# Empowering Quality of Recommendations by Integrating Matrix Factorization Approaches With Louvain Community Detection

TOKALA, Srilatha <http://orcid.org/0009-0005-5222-8749>, KRISHNA ENDURI, Murali <http://orcid.org/0000-0002-9029-2187>, TANGIRALA, Jaya Lakshmi <http://orcid.org/0000-0003-0183-4093>, ABDUL, Ashu <http://orcid.org/0000-0003-0221-8225> and CHEN, Jenhui <http://orcid.org/0000-0002-8372-9221>



**Published version**

TOKALA, Srilatha, KRISHNA ENDURI, Murali, TANGIRALA, Jaya Lakshmi, ABDUL, Ashu and CHEN, Jenhui (2024). Empowering Quality of Recommendations by Integrating Matrix Factorization Approaches With Louvain Community Detection. IEEE Access, 12, 164028-164062. [Article]

**Copyright and re-use policy**

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2024.DOI*

# Empowering Quality of Recommendations by Integrating Matrix Factorization Approaches with Louvain Community Detection

**SRILATHA TOKALA[1], MURALI KRISHNA ENDURI[1], (Member, IEEE), T. JAYA LAKSHMI[2,1], (Member, IEEE), ASHU ABDUL[3,1], (Senior Member, IEEE), JENHUI CHEN[4,5,6], (Senior Member, IEEE)**

[1]Department of Computer Science and Engineering, SRM University-AP, India. (e-mail: srilatha_tokala@srmap.edu.in; muralikrishna.e@srmap.edu.in; ashu.a507@gmail.com)
[2]Department of Computing and Software Engineering, Sheffield Hallam University, United Kingdom. (e-mail: jaya.phd.hcu@gmail.com)
[3]Center for Artificial Intelligence in Medicine, Chang Gung Memorial Hospital, Guishan Dist., Taoyuan City 33375, Taiwan.
[4]Department of Computer Science and Information Engineering, Chang Gung University, Guishan Dist., Taoyuan City 33302, Taiwan. (e-mail: jhchen@mail.cgu.edu.tw)
[5]Division of Breast Surgery and General Surgery, Department of Surgery, Chang Gung Memorial Hospital, Guishan Dist., Taoyuan City 33375, Taiwan
[6]Department of Electronic Engineering, Ming Chi University of Technology, Taishan Dist., New Taipei City 24301, Taiwan

Corresponding author: Jenhui Chen (e-mail: jhchen@mail.cgu.edu.tw)

This work was supported by the National Science and Technology Council, Taiwan under Grant MOST 110-2221-E-182-041-MY3, and in part by the Chang Gung Memorial Hospital, Taoyuan, Taiwan under Grant CMRPD2N0051.

**ABSTRACT** Recommendation systems play an important role in creating personalized content for consumers, improving their overall experiences across several applications. Providing the user with accurate recommendations based on their interests is the recommender system's primary goal. Collaborative filtering-based recommendations with the help of matrix factorization techniques is very useful in practical uses. Owing to the expanding size of the dataset and as the complexity increases, there arises an issue in delivering accurate recommendations to the users. The efficient functioning of the recommendation system undergoes the scalability challenge in controlling large and varying datasets. This paper introduces an innovative approach by integrating matrix factorization techniques and community detection methods where the scalability in recommendation systems will be addressed. The steps involved in the proposed approach are: (1) The rating network is modeled as a bipartite network. (2) Communities are generated from the network. (3) Extract the rating matrices that belong to the communities and apply MF to these matrices in parallel. (4) Merge the predicted rating matrices belonging to the communities and evaluate root mean square error (RMSE). In our paper different matrix factorization approaches like basic MF, NMF, SVD++, and FANMF are taken along with the Louvain community detection method for dividing the communities. The experimental analysis is performed on four different diverse datasets to enhance the quality of the recommendation. To determine the method's efficiency, the evaluation metric RMSE is used and the time required to evaluate the computation is also computed. It is observed in the results that almost 95% of our result is proven effective by getting a better RMSE value. Thus, the main aim of the user will be satisfied in getting accurate recommendations based on the user experiences.

**INDEX TERMS** Recommendation system, collaborative filtering, community detection, matrix factorization

## I. INTRODUCTION

Recommender systems filter information that forecasts the tastes of users for products or services, such as books, music, films, articles, or online shop items. These systems are commonly utilized in online shopping, entertainment, social networking sites, and other online platforms to make tailored suggestions to consumers [1]. These recommender systems are broadly categorized into content-based recommendations and collaborative filtering recommendations [2]. The user's product choices are recommended in the content-based recommendations based on their user profile [3]. For example, if a user has previously enjoyed action movies, the content-based method will recommend action movies with comparable features. In the collaborative filtering recommendations, the items are recommended to the users by recognizing trends in user behavior and preferences through the collection and analysis of data from a large number of users [4]. Matrix factorization is a crucial tool in collaborative filtering suggestions.

Matrix Factorization strategies hold great significance across diverse fields and applications. One distinguishing trait is their capacity to decrease dimensionality or transform high-dimensional to low-dimensional representations [5]. The main functionality of the matrix factorization method is to decompose a matrix into two latent feature matrices that capture the main information from the original matrix by removing noise and redundancy [6]. This decomposition helps in the problems of storage and computation for diverse datasets that are large in dimensions [7].

In network science and data analysis, complex datasets undergo several hurdles in achieving and handling computational efficiency and scalability issues [8], [9]. Complex network analysis has garnered substantial interest across diverse fields, including social sciences, biology, and computer science [10], [11]. In the concept of network analysis, there arises the fundamental concept of community detection, which is used to handle large and diverse datasets [12]. The detection of the communities is processed by densely connecting the group of nodes that exhibit strong internal behavior and weakly connecting to the nodes that are outside the community [13]. To define the communities effectively, there are several community detection methods, and the more significant method proven to be effective is the Louvain community detection method. The computational efficiency of the method proves that it captures better community structures which has wide applications across multiple domains [14], [15].

This paper provides several applications where the matrix factorization and community detection approaches are used. Some of the comparable works that the other authors have suggested are included in Section II. Section III defines the methodology used for the different matrix factorization methods and the community detection approach. Section IV is the proposed approach that integrates our matrix factorization approaches and community detection method. A thorough explanation of the datasets utilized and the analysis of the results is given in Section V. In Section VI, the study's results and future scope are outlined.

### A. APPLICATIONS

Matrix Factorization and community detection techniques present vast potential across diverse domains, delivering valuable insights and enriching data analysis in numerous applications [16]. Incorporating real-time applications is essential for addressing the dynamic problems associated with community detection and data processing. Keeping up with the dynamism of datasets and real-world events often presents challenges for traditional data analysis methodologies [17]. Applications that operate in real-time become essential resources for meeting the demands of sectors and fields where prompt insights are critical. These applications enable businesses to quickly gain insights from large datasets, enabling timely interventions and well-informed decision-making [18]. The ability to identify and analyze dynamic network topologies in real-time is critical for community detection since it allows for the modification of strategies, fortification of cybersecurity defenses, and optimization of resource allocation in smart city applications [19]. This powerful synergy has demonstrated remarkable success in several notable areas, such as

Matrix Factorization techniques are effective in modeling user-item interactions and extracting latent features in **recommender systems**. By applying community detection on the user-item latent feature matrices, the recommendation quality will be increased. This enhances the quality of the recommendations by considering the group preferences and the item similarities [20]. The detection of gene modules functionality from the gene analysis is possible by integrating the matrix factorization and community detection in **bioinformatics** [21]. This interaction helps in identifying the mechanism of the diseases which facilitates the exploration of biological and genetic interactions. When it comes to financial systems **fraud detection**, there will be several fraudulent activities that will be going on in the banking sector or online transactions [22], [23]. The community detection and matrix factorization approaches help identify and locate the network where the activities are going on. The system's behavior can be analyzed and helps in handling those risks.

In the stream of **social networking**, similar behavior persons are grouped into communities, and based on the interest of another recommendation that can be processed [24]. The group of persons involved in similar activities are identified by their social structures and influence patterns where the information can be diffused. By detecting suspicious activities in the network traffic where security is the main concern to be handled for **network security** [25]. By the matrix factorization and community approaches, the behavior of the network can be identified and provides better security and does not fall under any anomalous detection. The integration of matrix factorization and community detection helps in the construction of **knowledge graphs** that capture their entities and relationships [26]. This approach facilitates knowledge graph competition, entity linking, and relationship prediction by identifying communities of related entities, ultimately enhancing the depth and accuracy of knowledge representation and analysis.

### B. PROBLEM STATEMENT

The problem of recommender systems is defined by Schafer *et al.* [27] as follows:

Movies

| Users | Movie1 | Movie2 | Movie3 | Movie4 | Movie5 |
|---|---|---|---|---|---|
| User1 | 4 | | 5 | 3 | 3 |
| User2 | | 4 | 3 | | 3 |
| User3 | 3 | 4 | | 5 | 4 |
| User4 | 2 | | 3 | 4 | |
| User5 | 4 | 3 | | 4 | 2 |

**FIGURE 1.** A simple example of a rating matrix of rating range 1 to 5 with $(i, j)$ entry of user $i$ rated the movie $j$.

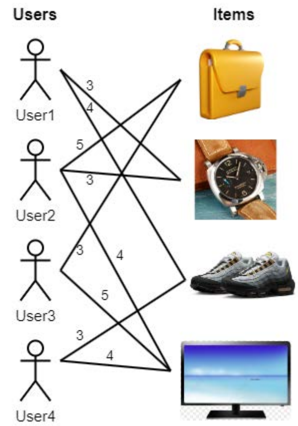



**FIGURE 2.** Rating networks visualizing user-item dynamics in bipartite structures.

Given for a set of users, $X = \{x_1, x_2, \ldots, x_m\}$, and for a set of items $Y = \{y_1, y_2, \ldots, y_n\}$. If the user has rated an item $R_{ij}$, which represents the rating given by a user $x_i$ on $y_j$. The main objective of the recommender system is to suggest a new item $y_b$ to the user $x_a$, where a particular item is not yet observed by the user.

Consider an example of the rating matrix presented in Fig. 1. The matrix is a rating matrix, with the values representing the ratings given by the user to an item. Each row of the matrix reflects each user, while the columns correspond to various items, such as movies. Each entry in the matrix represents the user's rating of the movie. The basic function of the recommender system is to forecast missing items in the rating matrix.

There are several strategies for handling the problem in recommender systems, mostly using content-based and collaborative filtering approaches. One of the most prevalent tactics used in collaborative filtering is matrix factorization (MF), which improves proposal quality while reducing time complexity. To increase the quality of proposals, we propose modeling the rating matrix as a complex network with community structures. We aim to incorporate community information into the matrix factorization technique concurrently. The rating matrix is represented as a bipartite graph to determine the community structures. A sample example is shown in Fig. 2, representing the scenario where the users purchase different products in e-commerce platforms, representing a bipartite graph.

## II. RELATED WORK

Hintz *et al.* introduce several latent feature models for matrix factorization techniques that are used for enhancing the quality of the recommendations [28]. Several matrix factorization algorithms are introduced to minimize the squared error [29]. In 2011, Linas *et al.* introduced a new recommendation algorithm that enhances matrix factorization by considering contextual factors [30]. This algorithm introduces extra parameters for how contextual factors interact with item ratings. The experiments conducted demonstrate that this approach yields results similar to the best existing methods, even those that are more intricate. Notably, this solution offers the advantage of being computationally efficient and allows for representing the interaction between context and items at various levels of detail.

Wei Xu *et al.* unveiled a unique version of NMF in 2003, offering a ground-breaking method for document clustering inside a given document corpus based on the non-negative factorization of the term-document matrix [31]. Documents are shown as a composite of these underlying subjects by employing the latent semantic space acquired through NMF, where each axis denotes the central theme of a particular document cluster. To guarantee that the rating profile of every user may be expressed as the additive linear combination of a canonical coordinate, non-negativity criteria are used in the linear model. Sheng Zhang *et al.* developed two versions of Non-negative Matrix Factorization in 2006 to achieve the limited linear modeling using an incomplete rating matrix [32]. A Bayesian approach to non-negative matrix factorization (NMF) in 2009 by Schmidt *et al.* is proposed, utilizing a normal likelihood and exponential priors [33]. An effective Gibbs sampler is derived to estimate the NMF components' posterior density. Additionally, an iterated conditional modes algorithm is presented, demonstrating comparable performance to utilizing the most recent NMF techniques and extracting visual features.

In 2008, Koren *et al.* and his team introduced a better way to recommend things by combining different types of feedback [34]. They called it SVD++, which is a fancy name for a method that improves recommendation accuracy and personalization. SVD++ analyzes explicit and implicit feedback, like what users click on or view. Doing this makes recommender systems work better because they can understand what users like more accurately and suggest things that match their tastes more effectively. In 2020, Sheng *et al.* introduced a new version of the SVD++ algorithm [35]. Noticed that recommendation systems often struggle when there isn't much data available. Different versions of the SVD algorithm have tried to tackle this issue, but they didn't see much improvement in the recommendation results. So, they came up with this new algorithm to address these

**TABLE 1.** Literature review of different authors with their approach and key findings.

| Author | Method | Keyfindings |
|---|---|---|
| Hintz *et al.* [28] | Matrix Factorization techniques | Several latent feature models for matrix factorization techniques that are used for enhancing the quality of recommendations. |
| Linas *et al.* [30] | Context-aware recommendation | Introduces extra parameters to account for how contextual factors interact with item ratings. |
| Wei Xu *et al.* [31] | Document clustering based on NMF | Documents are shown as a composite of these underlying subjects by employing the latent semantic space acquired through NMF, where each axis denotes the central theme of a particular document cluster. |
| Zhang *et al.* [32] | Non-negativity constrained linear model | Proposed limited linear modeling using an incomplete rating matrix. |
| Schmidst *et al.* [33] | Bayesian Non-Negative Matrix Factorization | A Bayesian approach to NMF is presented, based on the normal likelihood and exponential priors. |
| Koren *et al.* [34] | SVD++ | SVD++ works by taking into account both explicit feedback, like ratings, and implicit feedback, like what users click on or view. |
| Sheng *et al.* [35] | SVD++ application with time feature | Includes a special timing feature to adjust dynamically and evaluate recommendations |
| Wenchuan *et al.* [36] | UE-SVD++ | Looked closely at the ratings users give and used that information to build a special matrix called the user embedding matrix. |
| Aghdam *et al.* [37] | Asymmetric non-negative matrix factorization | Takes into account both user and item biases along with user-item interactions to enhance accuracy and recommendation quality. |
| Srilatha *et al.* [38] | Community-Based Matrix Factorization approach | Integrates community detection approach with matrix factorization method and finds out the incomplete matrix. |
| Zhang *et al.* [39] | Self-adaptive Louvain algorithm | Algorithm makes use of the small probability events principle to determine how many neighbors should be chosen at random. |
| Sayan *et al.* [40] | Distributed memory implementation of Louvain algorithm | Starts with a distributed graph input that has been randomly partitioned and then uses a number of heuristics to speed up the calculation of the various Louvain algorithm phases. |
| Maryam *et al.* [41] | Adaptive CUDA Louvain method | By using shared memory in GPU, and with minimum threads overhead is minimized. |

challenges. This version includes a special timing feature to adjust dynamically and uses measures like average absolute error, root mean square error, and standard average absolute error to evaluate recommendations. A new way to improve predicting ratings in collaborative filtering using SVD++ was suggested by Wenchuan *et al.* in 2020 [36]. A model called UE-SVD++ focuses on getting more detailed feedback from users. To do this, looked closely at the ratings users give and used that information to build a special matrix called the user embedding matrix. This matrix improves prediction accuracy by combining it with the already present user bias as well as additional parameters in SVD++. The FANMF method is designed to handle non-negative data that are unevenly distributed [37]. In real-world situations, data often show this uneven pattern, where the relationships between rows and columns aren't balanced. FANMF builds upon NMF to deal with these uneven scenarios. It takes into account both user and item biases along with user-item interactions to enhance accuracy and recommendation quality. User-item bias is defined as a user's innate inclinations for specific items or the intrinsic attractiveness of products to users, independent of their previous activities. In 2023, Srilatha *et al.* proposed an approach integrating matrix factorization and community detection where the appropriate number of communities are derived, and for each community, matrix factorization is applied [38]. The performance metric signifies that the recommendations are appropriate for the user to get a quality recommendation.

In 2018, Zhang *et al.* introduced an enhanced version of the Louvain algorithm [39]. The refined algorithm uses the small probability events principle to determine how many neighbors should be randomly chosen. The findings indicate that this enhanced version achieves partitioning results comparable to the original Louvain but at a faster pace. Notably, the algorithm also demonstrates robust performance on networks lacking distinct community structures. Sayan Ghosh *et al.* in 2018 described the architecture of a Louvain method distributed memory implementation meant for parallel community detection [40]. The approach starts with a distributed graph input that has been randomly partitioned and then uses several heuristics to speed up the calculation of the various Louvain algorithm phases. In 2020, a cutting-edge adaptive CUDA Louvain method algorithm was first presented by Maryam *et al.*, leveraging the power of GPU [41]. The matrix factorization approaches, and community detection method selected for our testing is briefly discussed in the section that follows. All the literature review is shown in a tabular format in Table. 1.

## III. METHODOLOGY

The following section provides a comprehensive explanation of each MF approach, including basic MF, NMF, SVD++, FANMF, and the Louvain community detection method.

### A. BASIC MATRIX FACTORIZATION (MF)

In the basic matrix factorization method, we consider a rating matrix $R$ of size $m \times n$, with $m$ users and $n$ items [42]. As the rating matrices are very huge, there will be many missing ratings, and by using the matrix factorization method, we define those unknown ratings. Initially, we create the user and item latent feature matrices $P$ and $Q$ of sizes $m \times k$

and $n \times k$, respectively, with random values. The number of latent characteristics is denoted by $k$, and its values vary. Using the dot product of the latent feature matrices $P$ and $Q$, the predicted rating matrix $\widetilde{R}$ is constructed as

$$\widetilde{R} = PQ^T. \tag{1}$$

The given rating matrix $R$ is the approximation of the latent feature matrices $PQ^T$ and is shown as

$$R \approx PQ^T. \tag{2}$$

The deviation between the original and the predicted ratings is given by

$$r_{mn} \approx p_m q_n^T. \tag{3}$$

A regularization term $\beta$ is added to the minimized squared error to avoid overfitting as in (4).

$$\min \sum_{m,n} (r_{mn} - p_m q_n^T)^2 + \beta(\parallel p_m \parallel^2 + \parallel q_n \parallel^2). \tag{4}$$

Using a constant $\beta$, the impact of the overfitting is controlled. $\parallel . \parallel$ is the Frobenius norm. Stochastic gradient descent is used to calculate the prediction error for each rating in the data as shown below

$$e_{mn} = r_{mn} - p_m q_n^T. \tag{5}$$

The entries of the predicted rating matrices as shown in (6) are updated to minimize the squared error by adding the learning rate $\alpha$ to the latent features.

$$\begin{aligned} q_n &\longleftarrow q_n + \alpha(e_{mn} p_m - \beta q_n) \\ p_m &\longleftarrow p_m + \alpha(e_{mn} q_n - \beta p_m) \end{aligned} \tag{6}$$

The above process is repeated until a fixed number of iterations or when the error becomes zero. The difference between the original and the predicted rating matrices can be obtained by

$$\text{RMSE} = \sqrt{\frac{1}{T} \sum (r_{mn} - \widetilde{r}_{mn})^2}, \tag{7}$$

where $T$ is the quantity of predictions, $r_{mn}$ is the original rating, and $\widetilde{r}_{mn}$ is the predicted rating.

The basic matrix factorization method has a time complexity of $\mathcal{O}(mnk)$, where there are $m$ users, $n$ items, and $k$ latent features.

### *B. NON-NEGATIVE MATRIX FACTORIZATION (NMF)*

NMF is a popular dimensionality reduction approach that involves taking a non-negative matrix and splitting it into the product of two non-negative matrices of lower rank. Paatero and Tapper defined positive matrix factorization, which helped to establish NMF [43]. Following the seminal work by Lee and Seng, NMF rapidly gained widespread recognition and popularity in the field [44]. Two non-negative latent feature matrices, $P$ and $Q$, are created from the partitioning of the rating matrix $R$. The product of these two latent feature matrices represents the estimation of the non-negative matrix $R$, as (2) illustrates.

In this context, $R$ represents a rating matrix of dimensions $m \times n$, $k$ denotes the latent features to be extracted. The latent feature matrices $P$ $(m \times k)$ and $Q$ $(n \times k)$ are estimated with the consideration that $k \leq (m, n)$. The values of $P$ and $Q$ are updated by using multiplicative update rules [44] as shown below.

$$\begin{aligned} P &= P \cdot \times ((R \cdot / (P \times Q + (R == 0))) \times Q^T) \\ Q &= Q \cdot \times (P^T \times (R \cdot / (P \times Q + (R == 0)))), \end{aligned}$$

where $P \cdot \times Q$ is the dot product of $P$ and $Q$, $P \cdot /Q$ is the dot division of $P$ and $Q$ which is element wise division. $P \times Q$ is the product of two matrices $P$ and $Q$. $P^T$, and $Q^T$ are the transpose of the matrices P and Q. To avoid division by zero, the denominator contains the expression $R == 0$. Values will be adjusted upon applying multiplicative revised rules for $P$ and $Q$. The predicted rating matrix $(PQ^T)$, known as $\widetilde{R}$, is produced by computing a dot product from the updated latent feature matrices. The RMSE value is the variance among the original rating matrix $R$ and the predicted rating matrix $\widetilde{R}$, as given in (7). The non-negative matrix factorization approach has a temporal complexity of $\mathcal{O}(mnk)$ for $m$ users, $n$ items, and $k$ latent features.

### *C. SVD++*

This method is the advanced version of singular value decomposition [34]. In updating the latent feature matrices, implicit feedback is added to the user's latent feature matrix $P$ [45], [46]. Implicit feedback for the user is the user feedback matrix $U$, and the item is the item feedback matrix $I$. The calculation of the user feedback matrix is $U = [u_{mn}]$ $\forall(x_m, y_n)$ will be 1, if $r_{mn}$ have a rating by user or else 0. The size of the user feedback matrix will be of the same size as the original rating matrix $R$. Each and every entry of $U$ is filled as, let $Y_j$ be the item that the user $x_i$ has rated, each non-zero entry in the $j^{th}$ row of $U$ is calculated as $\frac{1}{\sqrt{|Y_j|}}$. The item feedback matrix will be the same as the latent feature matrix. The dot product of $U$ and $I$ is added to the user latent feature matrix as shown in (8) and is defined as the predicted rating matrix.

$$\widetilde{R} = [(P + UI).Q^T]. \tag{8}$$

Then, the difference between the original and the predicted rating matrices is calculated as RMSE value as shown in (7). The SVD++ method has a time complexity of $\mathcal{O}(mnk)$, where there are $m$ users, $n$ items, and $k$ latent features.

### *D. FACTORIZED ASYMMETRIC NON-NEGATIVE MATRIX FACTORIZATION (FANMF)*

The FANMF method came into existence from 2019 [47]. This technique is designed to handle non-negative and asymmetric data. The difference between NMF and FANMF is that FANMF improves recommendation quality by considering both user and item bias and user-item interactions [48]. User and item bias refers to user's preferences for certain items regardless of their past interactions or behaviors. The latent feature matrices $P$ and $Q$ are updated by using the multiplicative update values [44]. The deviation between the original rating matrix $R$ and the predicted rating matrix $\widetilde{R}$ is calculated as RMSE as illustrated in (7). The FANMF method has a time complexity of $\mathcal{O}(mnk)$, where there are $m$ users, $n$ items, and $k$ latent features.

### E. LOUVAIN COMMUNITY DETECTION METHOD

Louvain community detection method is a prominent technique developed in 2008 by Blondel *et al.* [49]. This method is used to identify clusters or communities within intricate networks. It is extensively employed to unveil the organizational structure of complex networks, enabling insights into relationships, interactions, and functional modules. Several methods are used to assess whether the quality of the structure of the communities is effective or not, which drives the concept of modularity. To enhance the quality of the community structures, a quality metric named modularity score is calculated by iteratively merging and shifting nodes between the communities [50]. The Louvain community detection algorithm is defined in three steps. In the first step, form the communities of size 1, and in the second step, find the modularity score within the community. in the last step, shift the nodes to the nearby communities by comparing them with the modularity score value. This technique is continued until the modularity score shows no obvious change.

The modularity score is calculated using the formula as shown in (9).

$$\text{modularity}(C) = \Sigma_c \left[ \frac{E_c}{E} - \left( \frac{d_c}{2E} \right)^2 \right], \tag{9}$$

where $d_c$ is the degree of community c, $E$ is the number of edges in the graph $G$, $E_c$ is the number of edges in the community.

The modularity score evaluates the network's efficacy by modularity $C$. The range of modularity is between $-1$ to $+1$. The negative modularity indicates that the communities are not appropriately defined, and the positive modularity score defines the communities as well-structured. The Louvain community detection algorithm is not only efficient, but it also has another advantage of expandability. These advantages are very useful for working with large and diverse networks where the traditional approaches are undergoing many issues of computation [51]. The Louvain method is very important in networks as it can process fast and give accurate and well-structured communities. Moreover, community detection methods are used in various domains, including social networks, biological networks, and recommender systems. Due to its highly adaptable nature, the Louvain community detection method is applied in different domains in this modern world [52], [53]. The computing capabilities play a crucial role in the useful insights of real-world networks.

## IV. PROPOSED METHOD

To provide users with appropriate recommendations, matrix factorization emerges as one of the highly effective techniques employed. The sheer magnitude of the data available between users and items necessitates the construction of a rating matrix, which can be quite extensive. The evaluation of these vast matrices requires substantial computation time. To address this concern and enhance the recommendation process, we have proposed an integrated approach of matrix factorization method with the Louvain community detection method. Here, any kind of matrix factorization method is suitable for this context. The utilization of the matrix factorization method allows for the creation of effective community structures. As a result, we put forth the proposed approach that integrates the matrix factorization method and the Louvain community detection method. The overall procedure followed using this approach is shown in Fig. 3.

The procedure that is followed by the proposed method is as follows:

- Step 1: A rating matrix $(RM)$ is constructed by collecting the information from users, items, and their respective ratings. Users are taken on one axis, and the items are taken on the other. The values in the matrix are filled by considering the interactions between the users and the items, i.e., ratings.
- Step 2: Construct a bipartite network $BP$ where the nodes represent users and items, and the edges represent ratings, serving as weighted connections between them.
- Step 3: Use the created bipartite graph to find communities using the Louvain community detection method of size $c$ for $BP$. The size of the bipartite network will be of the size of communities that are divided using the Louvain community detection method. Let $BP$= $\{BP_1, BP_2, \cdots, BP_c\}$.
- Step 4: Obtain a rating matrix from each community of size $c$ divided from the bipartite graph. For each bipartite network divided by using the Louvain community detection method, a rating matrix will be obtained, i.e., $RM_1, RM_2, \cdots, RM_c$.
- Step 5: In parallel, apply the matrix factorization methods to each rating matrix obtained in the previous step. As a result, predicted rating matrices $\widetilde{RM}_1, \widetilde{RM}_2, \cdots, \widetilde{RM}_c$ are obtained with the same size as the number of communities.
- Step 6: Obtain a single comprehensive predicted rating matrix $(\widetilde{RM})$, combining all the generated predicted rating matrices.
- Step 7: Calculate the recommendation accuracy by using the RMSE evaluation metric to measure the difference between the original rating matrix that is initially taken at step 1 $(RM)$ and the predicted rating matrix that is obtained in the previous step $(\widetilde{RM})$.

The time complexity of the MF method is $\mathcal{O}(mnk)$, where there are $m$ users, $n$ items, and $k$ latent features. The time complexity of the Louvain community detection method is $\mathcal{O}(n \log n)$, where $n$ is the number of nodes. In our case, the graph is a bipartite graph, and the number of nodes will be the sum of users and items, i.e., $(m + n)$. Hence the time

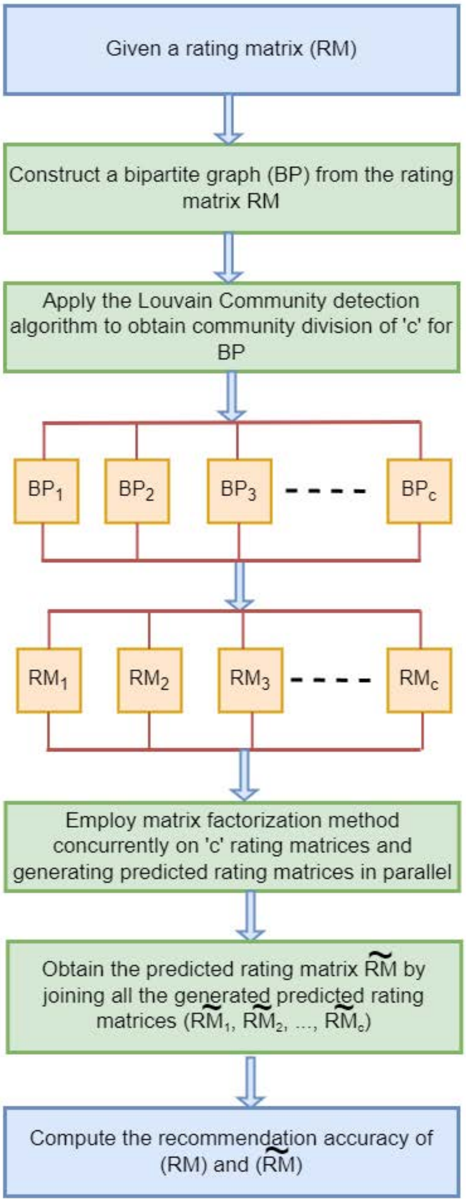



**FIGURE 3.** A snippet of the proposed approach.

complexity will be $\mathcal{O}((m+n)\log{(m+n)})$.

In our analysis, as there are $c$ community structures, we will be getting $c$ rating matrices. For each rating matrix, time complexity can be analyzed as $\mathcal{O}(m_1 n_1 k_1)$, $\mathcal{O}(m_2 n_2 k_2)$,$\cdots$, $\mathcal{O}(m_c n_c k_c)$; and considered the maximum of these i.e., $\mathcal{O}(m_l n_l k_l)$. Therefore, the overall time complexity of our approach will be $\mathcal{O}((m+n)\log{(m+n)}) + \mathcal{O}(m_l n_l k_l)$. Similarly, for the NMF, SVD++, and FANMF methods, the time complexity is defined as $\mathcal{O}(mnk)$. Hence, the time complexity of the integrated approach of any kind of matrix factorization with the Louvain community detection method will be $\mathcal{O}((m+n)\log{(m+n)}) + \mathcal{O}(m_l n_l k_l)$. Moving ahead, our attention will shift to performing experimental analysis, where we will explore the datasets in accordance with the approach mentioned before.

## V. EXPERIMENTAL RESULTS

For implementing our proposed method, we have taken four different datasets namely food recommendation, book-crossing, anime recommendation, and restaurant recommendation downloaded from Kaggle. The dataset statistics for the four datasets are produced in Table 2. These datasets are applied to our proposed method to determine the RMSE value and the time taken to evaluate the algorithm. We have evaluated the performance of the recommendation by using the RMSE measure as it will penalize large errors due to its calculation of squaring operation.

**TABLE 2.** Dataset Statistics for four different datasets, namely Food Recommendation, Book-crossing, Anime Recommendation, and Restaurant Recommendation.

| Dataset | Users | Items | Ratings | Rating Range | Sparsity |
|---|---|---|---|---|---|
| **Food Recommendation** | 100 | 309 | 508 | 1-10 | 0.983 |
| **Book-crossing** | 1295 | 17384 | 62656 | 1-10 | 0.997 |
| **Anime Recommendation** | 4714 | 7157 | 419943 | 1-10 | 0.987 |
| **Restaurant Recommendation** | 268 | 130 | 1161 | 0-2 | 0.974 |

The simulations are run at the central processing unit of $11^{th}$ Gen Intel (R) Core (TM) i9 - 11900, with CPU running at 2.50GHz with 64GB RAM of system type 64-bit operating system. Anaconda software is used to compute the community structures and integrates with the matrix factorization method to fill in the missing ratings in the matrix. All the visualization plots are drawn using Origin Pro software.

Fig. 4 displays the rating distribution of the food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets. The figure shows the information of the ratings that are distributed in the datasets. The X-axis acts for the distinct ratings, whereas the count of each rating in the dataset is shown on the Y-axis. In the food recommendation dataset, we can observe that 63 had the highest count for a 3 rating, and 38 had the lowest count for an 8 rating. In the book-crossing dataset, the highest count of 15629 is for an 8 rating, and the lowest count of 160 is for a 1 rating. In the same way for the anime recommendation dataset, the highest count of 106782 is for an 8 rating and the lowest count of 1278 is for a 1 rating. Similarly, for the restaurant recommendation dataset, the highest count of 486 is for a 2 rating and the lowest count of 254 is for a 0 rating.

### A. DISCUSSIONS ON RMSE RESULTS

Fig. 5 shows the RMSE value on four datasets for 25 communities and different latent features for $k$ = 10, 20, and 30 for the basic MF method. It is observed in the figure that for the food recommendation dataset, without using community detection at $c = 1$ for the basic MF method, the RMSE value is high. When applying the community detection method along

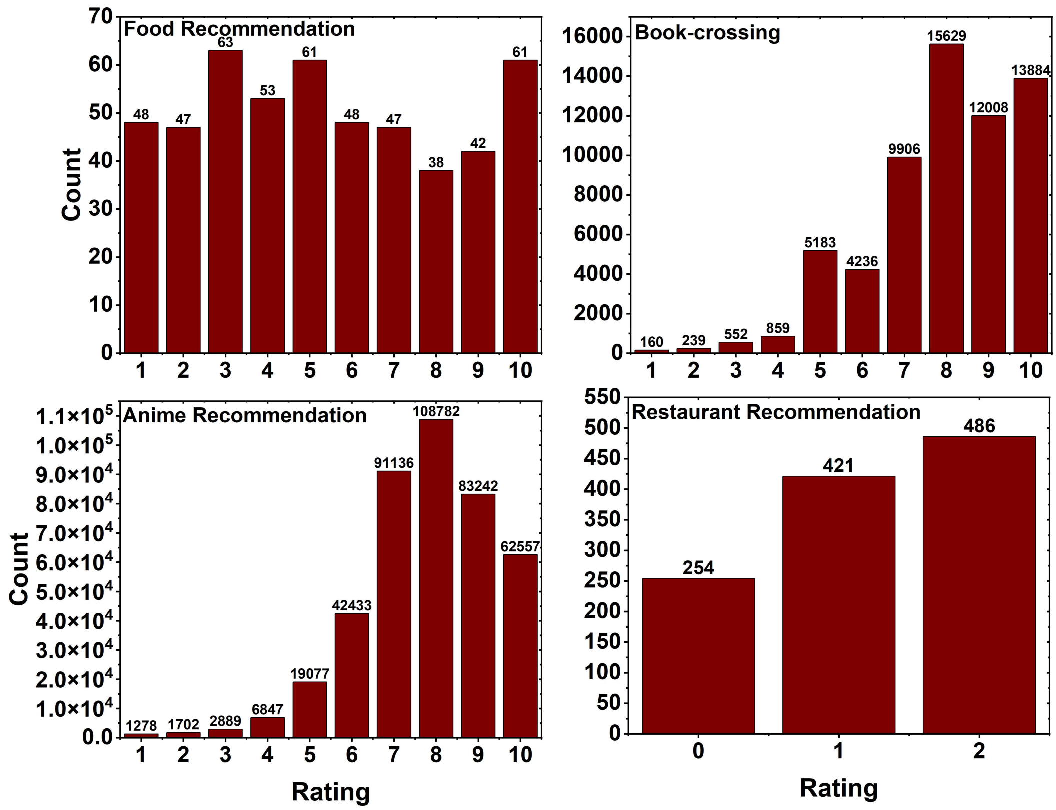



**FIGURE 4.** Rating distribution plots for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

with the basic MF, there is a decrease in the RMSE value as the communities increase. We can see a clear difference where only the matrix factorization method is applied and with the integration of the matrix factorization method with the Louvain method. The RMSE value is very high when there is no community division for a community value of 1. As the Louvain community detection method is applied and communities are increased, there is a drastic change in the RMSE value. Furthermore, we observe that after a certain number of communities, the RMSE value remains constant. This indicates the value of the better community division for the network. For the book-crossing dataset, we observed that as the latent features were increased, there was a decrease in RMSE value. Initially, there is a less RMSE value seen where there is no community division. When only the matrix factorization approach is employed, the RMSE value decreases in comparison to employing the community detection method, and the RMSE value decreases as the number of latent features and communities rises. We may observe that the RMSE value falls with the number of communities compared to the value obtained before the community split. In the anime recommendation dataset, it is observed that without using community division, the RMSE value is very high. After integrating the matrix factorization approach with the community division Louvain method, the RMSE value decreases as we increase the number of communities. Similarly, for the restaurant recommendation dataset, it is observed that there is a high RMSE value if only matrix factorization is applied. When integrated with the Louvain method there is a decrease in the RMSE value as the communities increase. We can say that by using the parallel approach with the basic MF and the Louvain community detection method, there is a better RMSE value for different communities. It is observed in all the datasets that the RMSE value is less for less number of latent features. Hence, the recommendations for the users will be accurate using this approach.

Fig. 6 shows the RMSE value on four datasets for 25 communities and different latent features for $k = 10$, 20, and 30 for the NMF method. It is observed in the figure that for the food recommendation dataset, without using community detection at $c = 1$ for the NMF method the RMSE value is high. When applying the community detection method along with the Louvain method, there is a decrease in the RMSE value as the communities increase. We can see a clear difference where only the matrix factorization method is applied and with the integration of the matrix factorization method with the Louvain community detection method is applied. The RMSE value is very high when there is no community division for a community value of 1. As the

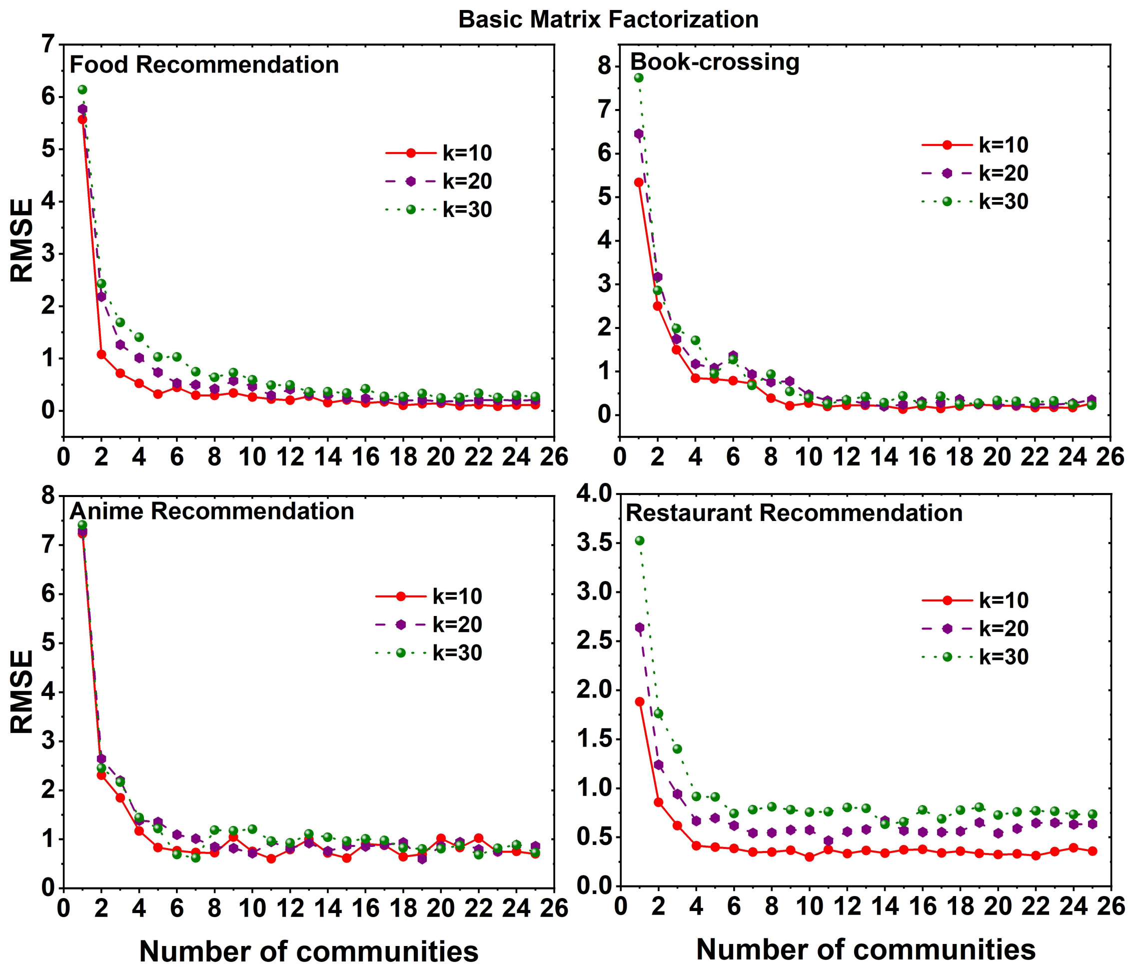



**FIGURE 5.** Examining the RMSE metrics for the basic matrix factorization method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

Louvain community detection method is applied and communities are increased there is a drastic change in the RMSE value. Furthermore, we observe that after a certain number of communities, the RMSE value remains constant. This indicates the value of the better community division that is for the network. For the book-crossing dataset we can observe that as the number of latent features is increased, there is a decrease in RMSE value. Initially, there is a less RMSE value seen where there is no community division. When only the matrix factorization approach is employed, the RMSE value decreases in comparison to employing the community detection method, and the RMSE value decreases as the number of latent features and communities rises. We can see that the RMSE value decreases with the number of communities compared to the value obtained prior to the community split. In the anime recommendation dataset, it is observed that there are several ups and downs for the RMSE value as the communities increase. It is observed in the figure that while applying only NMF, the RMSE value is very high compared to when NMF is integrated with the Louvain method. Similarly, in the restaurant recommendation dataset, the RMSE value is high when only the NMF approach is applied. When the NMF approach is applied with the Louvain method, a decrease in the RMSE value is observed. We can say that by using the parallel approach with Non-Negative Matrix Factorization and the Louvain community detection method, there is a better RMSE value for different communities. It is observed in all the datasets that as the number of latent features is increased, the RMSE value decreases. Hence, the better RMSE value is observed in the more latent feature. Therefore, the recommendations for the users will be accurate using this approach.

Fig. 7 shows the RMSE value on four datasets for 25 communities and different latent features for $k = 10$, 20, and 30 for the SVD++ method. It is observed in the figure that, for all the datasets, RMSE values are high when using only the SVD++ method. When the SVD++ method is integrated with the Louvain community division, the RMSE value falls.

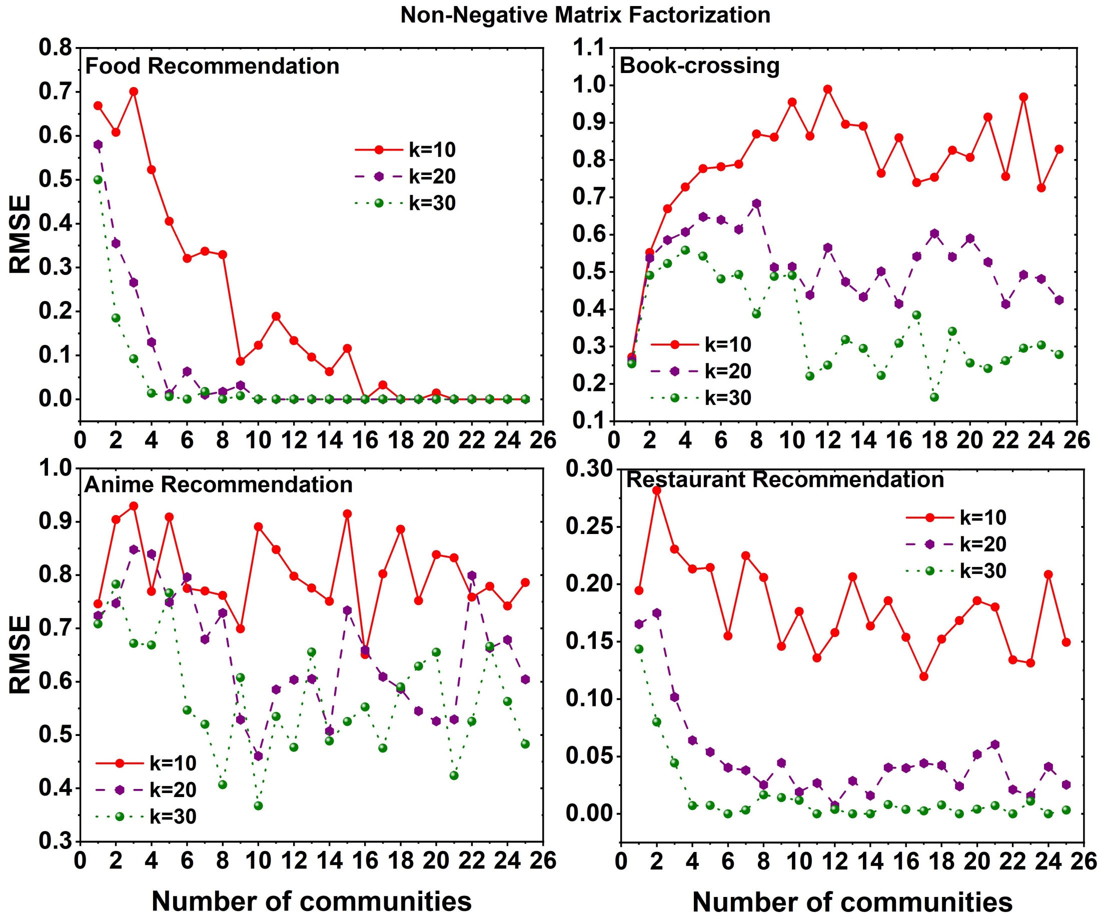



**FIGURE 6.** Examining the RMSE metrics for the NMF method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

As the communities increase, the RMSE value decreases, and after a certain number of communities, the RMSE value remains constant. We can say that by using the parallel approach with the basic MF and the Louvain community detection method, there is a better RMSE value for different communities. It is observed in all the datasets that the RMSE value is less for less number of latent features.

Fig. 8 shows the RMSE value on four datasets for 25 communities and different latent features for $k$ = 10, 20, and 30 for the FANMF method. In the food recommendation dataset, it is observed that there is more RMSE value at community 1. As the communities increased, we observed that there was a drastic fall in the RMSE value. After certain community iterations, the RMSE value is maintained constant. The number of latent features is also iterated and the more latent feature value gives the better RMSE value. For the book-crossing dataset, there are severe ups and falls for the RMSE value as the communities increased. As the latent features are increased, the RMSE value is decreased, and we observed that the RMSE value that we got at community 1 is higher than we observed when the communities are iterated for $k$ value 30. In the anime recommendation dataset, it is observed that there are severe ups and falls for the RMSE value as the number of latent features varies. The higher RMSE value is observed when only the FANMF approach is applied. When it is integrated with the Louvain method, the RMSE value observed is much less. As the number of latent features increases, the RMSE value decreases for the increase in the number of communities. Similarly, for the restaurant recommendation dataset, it is observed that the RMSE value varies for different communities as well as the increase in the number of latent features. It is observed that a high RMSE value is observed when only the FANMF method is applied. When the integration of the Louvain method is applied the RMSE value has severe ups and falls for $k$ value 1, and there is a drastic fall for RMSE value for $k$ values 20 and 30. It is observed in all four datasets that, the RMSE value is low when the latent feature value is high. By using the parallel approach we observe that we get a better RMSE value than by only using the FANMF method.

Table 3 provides the comparison of the results of the RMSE values for four different datasets for four different MF

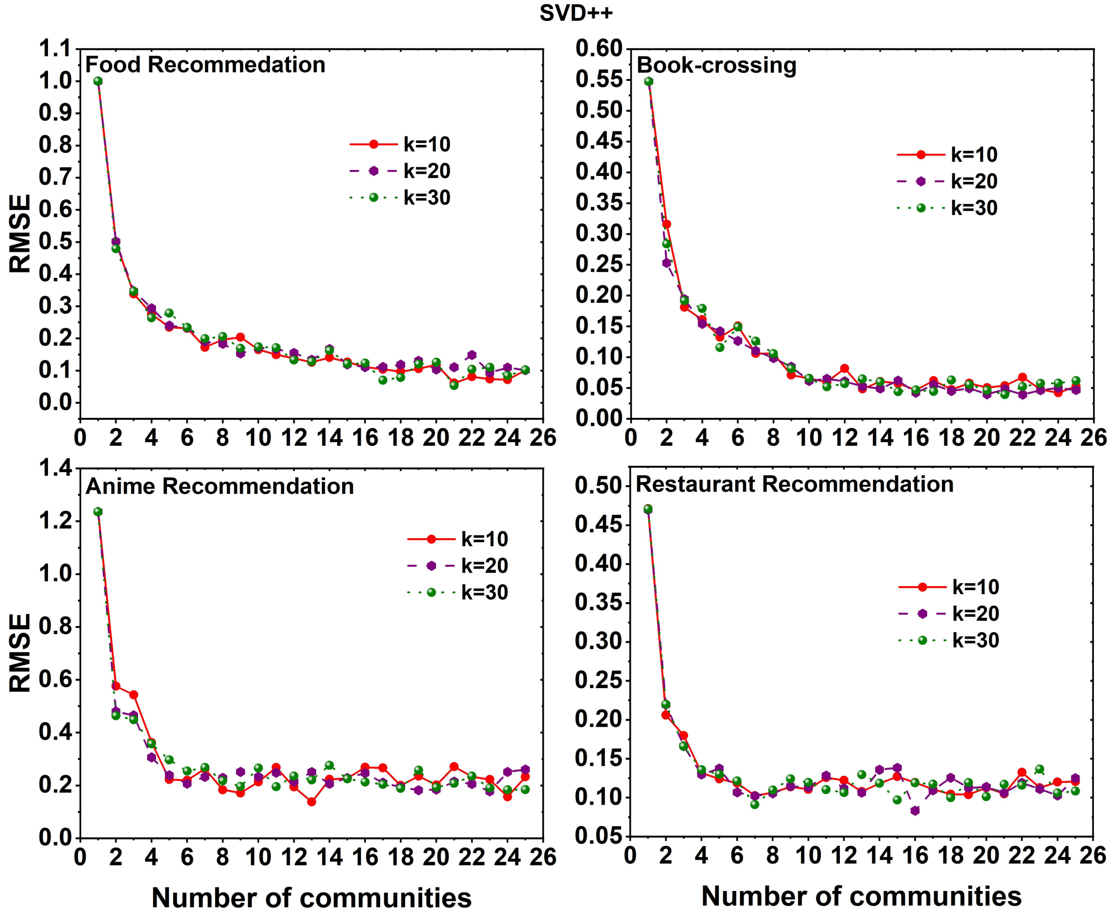



**FIGURE 7.** Examining the RMSE metrics for the SVD++ method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

**TABLE 3.** Comparison of RMSE values for four different datasets on different MF methods by integrating Louvain and MF approaches Vs by not integrating Louvain and MF approaches.

| | without using community | | | | with using community (number of communities) | | | |
|---|---|---|---|---|---|---|---|---|
| **MF Method (→) / Dataset (↓)** | **Basic MF** | **NMF** | **SVD++** | **FANMF** | **Basic MF** | **NMF** | **SVD++** | **FANMF** |
| **Food Recommendation** | 6.13 | 0.66 | 1.0 | 0.67 | 0.08 (23) | 0.007 (9) | 0.06 (21) | 0.0001 (6) |
| **Book-crossing** | 7.73 | 0.27 | 0.54 | 0.27 | 0.13 (15) | 0.16 (18) | 0.04 (16) | 0.21 (16) |
| **Anime Recommendation** | 7.41 | 0.74 | 1.23 | 0.74 | 0.64 (18) | 0.36 (10) | 0.15 (24) | 0.37 (25) |
| **Restaurant Recommendation** | 3.52 | 0.19 | 0.47 | 0.27 | 0.29 (10) | 0.0001 (6) | 0.09 (7) | 0.0001 (24) |

methods by not using and using the community approach. The table provides a detailed analysis of the RMSE values that are obtained without using the community approach in the MF method and by integrating the community approach with the MF method. In brackets, we have given the community number at which the RMSE value is low by using the proposed approach. It can be seen in the table that when we are not using the community approach the RMSE value is high and when by using the Louvain community approach that integrates with the MF method, we observe a less RMSE value. For instance, we observe that the food recommendation dataset shows a better score of RMSE when divided into 23 communities for the basic MF method, 9 communities for the NMF method, 21 communities for the SVD++ method, and 6 communities for the FANMF method. For the book-crossing dataset, we observe a better score when divided into 15 communities for the basic MF method, 18 communities for the NMF method, and 16 communities for the SVD++ and the FANMF methods. In the anime recommendation dataset, it is observed that a better score of RMSE value is seen at 18 communities for the basic MF method, 10 communities for the NMF method, 24 commu-

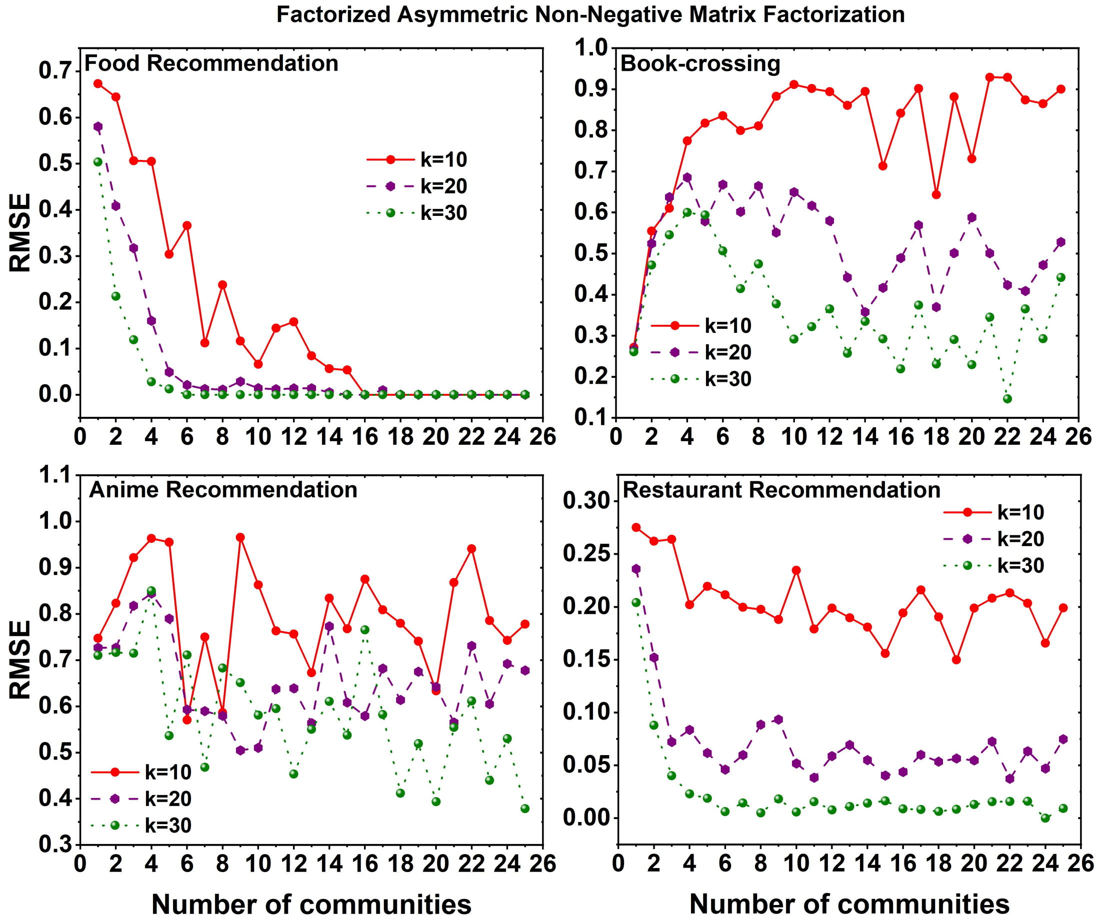



**FIGURE 8.** Examining the RMSE metrics for the FANMF method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

nities for the SVD++ method, and at 25 communities for the FANMF method. Similarly, for the restaurant recommendation dataset, the better score of RMSE value is observed at 10 communities for the basic MF method, 6 communities for the NMF method, 7 communities for the SVD++ method, and at 24 communities for the FANMF method. From the four networks, we observe that the Louvain community approach integrated with the MF approach gives a better result than not using the community approach. Thus, we can say that when the community approach is integrated with MF outperforms the non-utilization of the community approach with the MF method.

### B. DISCUSSIONS ON COMPUTATIONAL TIME

Fig. 9 displays the total time required to evaluate the basic MF approach and the community detection method in seconds. The total time is the sum of the time required for community division and calculating the RMSE value. It is observed from the figure that, for all the datasets, the time taken for computation without community division at $c$ value 1 community is more compared to the time taken by integrating the matrix factorization method and community division. In the food recommendation dataset, the value at 1 community is more, and when the communities are increased, for all the different latent features the value remains constant. After a given number of communities, the computation time decreases. In the book-crossing, anime recommendation, and restaurant recommendation datasets, the time taken is longer when only the matrix factorization approach is used. When combined with community division, the time required for all of the various latent properties decreases dramatically. In terms of time, the optimal number of communities may not be consistent across all networks nonetheless, following a specific community division, the result is obtained in a fraction of a second.

Fig. 10 shows the total time taken to evaluate the Non-Negative Matrix Factorization method along with the community detection method in the assessment of seconds. The total time is the sum of the time required for community division and calculating the RMSE value. It is observed from the figure that, for food recommendation and book-crossing datasets the time taken without using community

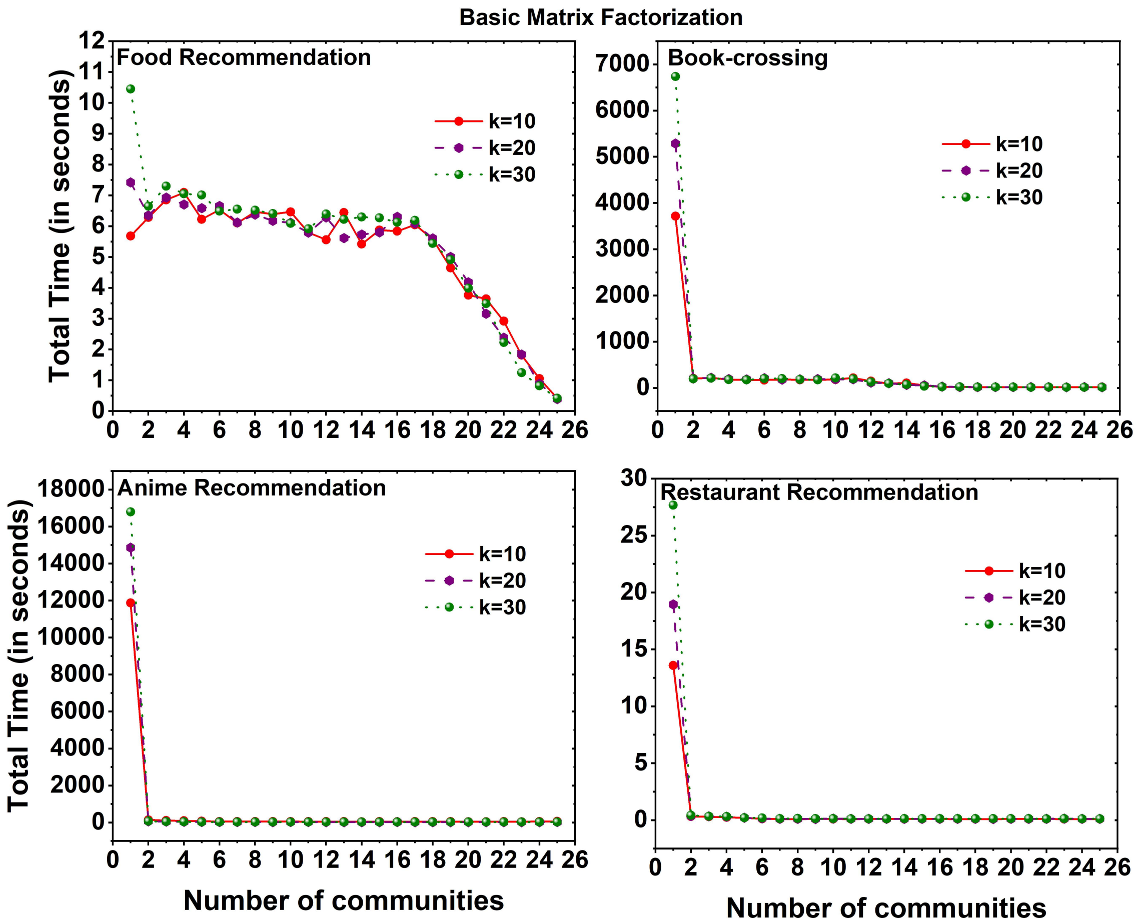



**FIGURE 9.** Comparing computational time for basic matrix factorization method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

division is less for $c$ value 1 community. As the communities increased, the time taken falls for all different latent features. We can also observe that the time taken without community division is more after a certain number of communities which indicates community division is preferable to get the recommendations in less time. In the food recommendation dataset, there is a continuous fall in time as the communities increased. In the book-crossing dataset, the time taken stays constant beyond a predetermined number of community divides. In the anime recommendation and restaurant recommendation datasets, the time taken at $c$ value 1 community is less compared to while the communities are iterated. As the time taken is not much more we get a better RMSE value compared to while only using the NMF approach. Even if the integrated approach takes more time than only using the NMF approach, the RMSE value is very less. In terms of time, the best number of communities may not be uniform for all the networks, after a certain community division it will be just a fraction of seconds we are getting the result.

Fig. 11 depicts the entire time required to evaluate the SVD++ technique and the community detection approach in seconds. The total time is the sum of the time required for community division and calculating the RMSE value. It is seen in the figure that, for the food recommendation the time taken at community 1 has less value. By using the integration with the SVD++ method with the Louvain community detection method, initially, at community 2, the time taken is more. As the number of communities increases the RMSE value falls and after a certain number of communities the RMSE value reaches down. We can also observe that as the number of latent features increased, the RMSE value also decreased. In the book-crossing dataset, the RMSE value at community 1 is less compared to community 2. After a certain number of communities iterated, the RMSE value decreases compared to the value at community 1 and remains constant. For all the different latent features that are iterated, we observe that the more latent feature value has less time taken. It is seen in the figure for the anime recommendation dataset, more time is taken at $c$ value at community 1. As the number of communities increases, when the SVD++ method is integrated with the Louvain method the time taken is very less. We get better results for time when the latent feature

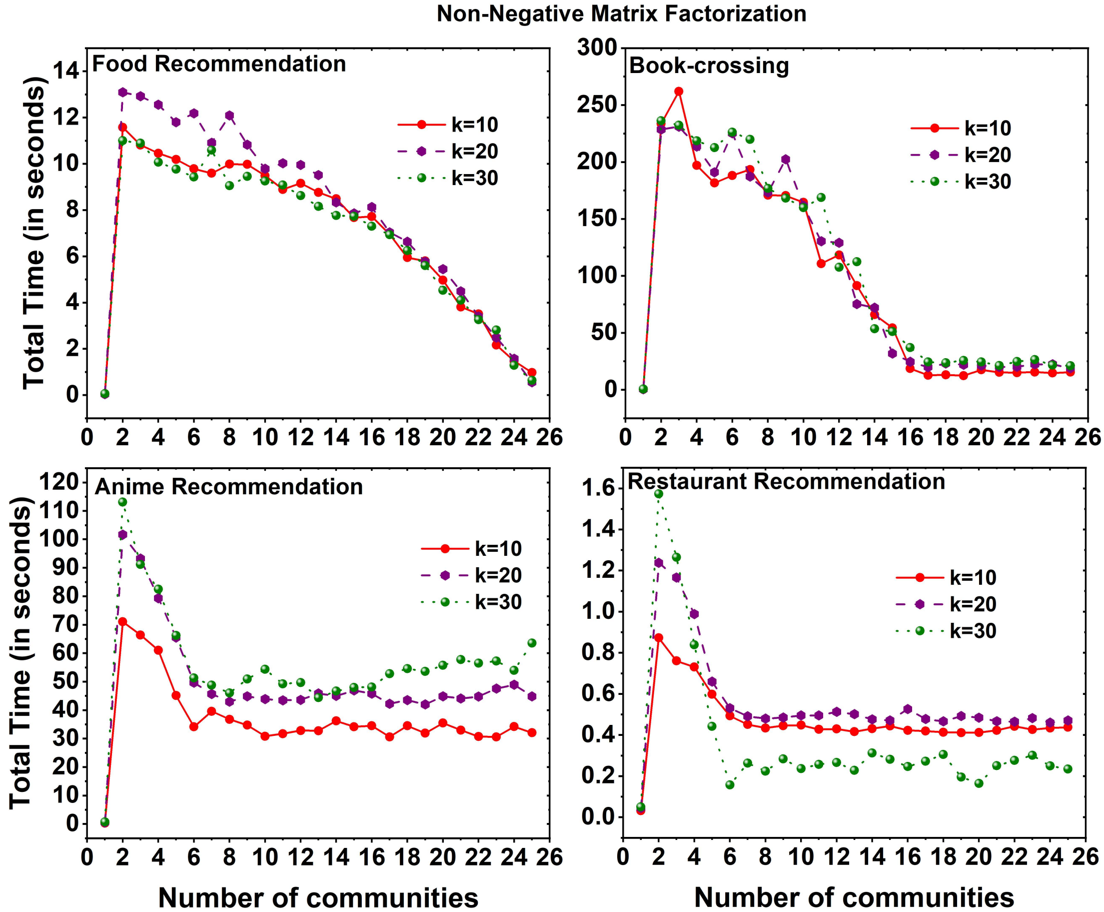



**FIGURE 10.** Comparing computational time for NMF method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

value is low. For the restaurant recommendation dataset, it is observed that when only the SVD++ method is applied for the $c$ value at community 1 the time taken is less compared to at $c$ value for community 2. As the number of communities increased we observed that the time taken was less and finally, after a certain number of communities the time taken varied less and remained constant.

Fig. 12 displays the entire time required to evaluate the FANMF technique and the community detection method in seconds. The total time is the sum of the time required for community division and calculating the RMSE value. It is observed in the figure from all the datasets that, the time taken for evaluation without using community division takes less time. As the communities increased we observed that the time taken decreased gradually. It is also observed in the book-crossing dataset that after certain communities, the time taken is maintained constant. It is observed in the anime recommendation dataset, that after a certain number of community divisions, the time taken is very less, and at community 7, we can see a fall in time than the time taken at $c$ value for community 1. But for the restaurant recommendation dataset, the time taken by using only the FANMF method is less than by using the integrated approach at community 2. As the communities increase the time taken is reduced and maintains a constant time for all the different latent features. By using this integrated approach of the FANMF method with the Louvain community detection method, the time taken for assessment is better than by using only the matrix factorization approach.

Table 4 provides the time assessment for the four different datasets on four different MF methods when integrating the Louvain approach with the MF method Vs by not integrating with the Louvain approach and the MF method. The table provides a detailed analysis of the time taken without using the community approach in the MF method and by integrating the community approach with the MF method. In brackets, we have given the community number at which the time taken is low by using the proposed approach. The time taken for the assessment is the sum of the community time and the RMSE time. It is seen in the table that when we use only the MF approach, it takes more time for computation. When the Louvain community detection method is integrated with the MF approach it takes less time for computation. Not only getting the less RMSE value the computation by using

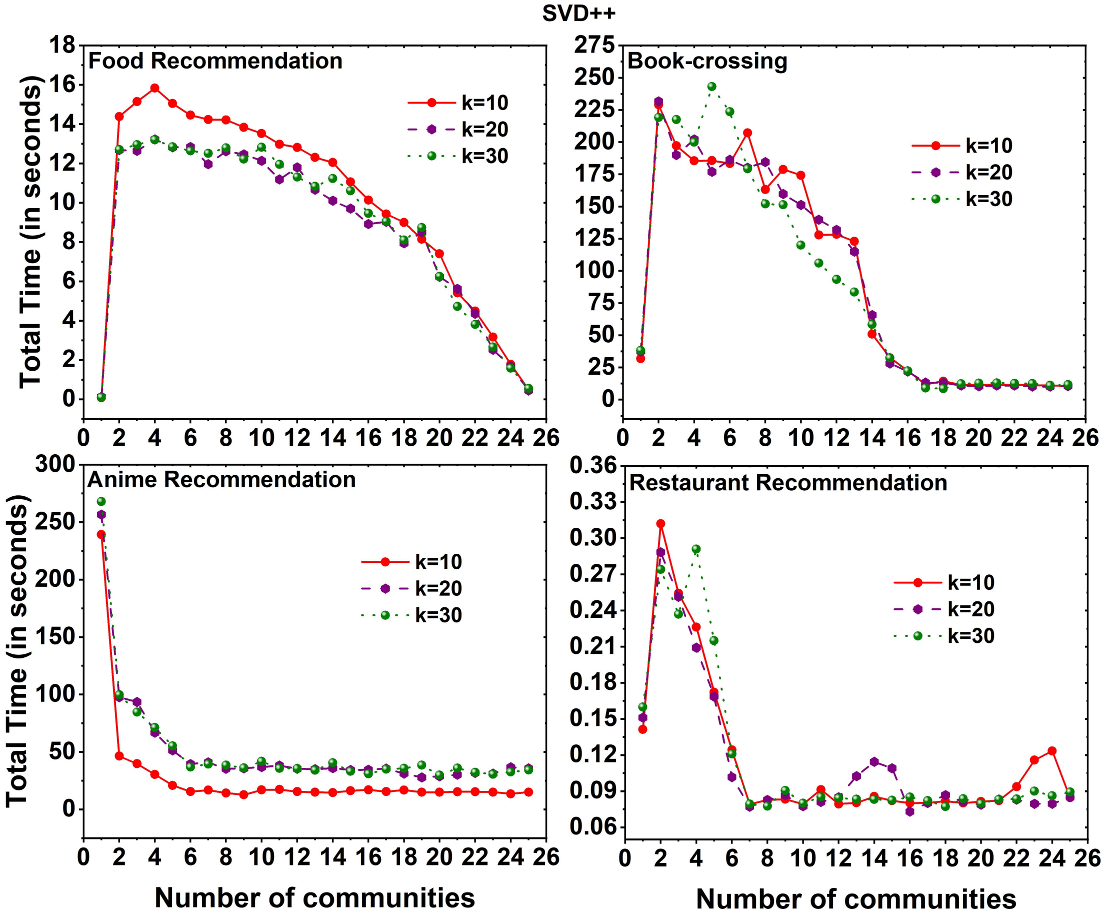



**FIGURE 11.** Comparing computational time for SVD++ method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

**TABLE 4.** Comparison of time (in seconds) for four different datasets on different MF methods by integrating Louvain and MF approaches Vs by not integrating Louvain and MF approaches.

| | without using community | | | | with using community (number of communities) | | | |
|---|---|---|---|---|---|---|---|---|
| **MF Method (→) / Dataset (↓)** | **Basic MF** | **NMF** | **SVD++** | **FANMF** | **Basic MF** | **NMF** | **SVD++** | **FANMF** |
| **Food Recommendation** | 10.44 | 0.06 | 0.12 | 0.07 | 0.39 (25) | 0.55 (25) | 0.44 (25) | 0.51 (25) |
| **Book-crossing** | 6737.25 | 0.61 | 38.21 | 8.3 | 15.75 (25) | 12.43 (19) | 10.52 (25) | 10.50 (23) |
| **Anime Recommendation** | 16789.64 | 0.71 | 267.92 | 19.02 | 18.26 (13) | 30.59 (23) | 13.52 (24) | 10.80 (19) |
| **Restaurant Recommendation** | 27.67 | 0.05 | 0.15 | 0.04 | 0.08 (24) | 0.15 (6) | 0.07 (7) | 0.09 (7) |

this method takes less time for computation. Hence, we say that our proposed approach is better in terms of calculating the RMSE value within less time.

## VI. CONCLUSION AND FUTUREWORK

Using parallel computing, this research aimed to accelerate two crucial data analysis techniques, matrix factorization and Louvain community detection. Through harnessing the capabilities of parallel processing, we successfully showcased substantial enhancements in the efficiency and speed of both matrix factorization and Louvain algorithms. We explore the significance and efficacy of the Louvain algorithm in community detection tasks. In different domains, we have explored computational efficiency in terms of the suggested approach's time and RMSE value. The results show that better recommendations can be provided by using our proposed approach. The well-structured communities are also formed by using the Louvain community detection method, which helps give better recommendations. The results also show that the method applies to large and diverse datasets and generates meaningful recommendations for the users based on their experience. The primary benefit is emphasizing the

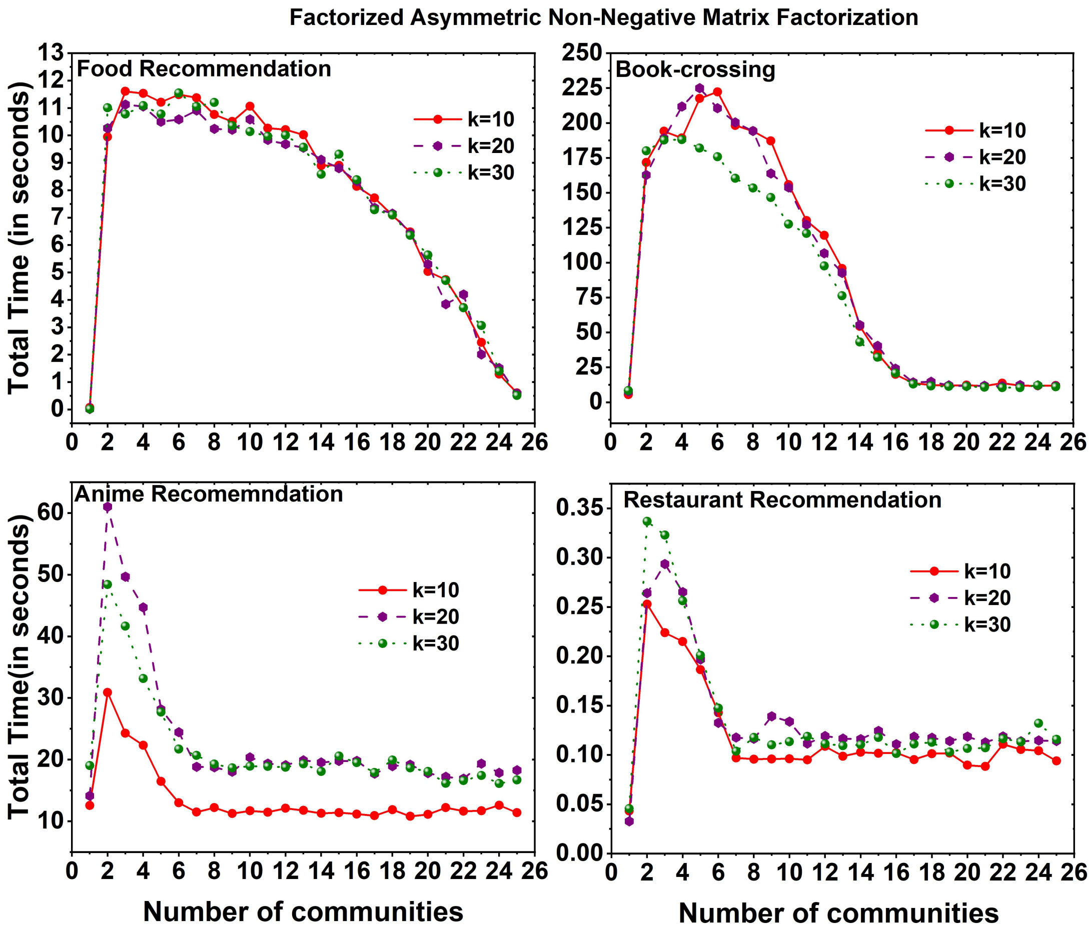



FIGURE 12. Comparing computational time for FANMF method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

algorithm's efficiency by giving interpretable results for large datasets. The combined matrix factorization and community detection approaches are used in real-world datasets across diverse domains which include nutritional networks, social networks, and recommender systems. Furthermore, this work can be extended to addressing real-world challenges and handling the data that is present in different domains. We seek to enhance data-driven insights across multiple domains and further develop high-performance computing by promoting a continuous dialogue between research discoveries and practical applications. This work can be extended by adding additional information based on the users. This framework initiates as the stepping stone for getting efficient user recommendations.

## REFERENCES

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-based systems*, vol. 46, pp. 109–132, 2013.
[2] G. Geetha, M. Safa, C. Fancy, and D. Saranya, "A hybrid approach using collaborative filtering and content based filtering for recommender system," in *Journal of Physics: Conference Series*, vol. 1000. IOP Publishing, 2018, p. 012101.
[3] R. Van Meteren and M. Van Someren, "Using content-based filtering for recommendation," in *Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop*, vol. 30. Barcelona, 2000, pp. 47–56.
[4] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The adaptive web: methods and strategies of web personalization*. Springer, 2007, pp. 291–324.
[5] D. Bokde, S. Girase, and D. Mukhopadhyay, "Matrix factorization model in collaborative filtering algorithms: A survey," *Procedia Computer Science*, vol. 49, pp. 136–146, 2015.
[6] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
[7] S. Marinetti, L. Finesso, and E. Marsilio, "Matrix factorization methods: Application to thermal ndt/e," *NDT & E International*, vol. 39, no. 8, pp. 611–616, 2006.
[8] B. Barney *et al.*, "Introduction to parallel computing," *Lawrence Livermore National Laboratory*, vol. 6, no. 13, p. 10, 2010.
[9] K. Asanovic, R. Bodik, J. Demmel, T. Keaveny, K. Keutzer, J. Kubiatowicz, N. Morgan, D. Patterson, K. Sen, J. Wawrzynek *et al.*, "A view of the parallel computing landscape," *Communications of the ACM*, vol. 52, no. 10, pp. 56–67, 2009.
[10] S. H. Strogatz, "Exploring complex networks," *nature*, vol. 410, no. 6825, pp. 268–276, 2001.
[11] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Physics reports*, vol. 659, pp. 1–44, 2016.
[12] B. M. Maggs, L. R. Matheson, and R. E. Tarjan, "Models of parallel computation: A survey and synthesis," in *Proceedings of the Twenty-Eighth Annual Hawaii International Conference on System Sciences*, vol. 2. IEEE, 1995, pp. 61–70.

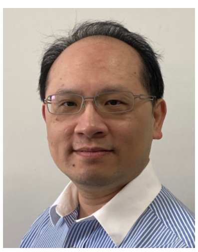
[13] S. Tokala, M. K. Enduri, and T. J. Lakshmi, "Unleashing the power of svd and louvain community detection for enhanced recommendations," in *2023 IEEE 15th International Conference on Computational Intelligence and Communication Networks (CICN)*. IEEE, 2023, pp. 807–811.
[14] X. Que, F. Checconi, F. Petrini, and J. A. Gunnels, "Scalable community detection with the louvain algorithm," in *2015 IEEE International Parallel and Distributed Processing Symposium*. IEEE, 2015, pp. 28–37.
[15] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, pp. 75–174, 2010.
[16] P. Chunaev, "Community detection in node-attributed social networks: a survey," *Computer Science Review*, vol. 37, p. 100286, 2020.
[17] X. Luo, Z. Liu, L. Jin, Y. Zhou, and M. Zhou, "Symmetric nonnegative matrix factorization-based community detection models and their convergence analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 1203–1215, 2021.
[18] J. Yang and J. Leskovec, "Overlapping community detection at scale: a nonnegative matrix factorization approach," in *Proceedings of the sixth ACM international conference on Web search and data mining*, 2013, pp. 587–596.
[19] F. Ye, C. Chen, and Z. Zheng, "Deep autoencoder-like nonnegative matrix factorization for community detection," in *Proceedings of the 27th ACM international conference on information and knowledge management*, 2018, pp. 1393–1402.
[20] M. A. Javed, M. S. Younis, S. Latif, J. Qadir, and A. Baig, "Community detection in networks: A multidisciplinary review," *Journal of Network and Computer Applications*, vol. 108, pp. 87–111, 2018.
[21] D. Huang, D. Mu, L. Yang, and X. Cai, "Codetect: Financial fraud detection with anomaly feature detection," *IEEE Access*, vol. 6, pp. 19 161–19 174, 2018.
[22] H. Tong and C.-Y. Lin, "Non-negative residual matrix factorization with application to graph anomaly detection," in *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM, 2011, pp. 143–153.
[23] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 135–142.
[24] V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joye, N. Taft, and D. Boneh, "Privacy-preserving matrix factorization," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 801–812.
[25] Y. Chen and X. Chen, "Research on knowledge graph application technology," in *Journal of Physics: Conference Series*, vol. 1187, no. 4. IOP Publishing, 2019, p. 042083.
[26] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He, "A survey on knowledge graph-based recommender systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3549–3568, 2020.
[27] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-commerce recommendation applications," *Data mining and knowledge discovery*, vol. 5, pp. 115–153, 2001.
[28] J. Hintz, "Matrix factorization for collaborative filtering recommender systems," 2015.
[29] E. Zangerle and C. Bauer, "Evaluating recommender systems: survey and framework," *ACM computing surveys*, vol. 55, no. 8, pp. 1–38, 2022.
[30] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in *Proceedings of the fifth ACM conference on Recommender systems*, 2011, pp. 301–304.
[31] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, pp. 267–273.
[32] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," in *Proceedings of the 2006 SIAM international conference on data mining*. SIAM, 2006, pp. 549–553.
[33] M. N. Schmidt, O. Winther, and L. K. Hansen, "Bayesian non-negative matrix factorization," in *Independent Component Analysis and Signal Separation: 8th International Conference, ICA 2009, Paraty, Brazil, March 15-18, 2009. Proceedings 8*. Springer, 2009, pp. 540–547.
[34] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *2008 Eighth IEEE international conference on data mining*. Ieee, 2008, pp. 263–272.
[35] S. Jiang, J. Li, and W. Zhou, "An application of svd++ method in collaborative filtering," in *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. IEEE, 2020, pp. 192–197.
[36] W. Shi, L. Wang, and J. Qin, "User embedding for rating prediction in svd++-based collaborative filtering," *Symmetry*, vol. 12, no. 1, p. 121, 2020.
[37] M. H. Aghdam and M. D. Zanjani, "A novel regularized asymmetric non-negative matrix factorization for text clustering," *Information Processing & Management*, vol. 58, no. 6, p. 102694, 2021.
[38] S. Tokala, M. K. Enduri, T. J. Lakshmi, and H. Sharma, "Community-based matrix factorization (cbmf) approach for enhancing quality of recommendations," *Entropy*, vol. 25, no. 9, p. 1360, 2023.
[39] Z. Zhang, P. Pu, D. Han, and M. Tang, "Self-adaptive louvain algorithm: Fast and stable community detection algorithm based on the principle of small probability event," *Physica A: Statistical Mechanics and Its Applications*, vol. 506, pp. 975–986, 2018.
[40] S. Ghosh, M. Halappanavar, A. Tumeo, A. Kalyanaraman, H. Lu, D. Chavarria-Miranda, A. Khan, and A. Gebremedhin, "Distributed louvain algorithm for graph community detection," in *2018 IEEE international parallel and distributed processing symposium (IPDPS)*. IEEE, 2018, pp. 885–895.
[41] M. Mohammadi, M. Fazlali, and M. Hosseinzadeh, "Accelerating louvain community detection algorithm on graphic processing unit," *The Journal of Supercomputing*, vol. 77, pp. 6056–6077, 2021.
[42] B. Hallinan and T. Striphas, "Recommended for you: The netflix prize and the production of algorithmic culture," *New media & society*, vol. 18, no. 1, pp. 117–137, 2016.
[43] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.
[44] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
[45] R. Kumar, B. Verma, and S. S. Rastogi, "Social popularity based svd++ recommender system," *International Journal of Computer Applications*, vol. 87, no. 14, 2014.
[46] S. Rendle, "Factorization machines," in *2010 IEEE International conference on data mining*. IEEE, 2010, pp. 995–1000.
[47] A. Tosyali, J. Kim, J. Choi, and M. K. Jeong, "Regularized asymmetric nonnegative matrix factorization for clustering in directed networks," *Pattern Recognition Letters*, vol. 125, pp. 750–757, 2019.
[48] M. Hosseinzadeh Aghdam and M. Daryaie Zanjani, "A novel regularized asymmetric non-negative matrix factorization for text clustering," 2021.
[49] V. Blondel, J.-L. Guillaume, and R. Lambiotte, "Fast unfolding of communities in large networks: 15 years later," *arXiv preprint arXiv:2311.06047*, 2023.
[50] M. Seifikar, S. Farzi, and M. Barati, "C-blondel: an efficient louvain-based dynamic community detection algorithm," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 2, pp. 308–318, 2020.
[51] M. Fazlali, E. Moradi, and H. T. Malazi, "Adaptive parallel louvain community detection on a multicore platform," *Microprocessors and Microsystems*, vol. 54, pp. 26–34, 2017.
[52] J. Zhang, J. Fei, X. Song, and J. Feng, "An improved louvain algorithm for community detection," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–14, 2021.
[53] C. D. Linhares, J. R. Ponciano, F. S. Pereira, L. E. Rocha, J. G. S. Paiva, and B. A. Travençolo, "Visual analysis for evaluation of community detection algorithms," *Multimedia Tools and Applications*, vol. 79, pp. 17 645–17 667, 2020.

SRILATHA TOKALA received B.Tech Degree in Computer Science and Engineering from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India in 2013, and M.Tech degree in Computer Science and Engineering from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India in 2015. She is currently pursuing a Ph.D. degree in Computer Science and Engineering from SRM University-AP, Amaravati, India. Her research interests include Recommender Systems, Algorithms, and Complex Networks. (e-mail: srilatha_tokala@srmap.edu.in)

MURALI KRISHNA ENDURI (Member IEEE) is an Assistant Professor at the Department of Computer Science and Engineering at SRM University-AP, Amaravati, India. He holds a M.Sc. in Mathematics from the Acharya Nagarjuna University, Andhra Pradesh, India in 2008. He obtained his M. Tech in Systems Analysis and Computer Applications from the National Institute of Technology Karnataka, India in 2011. He received his Ph. D in Computer Science and Engineering from the Indian Institute of Technology Gandhinagar, Gujarat, India in 2018. He has research experience/post Doc at Indian Institute of Technology Madras, Tamil Nādu, India in 2018. Murali Krishna Enduri's research interest includes Recommender Systems, Algorithms, Complexity Theory, and Complex Networks. Within these general areas, He has research experience in designing algorithms for graph isomorphism problems for restricted graph classes, measuring the impact of the diversity of an article, and predicting the spread of disease. (e-mail: muralikrishna.e@srmap.edu.in)

T. JAYA LAKSHMI (Member IEEE) is working as an Assistant Professor in the Department of Computer Science and Engineering, SRM University, Amaravathi, AP, India. She received her Ph.D. Degree for her work on "Link Prediction in Heterogeneous Social Networks" from the School of Computer and Information Sciences, University of Hyderabad, India in 2019. She is a reviewer of reputed international journals. She has an overall teaching experience of 22 years. Her research interests include graph mining, recommender systems, Natural Language Processing, and security analytics. (e-mail: jaya.phd.hcu@gmail.com)

ASHU ABDUL (Senior Member, IEEE) received the B.Tech. degree in computer science and engineering from Jawaharlal Nehru Technological University, Hyderabad, India, in 2009, and the M.Tech. degree in computer science and engineering from the International Institute of Information Technology, Bhubaneswar, India, in 2012. He completed the Ph.D. degree in computer science and information engineering from Chang Gung University, Guishan, Taoyuan, Taiwan, in 2019. He is currently a Post-Doctoral Researcher at Chang Gung Memorial Hospital, Taiwan. His main research interests include the design, analysis, and implementation of artificial intelligence algorithms and deep learning approaches. (e-mail: ashu.a507@gmail.com)



JENHUI CHEN (Senior Member, IEEE) received the B.S. and Ph.D. degrees in Computer Science and Information Engineering (CSIE) from Tamkang University, Taipei, Taiwan, in 1998 and 2003, respectively. Since 2003, he has been with the Department of CSIE, College of Engineering, Chang Gung University, Taiwan, where he is currently a full professor and chairperson. He is the section head of technology foresight, Artificial Intelligence (AI) Research Center, Chang Gung University. He is also a professor with the center for AI in medicine, Chang Gung Memorial Hospital, Taoyuan, Taiwan. His main research interests include the design, analysis, and implementation of human-like intelligence (HI), computer vision, image processing, deep learning, data science, and multimedia processing. He served as the technical program committee (TPC) member for IEEE Globecom, IEEE VTC, IEEE ICC, IEEE ICCC, IEEE ICCCN, IEEE 5G World Forum, and ACM CCIOT. He also served as a reviewer for many famous academic journals which are organized by ACM, Elsevier, IEEE, and Springer. He is currently a senior editor of Cogent Engineering. (e-mail: jhchen@mail.cgu.edu.tw)