. Social Web Co-Located, 7th ACM Conf. Recommender Syst. (RecSys)*, Hong Kong, Oct. 2013.

[25] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decis. Support Syst.*, vol. 74, pp. 1–38, Jun. 2015, doi: 10.1016/j.dss.2015.03.008.

[26] N. A. Najjar and D. C. Wilson, "Power of the group neighborhood in memory-based group recommender systems," in *Proc. 28th Int. Florida Artif. Intell. Res. Soc. Conf.*, 2015, pp. 495–500.

[27] R. M. Nawi, S. A. M. Noah, and L. Q. Zakaria, "Evaluation of group modelling strategy in model-based collaborative filtering recommendation," *Int. J. Mach. Learn. Comput.*, vol. 10, no. 2, pp. 330–338, Feb. 2020.

[28] J. Castro, J. Lu, G. Zhang, Y. Dong, and L. Martínez, "Opinion dynamics-based group recommender systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 12, pp. 2394–2406, Dec. 2018.

[29] J. Castro, F. J. Quesada, I. Palomares, and L. Martínez, "A consensus-driven group recommender system," *Int. J. Intell. Syst.*, vol. 30, no. 8, pp. 887–906, Aug. 2015, doi: 10.1002/int.21730.

[30] H. Mahyar, K. E. Ghalebi, S. M. Morshedi, S. Khalili, R. Grosu, and A. Movaghar, "Centrality-based group formation in group recommender systems," in *Proc. 26th Int. Conf. World Wide Web Companion*, Perth, WA, Australia, 2017, pp. 1187–1196.

[31] L. Quijano-Sánchez, B. Díaz-Agudo, and J. A. Recio-García, "Development of a group recommender application in a social network," *Knowl.-Based Syst.*, vol. 71, pp. 72–85, Nov. 2014, doi: 10.1016/j.knosys.2014.05.013.

[32] E. Yalcin, A. Bilge, and A. G. Yüksek, "An empirical evaluation of aggregation techniques used in group recommender systems," in *Proc. Int. Conf. Appl. Technol. (ICAT)*, Sarajevo, Bosnia Herzegovina, Aug. 2019, pp. 186–191.

[33] L. Boratto and S. Carta, "ART: Group recommendation approaches for automatically detected groups," *Int. J. Mach. Learn. Cybern.*, vol. 6, no. 6, pp. 953–980, Dec. 2015, doi: 10.1007/s13042-015-0371-4.

[34] E. Khazaei and A. Alimohammadi, "An automatic user grouping model for a group recommender system in location-based social networks," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 2, p. 67, Feb. 2018, doi: 10.3390/ijgi7020067.

[35] J. K. Kim, H. K. Kim, H. Y. Oh, and Y. U. Ryu, "A group recommendation system for online communities," *Int. J. Inf. Manage.*, vol. 30, no. 3, pp. 212–219, 2010, doi: 10.1016/j.ijinfomgt.2009.09.006.

[36] J. Castro, M. J. Barranco, and L. Martinez, "Improving group recommendation with outlier data filtering," in *World Scientific Proceedings Series on Computer Engineering and Information Science. Decision Making and Soft Computing*. Hackensack, NJ, USA: World Scientific, 2014, pp. 562–567, doi: 10.1142/9789814619998_0093.

[37] M. Rijlaarsdam, S. Scholten, and C. C. S. Liem, "Towards creating a non-synthetic group recommendation dataset," in *Proc. 1st Workshop Impact Recommender Syst. Co-Located 13th ACM Conf. Recommender Syst. (ACM RecSys)*. Copenhagen, Denmark: ACM, Sep. 2019, pp. 1–5.

[38] G. Fang, L. Su, D. Jiang, and L. Wu, "Group recommendation systems based on external social-trust networks," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–11, May 2018.

[39] V. R. Kagita, K. C. Meka, and V. Padmanabhan, "A novel social-choice strategy for group modeling in recommender systems," in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Dec. 2015, pp. 153–158, doi: 10.1109/ICIT.2015.18.

[40] Y.-M. Li, C.-L. Chou, and L.-F. Lin, "A social recommender mechanism for location-based group commerce," *Inf. Sci.*, vol. 274, pp. 125–142, Aug. 2014, doi: 10.1016/j.ins.2014.02.079.

[41] L. Boratto and S. Carta, "Modeling the preferences of a group of users detected by clustering: A group recommendation case-study," in *Proc. 4th Int. Conf. Web Intell., Mining Semantics (WIMS)*, 2014, pp. 1–7.

[42] M. Alam *et al.*, "Linked open data validity—A technical report from international semantic web research summer school (ISWS 2018)," Bertinoro, Italy, 2018. [Online]. Available: https://arxiv.org/pdf/1903.12554.pdf

[43] C. Cappiello, T. D. Noia, B. A. Marcu, and M. Matera, "A quality model for linked data exploration," in *Proc. Int. Conf. Web Eng. (ICWE), Web Eng.*, 2016, pp. 397–404.

[44] S. Auer, C. Bizer, J. Lehmann, and Z. G. Ives, "DBpedia: A nucleus for a web of open data," in *The Semantic Web*. Berlin, Germany: Springer, 2007, pp. 722–735, doi: 10.1007/978-3-540-76298-0.

[45] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer, "DBpedia—A large-scale, multilingual knowledge base extracted from Wikipedia," *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015, doi: 10.3233/SW-140134.

[46] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data—The story so far," *Int. J. Semantic Web Inf. Syst.*, vol. 5, no. 3, pp. 1–22, Jul. 2009.

[47] T. Heath, "How will we interact with the web of data," *IEEE Internet Comput.*, vol. 12, no. 5, pp. 88–91, Sep. 2008, doi: 10.1109/MIC.2008.101.

[48] M. de Gemmis, P. Lops, C. Musto, F. Narducci, and G. Semeraro, "Semantics-aware content-based recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, Eds. Boston, MA, USA: Springer, 2015, pp. 119–159.

[49] P. Yochum, L. Chang, T. Gu, and M. Zhu, "Linked open data in location-based recommendation system on tourism domain: A survey," *IEEE Access*, vol. 8, pp. 16409–16439, 2020.

[50] I. Fernández-Tobías, P. Tomeo, I. Cantador, T. Di Noia, and E. Di Sciascio, "Accuracy and diversity in cross-domain recommendations for cold-start users with positive-only feedback," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 119–122, doi: 10.1145/2959100.2959175.

[51] C. Musto, F. Narducci, P. Lops, M. de Gemmis, and G. Semeraro, "Linked open data-based explanations for transparent recommender systems," *Int. J. Hum.-Comput. Stud.*, vol. 121, pp. 93–107, Jan. 2019, doi: 10.1016/j.ijhcs.2018.03.003.

[52] R. Meymandpour and J. G. Davis, "Enhancing recommender systems using linked open data-based semantic analysis of items," *Proc. 3rd Australas. Web Conf. (AWC)*, vol. 166, Sydney, NSW, Australia, Jan. 2015, pp. 27–30.

[53] V. C. Ostuni, T. Di Noia, E. Di Sciascio, and R. Mirizzi, "Top-N recommendations from implicit feedback leveraging linked open data," in *Proc. 7th ACM Conf. Recommender Syst.*, Oct. 2013, pp. 85–92.

[54] S. Bostandjiev, J. O'Donovan, and T. Höllerer, "TasteWeights: A visual interactive hybrid recommender system," in *Proc. 6th ACM Conf. Recommender Syst. (RecSys)*, 2012, pp. 35–42, doi: 10.1145/2365952.2365964.

[55] C. Musto, P. Basile, P. Lops, M. de Gemmis, and G. Semeraro, "Introducing linked open data in graph-based recommender systems," *Inf. Process. Manage.*, vol. 53, no. 2, pp. 405–435, Mar. 2017, doi: 10.1016/j.ipm.2016.12.003.

[56] V. W. Anelli, T. Di Noia, E. Di Sciascio, and P. Lops, "Feature factorization for top-n recommendation: From item rating to features relevance," in *Proc. 1st Workshop Intell. Recommender Syst. Knowl. Transf. Learn. (RecSysKTL)*, Como, Italy. New York, NY, USA: ACM, Aug. 2017, pp. 1–6, doi: 10.1007/s11257-018-9217-6.

[57] V. Maccatrozzo, M. Terstall, L. Aroyo, and G. Schreiber, "SIRUP: Serendipity in recommendations via user perceptions," in *Proc. 22nd Int. Conf. Intell. User Interface*, Mar. 2017, pp. 35–44.

[58] I. Garcia, S. Pajares, L. Sebastia, and E. Onaindia, "Preference elicitation techniques for group recommender systems," *Inf. Sci.*, vol. 189, no. 7, pp. 155–175, Apr. 2012, doi: 10.1016/j.ins.2011.11.037.

[59] E. B. Ahmed, W. Tebourski, W. B. A. Karaa, and F. Gargouri, "SMART: Semantic multidimensional group recommendations," *Multimedia Tools Appl.*, vol. 74, no. 23, pp. 10419–10437, Dec. 2015, doi: 10.1007/s11042-014-2174-0.

[60] R. M. Nawi, S. A. M. Noah, and L. Q. Zakaria, "Issues and challenges in the extraction and mapping of linked open data resources with recommender systems datasets," *J. Inf. Sci. Theory Pract.*, vol. 9, no. 2, pp. 66–82, Jun. 2021, doi: 10.1633/JISTAP.2021.9.2.5.

[61] I. Cantador and P. Castells, "Group recommender systems: New perspectives in the social web," in *Recommender Systems for the Social Web*. Berlin, Germany: Springer, 2012, pp. 139–157.

[62] T. De Pessemier, S. Dooms, and L. Martens, "Comparison of group recommendation algorithms," *Multimedia Tools Appl.*, vol. 72, no. 3, pp. 2497–2541, 2014.

[63] W. Zhang, X. Chen, Y. Liu, and Q. Xi, "A distributed storage and computation k-nearest neighbor algorithm based cloud-edge computing for cyber-physical-social systems," *IEEE Access*, vol. 8, pp. 50118–50130, 2020, doi: 10.1109/ACCESS.2020.2974764.

[64] C. C. Aggarwal, *Recommender Systems: The Textbook*, vol. 39, no. 4. Cham, Switzerland: Springer, 2016.

[65] N. Ali, D. Neagu, and P. Trundle, "Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets," *Social Netw. Appl. Sci.*, vol. 1, no. 12, pp. 1–15, Dec. 2019.

[66] L. Ravi and S. Vairavasundaram, "A collaborative location based travel recommendation system through enhanced rating prediction for the group of users," *Comput. Intell. Neurosci.*, vol. 2016, p. 28, Mar. 2016, doi: 10.1155/2016/1291358.

[67] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi, "Evaluating collaborative filtering recommender algorithms: A survey," *IEEE Access*, vol. 6, pp. 74003–74023, 2018.

[68] Khazaei and Alimohammadi, "Context-aware group-oriented location recommendation in location-based social networks," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 9, p. 406, Sep. 2019.

[69] G. Fernández, W. López, B. Rienzi, and P. Rodríguez-Bocca, "Let's go to the cinema! A movie recommender system for ephemeral groups of users," *CLEI Electron. J.*, vol. 18, no. 2, pp. 1–22, Aug. 2015.

[70] A. Delic, T. N. Nguyen, and M. Tkalcic, "Group decision-making and designing group recommender systems," in *Handbook of e-Tourism*, Z. Xiang, M. Fuchs, U. Gretzel, W. Höpken, Eds. Cham, Switzerland: Springer, 2020, pp. 1–23.

[71] B. A. Hammou, A. A. Lahcen, and S. Mouline, "A distributed group recommendation system based on extreme gradient boosting and big data technologies," *Int. J. Speech Technol.*, vol. 49, no. 12, pp. 4128–4149, Dec. 2019.

[72] A. Pujahari and D. S. Sisodia, "Aggregation of preference relations to enhance the ranking quality of collaborative filtering based group recommender system," *Expert Syst. Appl.*, vol. 156, Oct. 2020, Art. no. 113476, doi: 10.1016/j.eswa.2020.113476.

[73] W. Wang and Y. Lu, "Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model," in *Proc. 5th Int. Conf. Mech. Eng., Mater. Sci. Civil Eng.*, vol. 324, no. 1. Kuala Lumpur, Malaysia: IOP Publishing, Dec. 2017, Art. no. 012049, doi: 10.1088/1757-899X/324/1/012049.

[74] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender System: An Introduction*, 1st ed. New York, NY, USA: Cambridge Univ. Press, 2011.

[75] M. R. McLaughlin and J. L. Herlocker, "A collaborative filtering algorithm and evaluation metric that accurately model the user experience," in *Proc. 27th Annu. Int. Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2004, pp. 329–336, doi: 10.1145/1008992.1009050.

[76] C. Musto, P. Lops, M. de Gemmis, and G. Semeraro, "Semantics-aware recommender systems exploiting linked open data and graph-based features," *Knowl.-Based Syst.*, vol. 136, pp. 1–14, Nov. 2017.

[77] L. Baltrunas, T. Makcinskas, and F. Ricci, "Group recommendations with rank aggregation and collaborative filtering," in *Proc. 4th ACM Conf. Recommender Syst. (RecSys)*, 2010, pp. 119–126.

[78] S. Shahab, *Next Level: A Course Recommender System Based on Career Interest*. San Jose, CA, USA: San Jose State Univ., 2019.

[79] R. B. Nozari and H. Koohi, "A novel group recommender system based on members' influence and leader impact," *Knowl.-Based Syst.*, vol. 205, Oct. 2020, Art. no. 106296, doi: 10.1016/j.knosys.2020.106296.

[80] A. Delic, J. Masthoff, J. Neidhardt, and H. Werthner, "How to use social relationships in group recommenders: Empirical evidence," in *Proc. 26th Conf. User Modeling, Adaptation Personalization*, Jul. 2018, pp. 121–129.

[81] J. Masthoff, "Group recommender systems: Combining individual models," in *Recommender Systems Handbook*, F. Ricci, Ed. Dordrecht, The Netherlands: Springer, 2011, pp. 677–702.

[82] C. Bernier, A. Brun, A. Aghasaryan, M. Bouzid, J. Picault, and C. Senot, "Topology of communities for the collaborative recommendations to groups," in *Proc. 3rd Int. Conf. Inf. Syst. Econ. Intell. (SIIE)*, Sousse, Tunisia, Feb. 2010, p. 6.

**ROSMAMALMI MAT NAWI** received the bachelor's degree in computer science from Universiti Teknikal Malaysia Melaka, and the master's degree in internet computing from the Universiti Pendidikan Sultan Idris. She is currently pursuing the Ph.D. degree with the Center for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia. Her current research interest includes applying linked open data technologies in group recommender systems.

**SHAHRUL AZMAN MOHD NOAH** (Member, IEEE) received the B.Sc. degree (Hons.) in mathematics from the Universiti Kebangsaan Malaysia, in 1992, and the M.Sc. and Ph.D. degrees in information studies from The University of Sheffield, U.K., in 1994 and 1998, respectively. He is currently a Professor with the Center for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, and also heads the Knowledge Technology Research Group. His research interests include information retrieval and ontology with special emphasis on semantic search and recommender systems. He has published more than 200 research articles in these areas. He is also the Deputy Dean of Academics, a member of the International Association for Ontology and its Applications (IAOA) and IEEE Computer Science Society associations.

**LAILATUL QADRI ZAKARIA** received the bachelor's and master's degrees in information science from the Universiti Kebangsaan Malaysia, and the Ph.D. degree from the University of Southampton, U.K. She is currently a Lecturer in natural language processing with the Faculty of Information Science and Technology, University Kebangsaan Malaysia. Her research interests include natural language processing, ontology development, and semantic web technologies.