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

IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 29 September 2024, accepted 30 October 2024, date of publication 5 November 2024, date of current version 14 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3491829

RESEARCH ARTICLE

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[1,2], (Member, IEEE), ASHU ABDUL[1,3], (Senior Member, IEEE), AND JENHUI CHEN[4,5], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, SRM University-AP, Mangalagiri 522240, India
[2]Department of Computing and Software Engineering, Sheffield Hallam University, S1 1WB Sheffield, U.K.
[3]Center for Artificial Intelligence in Medicine, Chang Gung Memorial Hospital, Guishan District, Taoyuan 33375, Taiwan
[4]Department of Computer Science and Information Engineering, Chang Gung Memorial Hospital, Guishan District, Taoyuan 33302, Taiwan
[5]Division of Breast Surgery and General Surgery, Department of Surgery, Chang Gung Memorial Hospital, Guishan District, Taoyuan 33375, Taiwan

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

This work was supported in part by the National Science and Technology Council, Taiwan, under Grant NSTC 113-2221-E-182-024; 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 matrix 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), mean square error (MSE), and mean absolute error (MAE). 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 five different diverse datasets to enhance the quality of the recommendation. To determine the method's efficiency, the evaluation metrics RMSE, MSE, and MAE are used, and the time required to evaluate the computation is also computed. It is observed in the results that almost 95% of our results are proven effective by getting lower RMSE, MSE, and MAE values. 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], [2]. These recommender systems are broadly categorized into content-based recommendations and collaborative filtering recommendations [3].

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

The user's product choices are recommended in the content-based recommendations based on their user profile [4], [5]. For instance, if a user has already appreciated action films, the content-based method will recommend action movies with comparable features [6]. 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 [7]. Matrix factorization is one of the 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 [8]. 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 [9], [10]. This decomposition helps in the problems of storage and computation for diverse datasets that are large in dimensions [11].

In network science and data analysis, complex datasets undergo several hurdles in achieving and handling computational efficiency and scalability issues [12], [13]. Complex network analysis has garnered substantial interest across diverse fields, including social sciences, biology, and computer science [14], [15]. In the concept of network analysis, there arises the fundamental concept of community detection, which is used to handle large and diverse datasets [16]. 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 [17]. To define the communities effectively, there are several community detection methods, and the most 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 [18], [19], [20].

There are billions of users in day-to-day life who need recommendations for products, social media, job vacancies, music, etc. Several matrix factorization techniques are used to provide recommendations for users. While constructing the user-item matrix, for these billions of users the time complexity will be increased, and not sure of suggesting accurate recommendations. Our main study focuses on improving the recommendation quality for users by suggesting accurate recommendations, with less time complexity. Behind this motivation, we have proposed a new approach that integrates the matrix factorization approaches with the community detection method, i.e., the Louvain approach, for better detection of the community structures and to provide accurate recommendations.

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 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. Some of the advantages and limitations are addressed in Section VI. In Section VII, 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 [21]. 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. 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 [22]. 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 [23]. This powerful synergy has demonstrated remarkable success in several notable areas.

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 [24]. The detection of gene modules functionality from the gene analysis is possible by integrating the matrix factorization and community detection in **bioinformatics** [25]. This interaction helps identify 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 [26], [27]. The community detection and matrix factorization approaches help to 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 [28]. 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** [29]. 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 [30]. 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.

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*.**

B. PROBLEM STATEMENT

The problem of recommender systems is defined by Schafer et al. [31] as follows:

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 recommender system's primary goal is to recommend a new product Y_b to a user X_a when the user has not yet seen a certain item.

Considering an example of a rating matrix in Fig. 1. The matrix is a rating matrix, and the values suggest the user's ratings of an item. Every row of the matrix represents a user, while each of the columns relates to numerous objects, such as movies. Every entry in the matrix indicates a user's rating of the film. The primary role of the recommendation system is to anticipate the absence of entries in the rating matrix.

There are several strategies for handling the problem in recommender systems, mostly using content-based and collaborative filtering approaches [32], [33]. One of the most used collaborative filtering strategies is the use of matrix factorization (MF), which boosts proposal quality while lowering time complexity. To improve the quality of ideas, we suggest representing the matrix of ratings as an intricate network, including community structures. We aim to concurrently incorporate community information into the matrix factorization technique. 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.

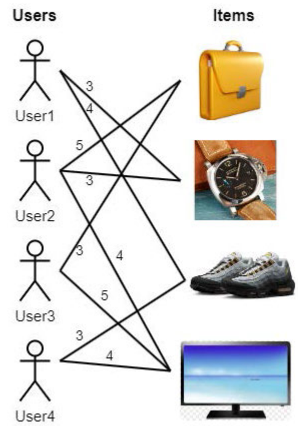

FIGURE 2. **Rating networks visualizing user-item dynamics in bipartite structures.**

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 [34]. Several matrix factorization algorithms are introduced to minimize the squared error [35]. In 2011, the authors introduced a new recommendation algorithm that enhances matrix factorization by considering contextual factors [36]. 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. A novel trust-based MF technique is developed, which utilizes social network data in the recommendation process by representing users as both trustees and trustees, according to the trust network's structural information [37]. The strategy attempts to address the lack of data and the issue of a cold start by including several information sources in the recommendations model, including ratings and trusted statements.

A unique version of NMF is developed 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 [38]. 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. Zhang et al. developed two versions of Non-negative Matrix Factorization in 2006 to achieve limited linear modeling using an incomplete rating matrix [39]. A Bayesian approach to non-negative matrix factorization (NMF) in 2009 is proposed, utilizing a normal likelihood and exponential priors [40]. An effective Gibbs sampler is

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

Author	Method	Keyfindings
Hintz *et al.* [34]	Matrix Factorization techniques	Several latent feature models for matrix factorization techniques that are used for enhancing the quality of recommendations.
Linas *et al.* [36]	Context-aware recommendation	Introduces extra parameters to account for how contextual factors interact with item ratings.
Navid *et al.* [37]	Trust-based matrix factorization	Mitigates the data sparsity and cold start problems from multiple sources and produces recommendations.
Wei Xu *et al.* [38]	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.* [39]	Non-negativity constrained linear model	Proposed limited linear modeling using an incomplete rating matrix.
Schmidst *et al.* [40]	Bayesian Non-Negative Matrix Factorization	A Bayesian approach to NMF is presented, based on the normal likelihood and exponential priors.
Xin *et al.* [41]	Symmetric Non-Negative Matrix Factorization	Loss function and the convergence models are studied which in turn gain a significant accuracy gain for all the community models.
Koren *et al.* [42]	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.* [43]	SVD++ application with time feature	Includes a special timing feature to adjust dynamically and evaluate recommendations
Wenchuan *et al.* [44]	UE-SVD++	Looked closely at the ratings users give and used that information to build a special matrix called the user embedding matrix.
Ali *et al.* [45]	Clustering SVD++	Combines similarity and confidence values and proposed a clustering SVD and SVD++ method that predicts the interests of the cold-start users.
Aghdam *et al.* [46]	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.* [47]	Community-Based Matrix Factorization approach	Integrates community detection approach with matrix factorization method and finds out the incomplete matrix.
Zhang *et al.* [48]	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.* [49]	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.* [50]	Adaptive CUDA Louvain method	By using shared memory in GPU, and with minimum threads overhead is minimized.
Jicun *et al.* [51]	Fast Louvain method	Improves iterative logic by shifting from cyclic iteration to dynamic iteration.
Whitney *et al.* [52]	Community Zones using Louvain method	Develops community zones that enhance the existing approaches and produce new zones that delineate the community

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 2021 non-negative matrix factorization based on a symmetric nature is developed that implements community-based models [41]. By using this model, the loss function and the convergence models are studied, which in turn gain a significant accuracy gain for all the community models.

In 2008, Hu et al. and his team introduced a better way to recommend things by combining different types of feedback [42]. 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, a new version of the SVD++ algorithm was introduced [43]. Noticed that recommendation systems often struggle when there is not much data available. Different versions of the SVD algorithm have tried to tackle this issue, but they did not see much improvement in the recommendation results. So, they came up with this new algorithm to address these 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 the authors in 2020 [44]. 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++. In 2021, the authors employ a clustering method that uses trust relationships and rating information to determine weights [45]. By combining similarity and confidence values, the study generates weight values, which are then used to deduce distance values. It adopts the partitioning around medoids clustering algorithm to categorize users according to these calculated distances. To forecast items for cold-start users, SVD and SVD++ techniques are combined. The FANMF method is designed to handle non-negative data that are unevenly distributed [46]. 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 consideration 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 [47]. The performance metric signifies that the recommendations are appropriate for the user to get a quality recommendation.

In 2018, the authors introduced an enhanced version of the Louvain algorithm [48]. 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 et al. in 2018 described the architecture of a Louvain method distributed memory implementation meant for parallel community detection [49]. 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, leveraging the power of GPU [50]. Zhang et al. in 2021 developed an enhanced Fast Louvain algorithm to boost the detection efficiency of large-scale networks [51]. This solution improves iterative logic by moving from cyclic to dynamic iteration, which speeds up convergence and separates the local tree framework inside the network. The entire network is repeatedly partitioned, and the tree structure is integrated into the split results, which are then refined to reduce computing costs. As a result, it achieves superior community aggregation and enhances the effectiveness of community detection. In 2022, the community zones using the Louvain community detection method are developed, which improves the existing metrics [52]. The matrix factorization algorithms and the community detection method used in our tests are briefly addressed in the section that follows. All the literature review is shown in a tabular format in Table. 1.

Khaledian and Mardukhi in the year 2021 proposes a method named CFMT, that integrates the matrix factorization method with the trust information to improve the recommendation accuracy for the cold start users and to handle sparse data [37]. This method handles missing data using the SVD technique for approximate ratings. Then, regularization is added to the feature matrices with the Frobenius norm to improve the accuracy of the recommendation. CFMT also introduces a trust regularization term to align the user preferences with those of their friends and computes the similarity using the cosine similarity metric. Hence, merging rating data and trust information linearly addresses the cold start problem and sparse data. The authors' study focuses on improving the accuracy of community detection using the SNMF technique and the non-negative multiplicative update (NMU) schemes [41]. Four different novel models are proposed by adjusting the NMU scaling factor through linear and non-linear strategies. These four models outperform the traditional methods in detecting efficient communities. Appropriate values for scaling factors of α and β are chosen in detecting the efficient structure of communities. The method proposed by the authors enhances the recommendation accuracy of the cold start users by integrating the SVD++ technique with the clustering method [45]. Here the users are split into two groups based on their ratings. The similarity of this rating relation is calculated by a binary user-user matrix. The outliers are reduced by computing the confidence values between users and are integrated with the similarity scores to calculate the weights. Then the PAM algorithm clusters the users based on these weights. The error in the clusters is calculated by using different metrics RMSE and MAE.

The fast Louvain method proposed by the authors improves the efficiency and quality of the partitioning in large networks [51]. Two different key optimizations are introduced that handle the efficiency of communities in large networks. The dynamic iterative optimization reduces the redundant computations by skipping the modularity gain between the stable communities. The splitting of local tree structures handles the faster processing of nodes by splitting the tree before the main iteration. These improvements allow the fast Louvain algorithm to reduce its computational complexity and improve modularity. The study of the authors introduces new US community zones using the Louvain algorithm instead of using clustering techniques [52]. This work addresses two main key limitations in the existing community zones. Firstly, it improves the community flow measurements between countries. And second, it eliminates the arbitrary cutoff parameter using modularity.

III. METHODOLOGY

Matrix factorization is a prevalent technique in recommender systems, however, its effectiveness is notably challenged by the presence of sparse datasets. These recommender systems often exhibit high levels of sparsity, characterized by a user-item interaction matrix that contains a substantial proportion of missing or zero values. This sparsity complicates the application of matrix factorization, as it restricts the model's capacity to learn meaningful latent factors and produce accurate predictions.

In this study, we propose a solution to this challenge by segmenting the user-item matrix into dense communities and confining the matrix factorization process within these communities. Communities are generally formed by nodes sharing similar characteristics, and it is infrequent for nodes within a single community to maintain significant connections with nodes from other communities. By limiting recommendations to these community boundaries, recommender systems can substantially reduce the incidence of false positives, thereby enhancing the overall quality and relevance of the recommendations provided. Furthermore, community detection improves the efficiency of the recommendation process by narrowing the search space. Instead of evaluating the entire user base or item pool, the recommendation process can be restricted to users or items within specific communities, leading to a significant reduction in computational complexity and enhanced scalability.

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

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 [53]. 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 [54], [55]. 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 as shown in (1), 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 in (2).

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

The deviation between the original and the predicted ratings is given by as shown in (3).

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

A regularization term β of value 0.02 is added to the minimized squared error to avoid overfitting as in (4) [56].

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

Using a constant β, the impact of the overfitting is controlled. $\| . \|$ is the Frobenius norm. Stochastic gradient descent is used to calculate the prediction error for each rating in the data as shown below in (5).

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

The entries of the predicted rating matrices as shown in (6) are updated to minimize the squared error by adding the learning rate α value of 0.005 to the latent features [56].

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

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 is calculated as RMSE can be obtained as shown in (7).

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

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

The averaged squared difference between the original and the predicted rating matrices is calculated as MSE and can be obtained as shown in (8).

$$\text{MSE} = \frac{1}{T} \sum (r_{mn} - \widetilde{r}_{mn})^2 \tag{8}$$

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

The average of the absolute difference between the original and the predicted rating matrices is calculated as MAE, which can be obtained by as shown in (9).

$$\text{MAE} = \frac{1}{T} \sum (|r_{mn} - \widetilde{r}_{mn}|) \tag{9}$$

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 is performed on three different latent features likely $k = 10, 20, 30$, with a regularization parameter β of 0.02, and with a learning rate α of 0.005. 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 [57]. Paatero and Tapper defined positive matrix factorization, which helped to establish NMF [58]. Following the seminal work by Lee and Seng, NMF rapidly gained widespread recognition and popularity in the field [59], [60]. 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 [59] 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 average squared difference between the original rating matrix R and the predicted rating matrix $\widetilde{R}$, as given in (8). The average of the absolute difference between the original rating matrix R and the predicted rating matrix $\widetilde{R}$, as given in (9).

The non-negative matrix factorization approach is iterated for 100 steps, with different latent features for $k = 10, 20, 30$ and using the multiplicative update optimization algorithm. 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 [42]. In updating the latent feature matrices, implicit feedback is added to the user's latent feature matrix P [61], [62]. 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 (10) and is defined as the predicted rating matrix.

$$\widetilde{R} = [(P + UI).Q^T]. \tag{10}$$

Then, the difference between the original and the predicted rating matrices is calculated as RMSE value as shown in (7). The average squared difference between the original rating matrix R and the predicted rating matrix $\widetilde{R}$, as given in (8). The average of the absolute difference between the original rating matrix R and the predicted rating matrix $\widetilde{R}$, as given in (9). The SVD++ method is performed on three latent features of $k = 10, 20, 30$, and with 2 k-folds. 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 in 2019 [63]. 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 [46]. 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 using the multiplicative update values [59]. The deviation between the original rating matrix R and the predicted rating matrix $\widetilde{R}$ is calculated as RMSE as illustrated in (7). The average squared difference between the original rating matrix R and the predicted rating matrix $\widetilde{R}$, as given in (8). The average of the absolute difference between the original rating matrix R and the predicted rating matrix $\widetilde{R}$, as given in (9). The FANMF method is performed on $k = 10, 20, 30$ latent features, with the nndsvdar initialization method, by using the multiplicative update optimization algorithm. 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. [64]. 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 [65]. Louvain community detection approach is one of the non-overlapping community detection where the community structures that are identified are non-overlapping, where each node belongs to only one community. Several metrics 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 [66]. 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 (11).

$$\text{modularity}(C) = \Sigma_c \left[\frac{E_c}{E} - \left(\frac{d_c}{2E} \right)^2 \right] \tag{11}$$

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 [67]. The Louvain method is very important in networks as it can process fast and give accurate and well-structured communities [68]. 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 [69]. The computing capabilities play a crucial role in the useful insights of real-world networks.

Since community detection is a crucial aspect of our approach, we assess the modularity score of four datasets namely food recommendation, book-crossing, anime recommendation, and restaurant recommendation using the Girvan Neuman, Louvain, Leiden, and Label Propagation algorithms across different communities. As all the community detection algorithms involve randomness, we have iterated all the algorithms for 25 communities. Fig. 3 presents the modularity for four distinct datasets with 25 communities. It is evident from all datasets that the modularity score increases as the number of communities grows, stabilizing after reaching a certain point. In the food recommendation dataset it is observed that, for all different community detection algorithms, as the communities increase, there is an increase in the modularity score. The highest modularity score is given by the Louvain community detection approach and the lowest modularity is seen by the Leiden and Girvan Neuman algorithms for different communities. In the book-crossing dataset, it is observed that the highest modularity is given by the Louvain algorithm, and the lowest modularity is given by the Leiden and Label Propagation algorithms. For the anime recommendation dataset, the highest modularity is seen by the Louvain algorithm, and the least modularity is observed by the Leiden algorithm. For the restaurant recommendation dataset, it is observed that the Louvain algorithm gives the highest modularity, whereas the Label Propagation and the Leiden algorithms have the least modularity. From all these observations, we analyze Louvain's superiority in terms of modularity score compared to all the different community detection approaches. That's why we have chosen the Louvain community detection algorithm for its superior performance in terms of modularity score.

Table 2 shows the time required to calculate the modularity score for the food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets for four different community detection algorithms namely Girvan-Neuman, Louvain, Leiden, and Label Propagation algorithms. It is observed from the table that the label propagation algorithm takes less time to compute and gives a lower modularity score which is not preferable. The Girvan-Neuman algorithm takes the highest time to compute and gives the lower modularity score which is also not preferable. The Leiden algorithm gives the lowest modularity in less time which is also not preferable. The Louvain gives the best modularity score in less time, where the appropriate community structures are detected for providing accurate recommendations for the users. That's why we have chosen the Louvain community detection algorithm for its superior performance in terms of time.

TABLE 2. Time efficiency (in seconds) of community modularity score calculation for four different datasets, namely food recommendation, book-crossing, anime recommendation, and restaurant recommendation for the different community detection algorithms.

Dataset	Girvan Neuman (sec)	Louvain (sec)	Leiden (sec)	Label Propagation (sec)
Food Recommendation	21.09	138.53	1.13	0.569
Book-crossing	145445.76	44919.07	34.30	39.1
Anime Recommendation	48318.95	18658.23	102.76	40.17
Restaurant Recommendation	28.83	32.64	0.76	0.344

TABLE 3. Time complexity analysis of different community detection algorithms where n is the number of nodes and m is the number of edges of the input graph.

Dataset	Time Complexity
Girvan-Neuman	$\mathcal{O}(nm^2)$
Louvain	$\mathcal{O}(n \log n)$
Leiden	$\mathcal{O}(n+m)$
Label Propagation	$\mathcal{O}(m)$

Next, we consider the time complexity analysis of algorithms. Table 3 shows the time complexity of different community detection algorithms namely Girvan-Neuman [70], Louvain [66], Leiden [71], and Label Propagation [70] algorithms. Our analysis shows that in terms of modularity score and time complexity, the Louvain method aims to maximize modularity, a metric that evaluates the robustness of network community divisions. It accomplishes this by iteratively combining nodes into communities and subsequently merging these communities into larger ones, thereby effectively optimizing the modularity score and ensuring the formation of highly modular and clearly defined community structures. The Louvain method has a time complexity of $\mathcal{O}(n \log n)$ where n is the number of nodes. This makes it especially effective for large networks, as it can process a vast number of nodes and edges without a significant computational load. By iteratively refining communities through a series of local optimizations, the Louvain method ensures that each step remains computationally feasible, resulting in an overall efficient algorithm that rapidly converges to a high-quality solution.

IV. PROPOSED METHOD

A brief explanation of our proposed approach is detailed in this section. 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

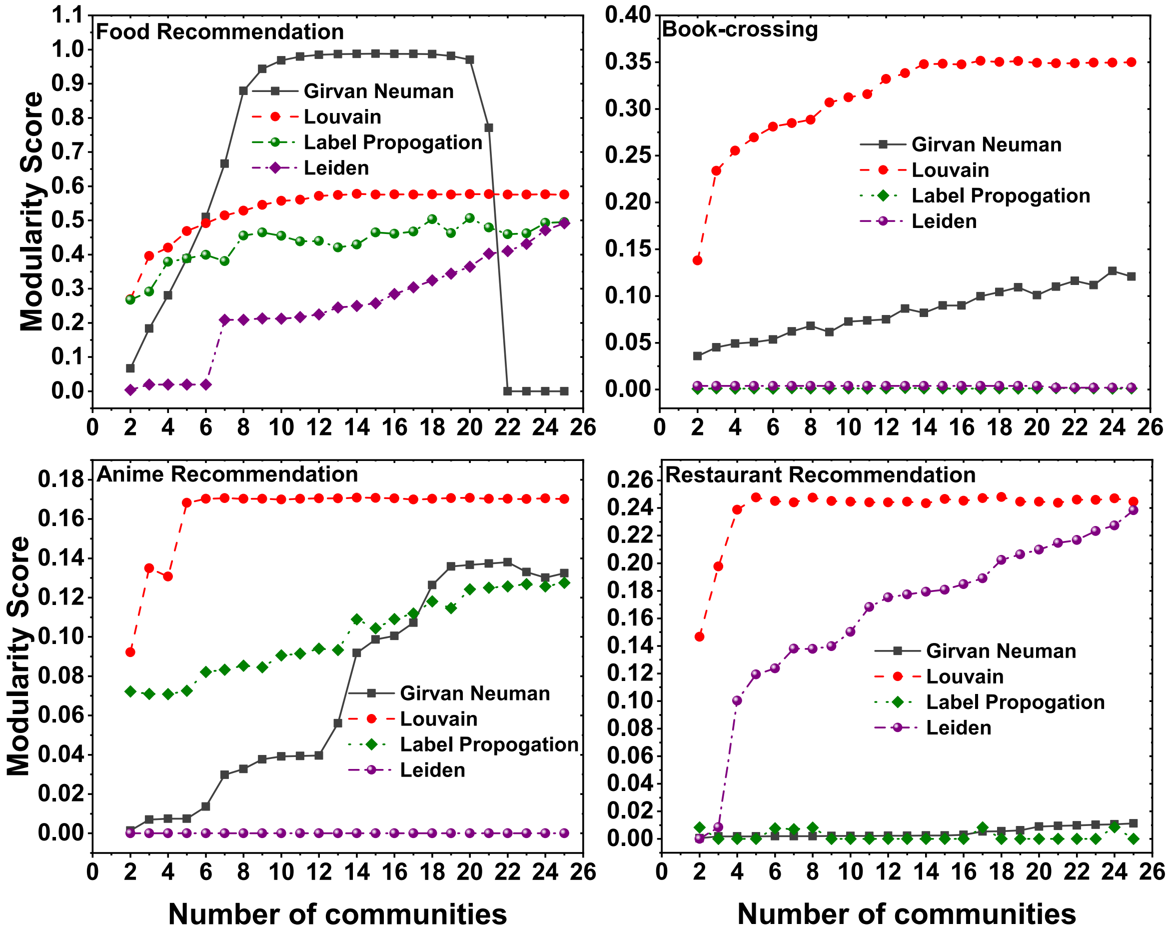

FIGURE 3. Evaluating modularity performance for four different datasets, namely food recommendation, book-crossing, anime recommendation, and restaurant recommendation for the different community detection algorithms.

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 in the proposed approach is shown in Fig. 4.

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 $BP = \{BP_1, BP_2, \ldots, BP_c\}$.
- 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., $RM_1, RM_2, \ldots, RM_c$.
- Step 5: In parallel, apply the matrix factorization methods to each rating matrix obtained in the previous step. As a result, predicted rating matrices $\widetilde{RM}_1, \widetilde{RM}_2, \ldots, \widetilde{RM}_c$ 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 ($\widetilde{RM}$).

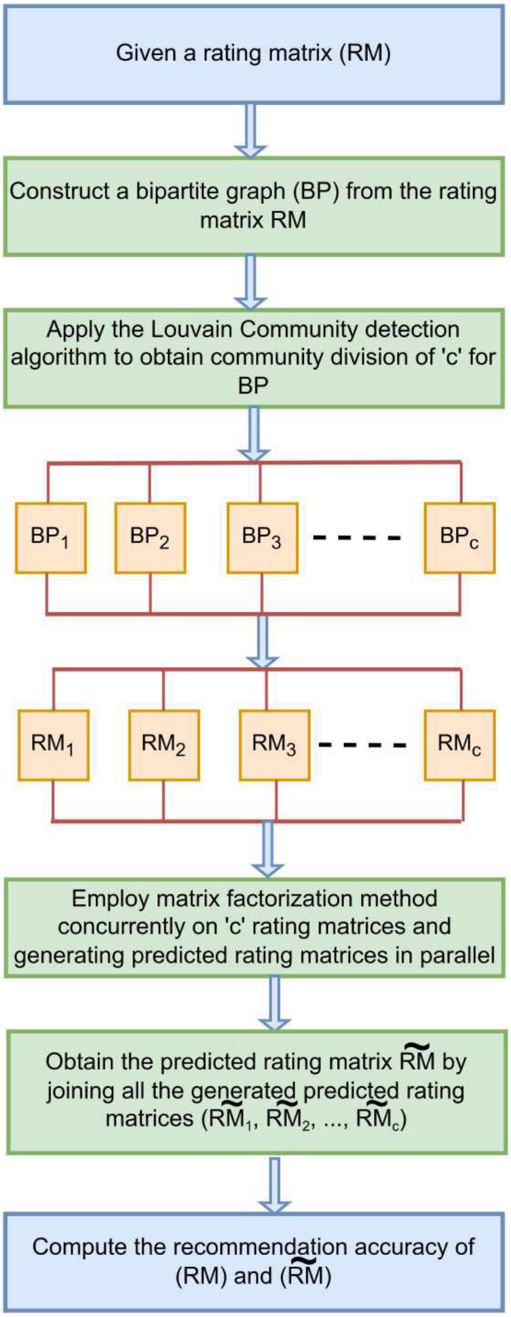

FIGURE 4. Workflow of matrix factorization approaches and Louvain community detection integration.

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

In our analysis, as there are c community structures, we will get 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.

In Fig. 5 we showcased a small toy example to clearly state the importance of the proposed approach. Matrix factorization is one of the emerging techniques used to provide user recommendations. Billions of users need recommendations in day-to-day life in E-commerce, social media, and job recommendations in real-time scenarios. By using matrix factorization, all these applications can be constructed in the form of a user-item matrix $m \times n$, where there are m users and n items. The user-item matrix that is formed is huge and lacks in providing accurate recommendations. The main motivation behind our study is to improve the recommendation quality and reduce the time complexity. Our proposed approach, as shown in Fig. 4 ensures improving the quality of recommendations and reducing the time complexity, which is very helpful for real-time scenarios. One of the applications our model can use is E-commerce, where the users are the people and the items are the products; we have the rating that is given by each user on the product. The product recommendations are given to the users. In social media recommendations for suggesting new friends and groups to join, where the users are the people interacting on the platform, the items are the posts or videos, ratings measure how much a user likes the item. The recommendations will be the suggestions given to the users based on their past behavior and preferences. For job recommendations, it will be more helpful for people to get suggestions about the vacancies regarding their relevant job postings. Here the users are the job seekers, items are the job postings, and the ratings are the feedback that the job seekers give to improve the performance of suggestions. Our proposed approach is applicable in other practical applications like E-learning platforms, retail stores, and streaming services. This information can be viewed in the form of a rating matrix RM and suggests recommendations for the users. Initially, we have taken an original rating matrix (RM) with 6 users i.e., X_1, X_2, X_3, X_4, X_5, and X_6 along with 5 items Y_1, Y_2, Y_3, Y_4, and Y_5. There are a total of 8 ratings in the rating matrix. The rating matrix with six users and five items is constructed as a bipartite graph (BP) representing nodes as users and items and the edges as ratings. Hence, the total number of nodes in the bipartite graph is 11, with 8 edges between the nodes. Next, the bipartite graph is given as an input for the Louvain community detection method of size 2. So, the bipartite graph with size 2 is divided for the Louvain community detection method i.e., BP_1 and BP_2.

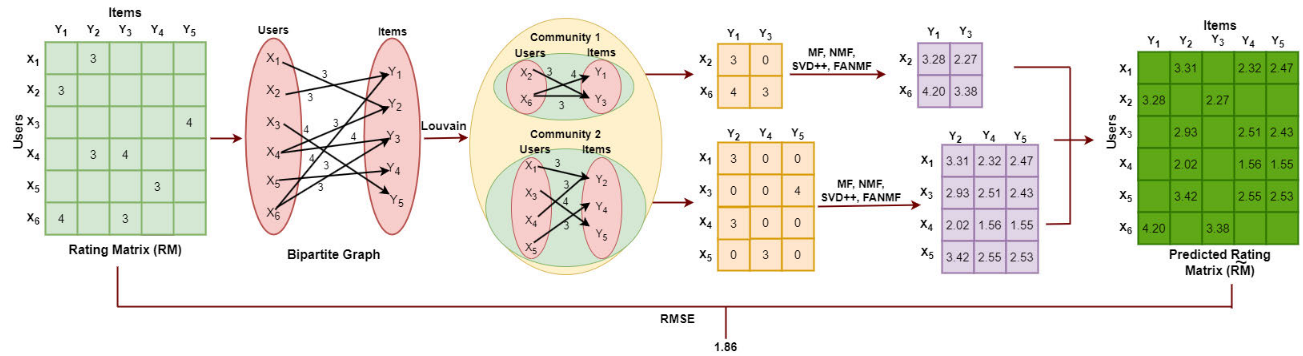

FIGURE 5. A toy example with 6 users and 5 items as a rating matrix showcasing the procedure of the proposed approach.

In the community 1, there are two users i.e., X_2, and X_6, with two items i.e., Y_1 and Y_3. Similarly for the community 2, there are four users i.e., X_1, X_3, X_4, and X_5, with three items i.e., Y_2, Y_4, and Y_5. This indicated that for the given input rating matrix, X_2 and X_6 are similar users and are grouped as one community, and the remaining users, i.e., X_1, X_3, X_4, and X_5 are grouped into the second community.

In the next step, we are obtaining small rating matrices from the bipartite graphs that are evaluated by the communities. As there are two communities, we are getting two different small rating matrices i.e., RM_1 and RM_2, of sizes 2×2 and 4×3. Then apply any kind of matrix factorization approach in parallel to these small rating matrices to obtain the predicted rating matrices i.e., $\widetilde{RM}_1$, $\widetilde{RM}_2$. As a final step, we combine all the predicted rating matrices to obtain a final predicted rating matrix ($\widetilde{RM}$). We can see in the figure that after combining all the predicted rating matrices, we can observe the new recommendations are predicted for the users. Now we have a total of 16 ratings in the predicted rating matrix. After obtaining the final predicted rating matrix ($\widetilde{RM}$), we calculated the recommendation accuracy by using the RMSE evaluation metric to measure the difference between the original rating matrix (RM) and the final predicted rating matrix ($\widetilde{RM}$). A detailed experimental setup of our proposed approach is shown in the next section.

V. EXPERIMENTAL RESULTS

For implementing our proposed method, we have taken five different datasets namely food recommendation, book-crossing, anime recommendation, restaurant recommendation, and movielens-1M downloaded from Kaggle [72]. The dataset statistics for the five datasets are produced in Table 4. These datasets are applied to our proposed method to determine the RMSE, MSE, and MAE values and the time to evaluate the algorithm. We have evaluated the performance of the recommendation by using the RMSE, MSE, and MAE measures as it will penalize large errors.

All 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 integrate them with the matrix factorization method to fill in the missing ratings in the matrix. All the visualization plots are drawn using Origin Pro software. When the size of the dataset increases, there will be more CPU memory and space utilization. If the food recommendation dataset is processed, as it contains only 508 ratings the memory utilization will be 50MB RAM, when the movie lens-1M dataset is processed 5GB RAM will be utilized. When dealing with large datasets more computational power and time is required which can be costly and resource intensive.

TABLE 4. Dataset Statistics for five different datasets, namely Food Recommendation, Book-crossing, Anime Recommendation, Restaurant Recommendation, and MovieLens-1M.

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
MovieLens-1M	6040	3900	1000209	1-5	0.958

Fig. 6 displays the rating distribution of the food recommendation, book-crossing, anime recommendation, restaurant recommendation, and movielens-1M 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. In the movielens-1M

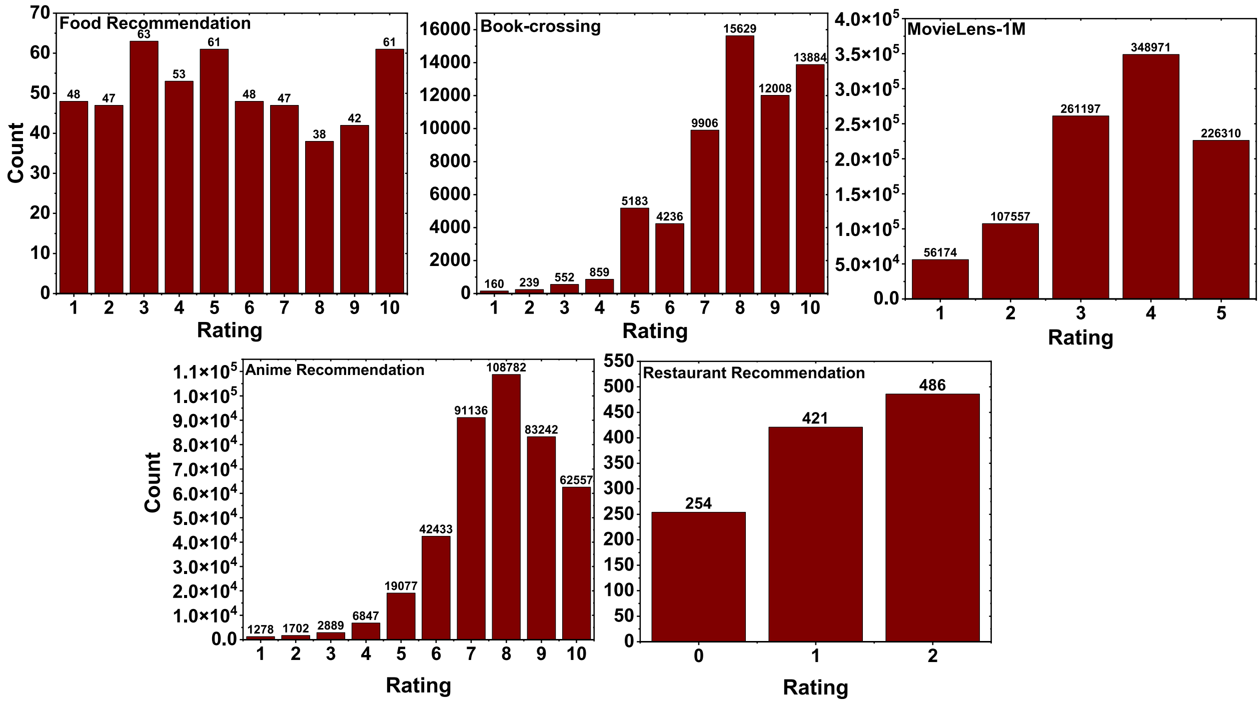

FIGURE 6. Rating distribution plots for food recommendation, book-crossing, anime recommendation, restaurant recommendation, and movielens-1M datasets.

dataset, we observe that the highest count of 348971 is for a 4 rating, and the lowest count of 56174 is for a 1 rating.

A. DISCUSSIONS ON RMSE RESULTS

Fig. 7 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 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 value of the RMSE drops as theÂ latent features in communities increase. We can see that the RMSE value decreases with the number of communities as compared to the score 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. By using the parallel approach with the basic MF and the Louvain community detection method, we can say that 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 provided for the users will be accurate thus improving the recommendation accuracy.

Fig. 8 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

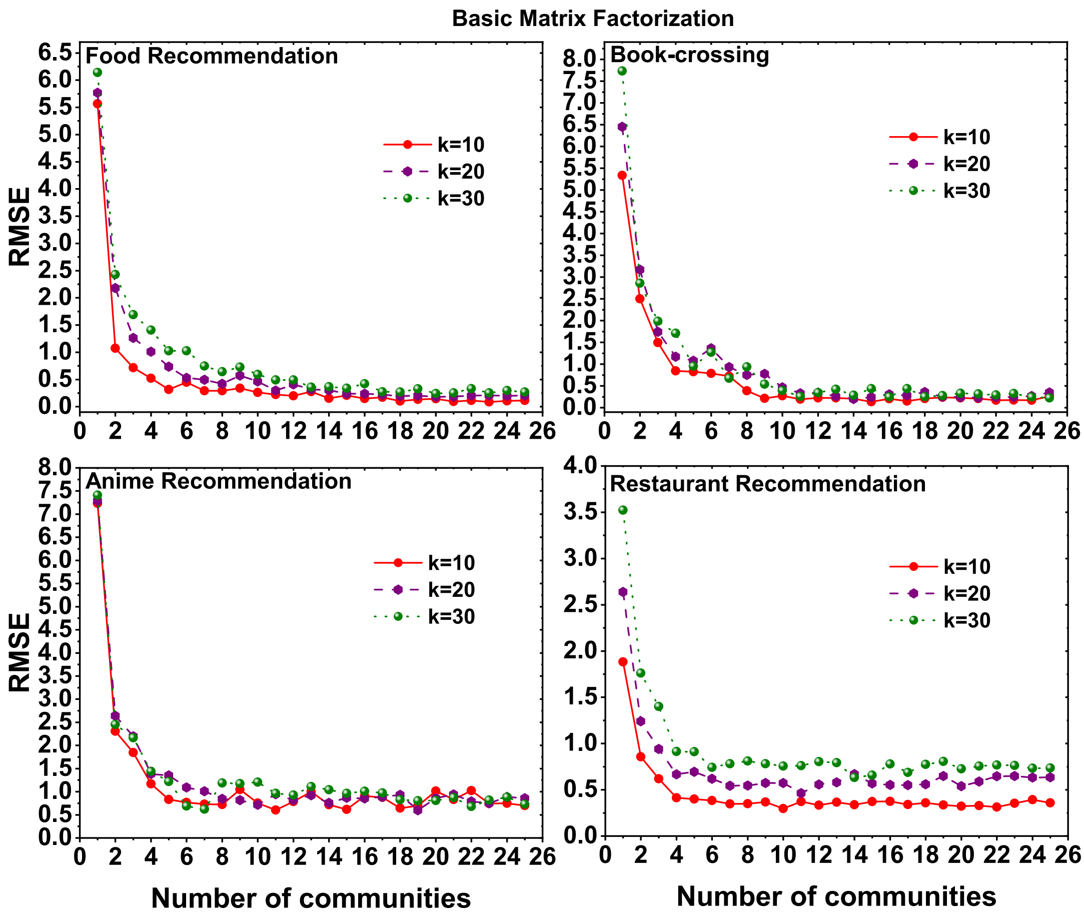

FIGURE 7. Examining the RMSE metrics for the basic matrix factorization method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

detection method is applied. 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 that is for the network. For the book-crossing dataset, we can observe that as the number of latent features increases, 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 before 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, the RMSE value is high in the restaurant recommendation dataset 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 which leads to the improvement of recommendation accuracy.

Fig. 9 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. 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 SVD++ and the Louvain community detection method, there is a better RMSE value

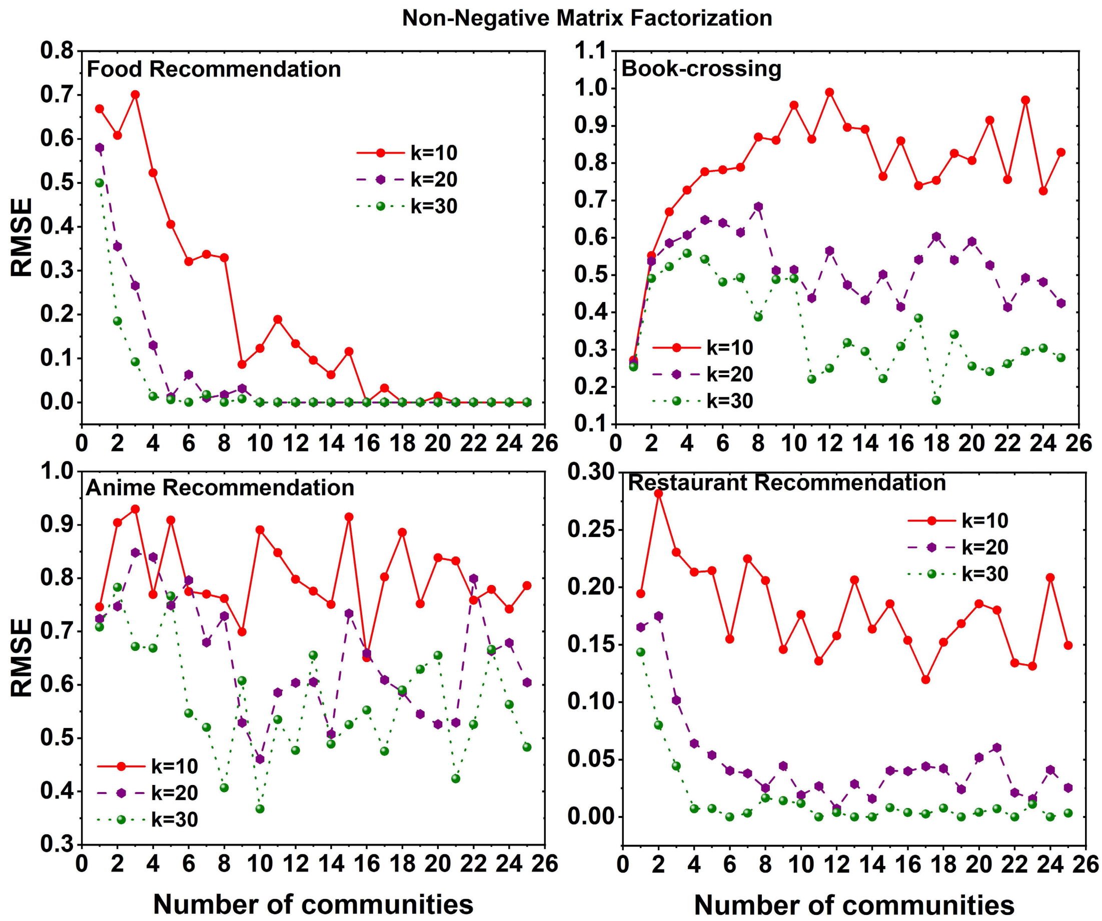

FIGURE 8. Examining the RMSE metrics for the NMF method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

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 provided for the users will be accurate thus improving the recommendation accuracy.

Fig. 10 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. The RMSE value drops as the assortment of latent features grows in proportion to the growing number of communities. Similarly, with the restaurant recommendation dataset, the RMSE value varies among communities, as does the amount of hidden characteristics. When just the FANMF approach is used, the RMSE is shown to be high. 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. Therefore, the recommendations for the users will be accurate using this approach which leads to the improvement of recommendation accuracy.

Fig. 11 shows the RMSE value on MovieLens-1M dataset for 25 communities and different latent features for $k = 10$, 20, and 30 for the basic MF, NMF, SVD++, and FANMF methods. For the basic MF method, it is observed that without using community detection at $c = 1$, the RMSE

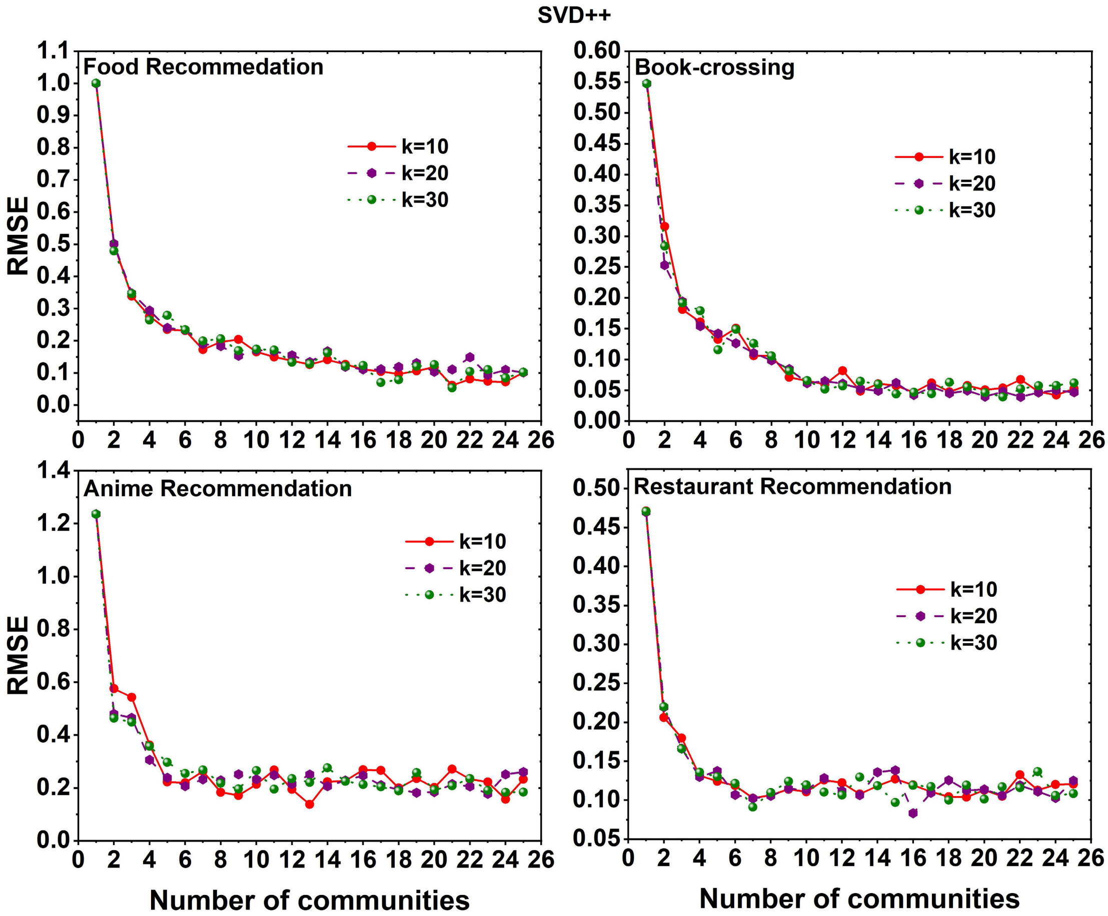

FIGURE 9. Examining the RMSE metrics for the SVD++ method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

value is high. While applying community detection for the basic MF 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 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 NMF method, we can observe that as the number of latent features increases, there are several ups and falls in RMSE value. Initially, there is a less RMSE value seen where there is no community division. When only the NMF approach is employed, the RMSE value decreases in comparison to employing the community detection method, and the RMSE value increases as the number of latent features and communities rises. We can see that the RMSE value increase with the number of communities compared to the value obtained before the community split. It is observed in the figure that, 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. 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 SVD++ and the Louvain community detection method, there is a better RMSE value for different communities. For the FANMF method, 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. Therefore, the recommendations for the users will be accurate using this approach which leads to the improvement of recommendation accuracy. By using the MovieLens-1M dataset, we demonstrated that our proposed technique is effective in handling large

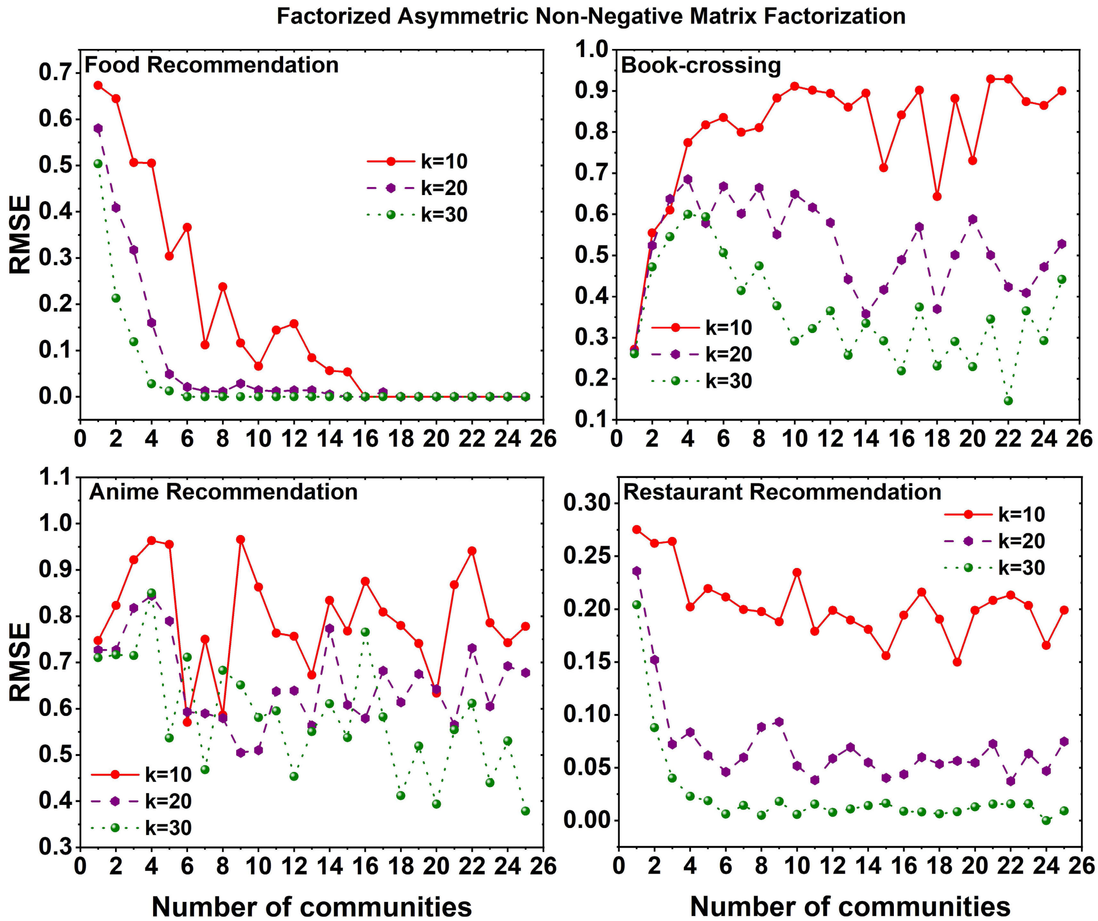

FIGURE 10. Examining the RMSE metrics for the FANMF method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

datasets in terms of RMSE for different matrix factorization techniques integrating with the Louvain community detection method.

Table 5 provides the comparison of the results of the RMSE values for five different datasets for four different MF 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 communities 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. For the movielens-1M dataset, we observe a better score when divided into 9 communities for the basic MF method, 21 communities for the NMF method, and 25 communities for the SVD++ and 7 communities for the FANMF method. From the five 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, it outperforms the non-utilization of the community approach with the MF method, which improves the recommendation accuracy.

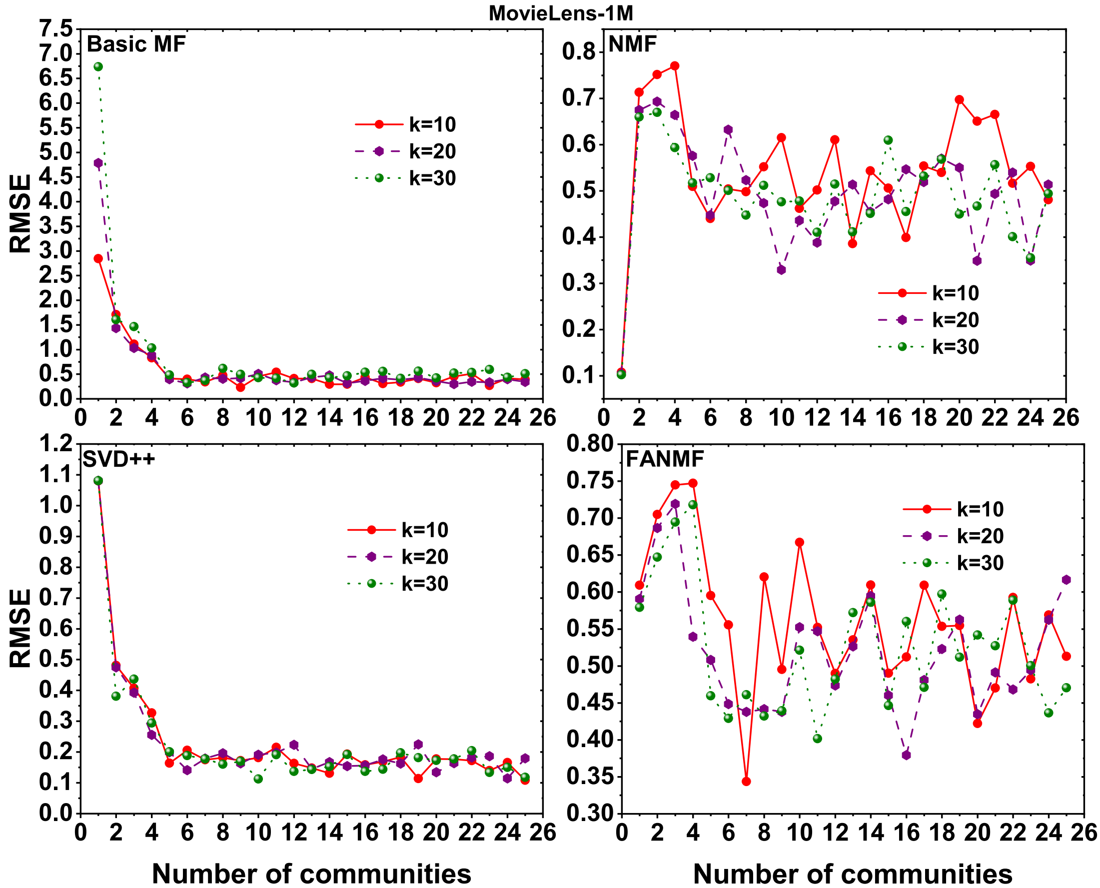

FIGURE 11. Examining the RMSE metric for the Basic MF, NMF, SVD++, and FANMF methods across different latent features and communities for movielens-1M dataset.

TABLE 5. Comparison of RMSE values for five 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)
MovieLens-1M	6.73	0.1	1.08	0.6	0.23 (9)	0.34 (21)	0.1 (25)	0.34 (7)

B. DISCUSSIONS ON COMPUTATIONAL TIME

Fig. 12 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 time complexity is reduced with 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. 13 shows the total time taken to evaluate the Non-Negative Matrix Factorization method along with the community detection method in the assessment of seconds.

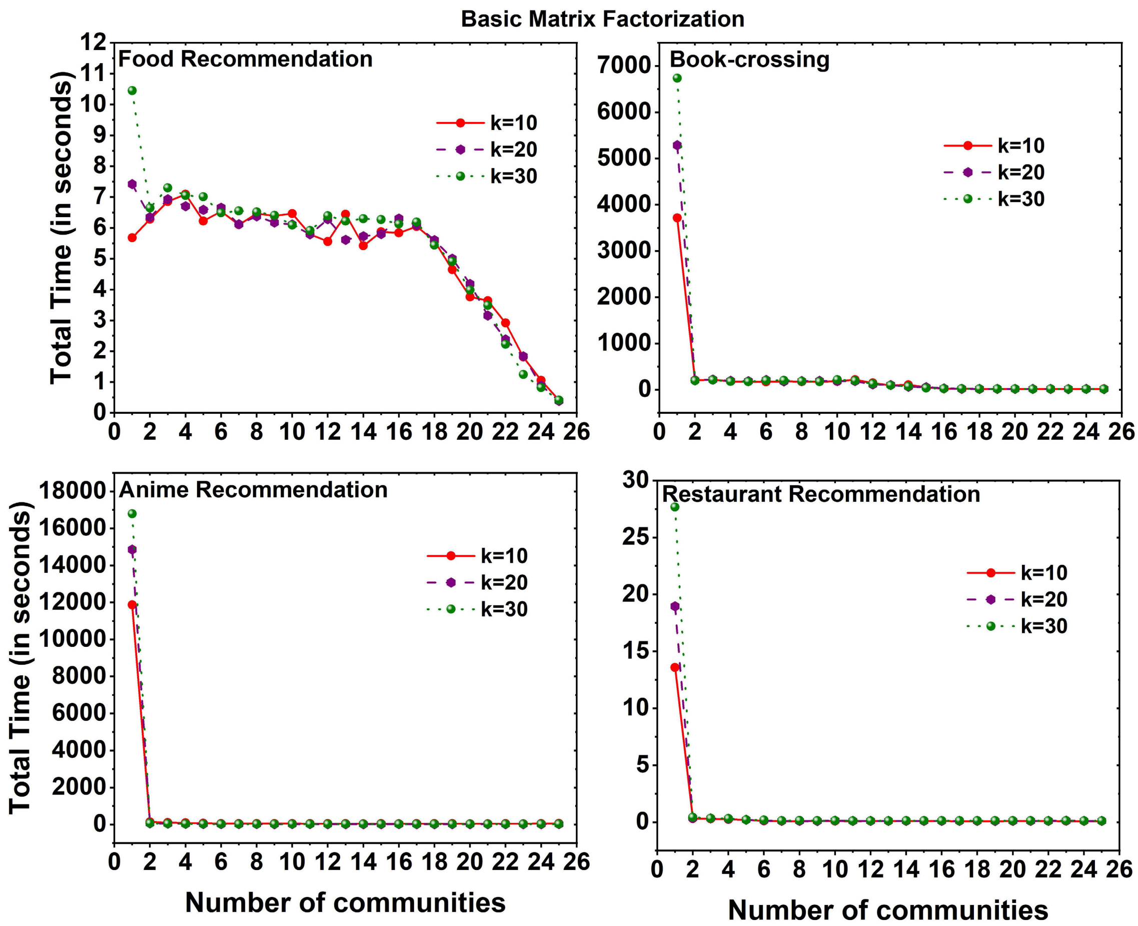

FIGURE 12. Comparing computational time for basic matrix factorization method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

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 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 increase. 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, where the time complexity of the method is reduced.

Fig. 14 shows the time necessary to assess the SVD++ technique and the community detection strategy in seconds. The total time is the sum of the time spent on community division and calculating the RMSE value. The statistic shows that the time spent in the community seeking food recommendations is less valuable. The combination of the SVD++ approach with the Louvain community detection method takes more time at community 2. The RMSE value decreases as the number of communities rises, and it achieves a minimum after a given number of communities. We can also see that as the number of latent characteristics rise, the RMSE value declined. In the book-crossing dataset, the RMSE value for community 1 is lower than community 2. After a specific number of iterations, the RMSE value declines relative to the value at community 1 and remains constant. For all the distinct latent characteristics that have been iterated, we see that the higher the latent feature value, the less time it takes. The figure from the anime recommendation dataset shows that greater time is spent on the *c* value at community 1. When the SVD++ approach is used with the

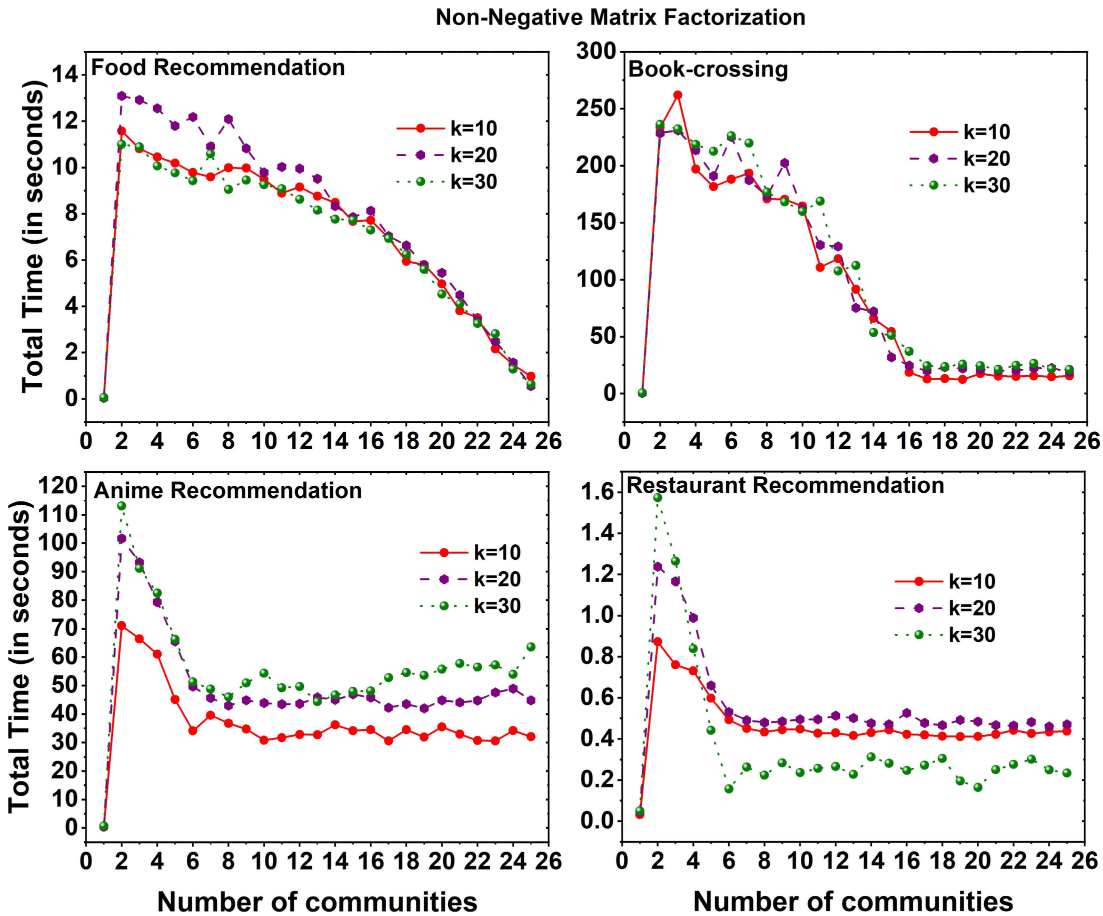

FIGURE 13. Comparing computational time for NMF method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

Louvain method, the time required decreases significantly as a number of communities grows. We get superior outcomes for time while the latent feature value is small. For the restaurant recommendation dataset, it is noted that when just the SVD++ technique is performed for the c value at community 1, the time consumed is shorter than at c value at community 2. As the number of communities expanded, we saw that the time required decreased, and at a certain number of communities, the time taken varied less and stayed consistent. The time complexity of the algorithm is also reduced as the evaluation takes only a fraction of second.

Fig. 15 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 much 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 reduced than by using only the matrix factorization approach.

Fig. 16 depicts the entire time required to evaluate the basic MF, NMF, SVD++, and FANMF techniques and the community detection approach in seconds for the MovieLens-1M dataset. 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 the basic MF method, 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

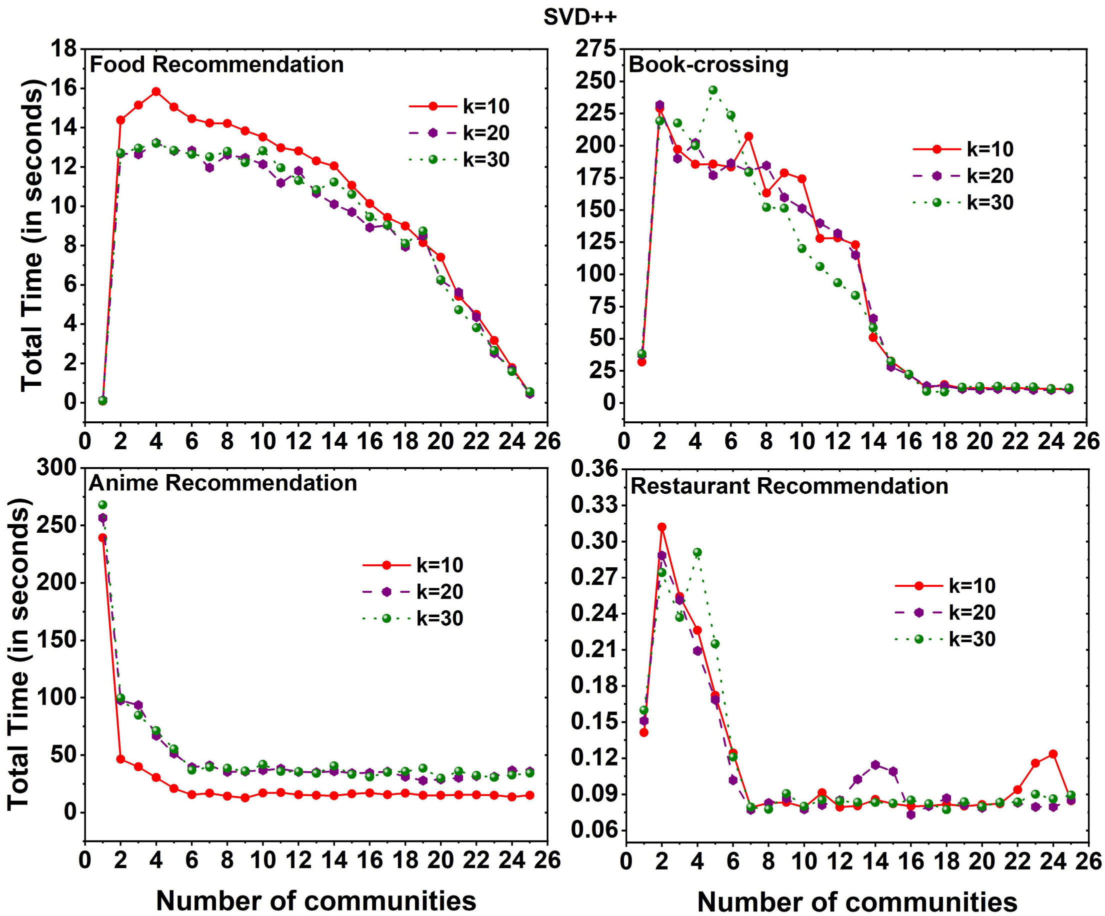

FIGURE 14. Comparing computational time for SVD++ method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

TABLE 6. Comparison of time (in seconds) for five different datasets on different MF methods by integrating Louvain and MF approaches Vs by not integrating Louvain and MF approaches for calculating RMSE value.

	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)
MovieLens-1M	79925.78	2.41	728.98	14.81	24.11 (11)	37.42 (5)	40.83 (15)	33.34 (23)

community division. As the communities are increased, for all the different latent features the value remains constant. For the NMF method, the time taken at c value 1 community is less compared to while the communities are iterated. As the communities increased, the time taken has ups and falls for all the different latent features. Even if the integrated approach takes more time than only using the NMF approach, the RMSE value is very less. In the figure for the SVD++, 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 much less. We get better results for time when the latent feature value is low. 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 FANMF approach. As the time taken by the integrated approach is less compared to the MF approach, indicating the reduction of time complexity. By using

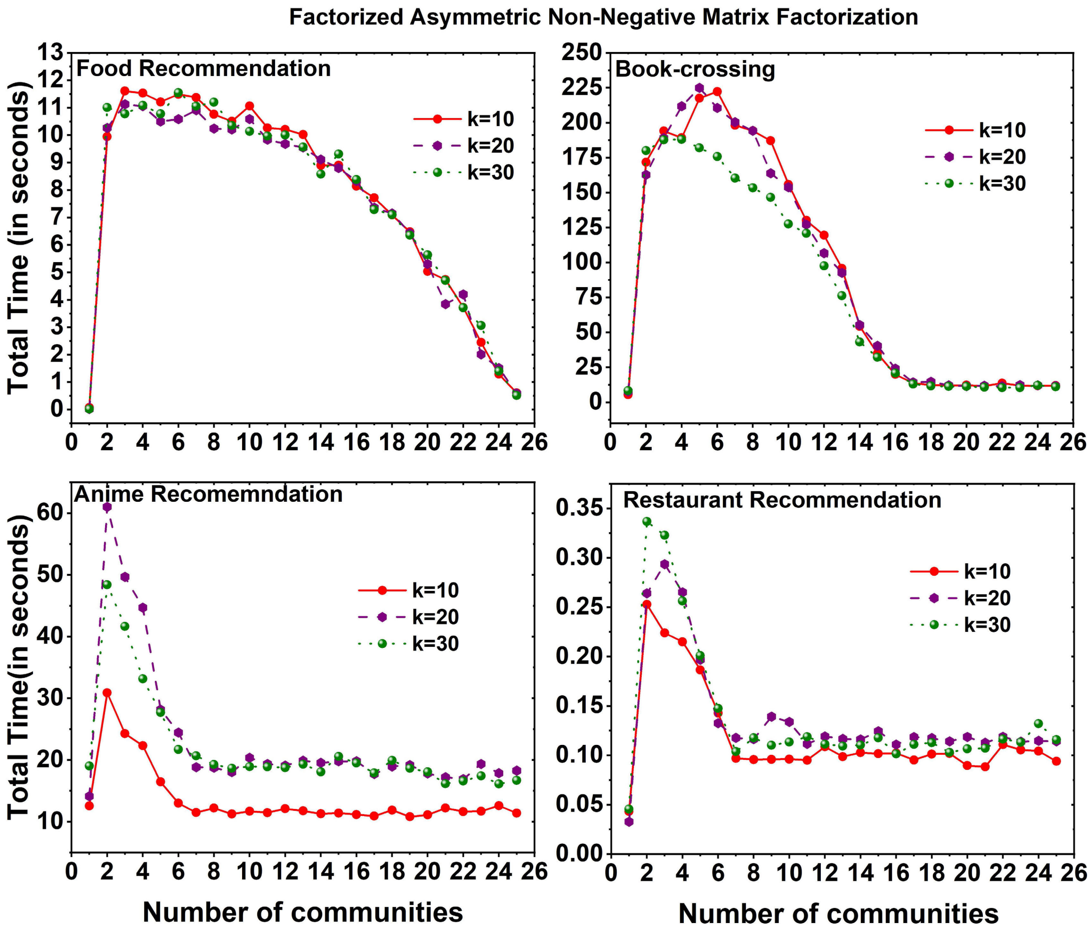

FIGURE 15. Comparing computational time for FANMF method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

the MovieLens-1M dataset, we ensure that our proposed technique is very effective in terms of computational time for handling large datasets for different matrix factorization techniques integrating with the Louvain community detection method.

Table 6 provides the time assessment for the five 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 this method takes less time for computation. Hence, we say that our proposed approach is better for calculating the RMSE value within less time.

Fig. 17 depicts the bar plot for the comparative analysis graph of the RMSE evaluation metric for different datasets. The two colors in the plot indicate the green where the metric value shown is without using the community, and the red color indicates with using the community. In the brackets, the value indicates the best community number where the RMSE value is low. It is observed from the figure that only using the matrix factorization approaches gives more RMSE value. When the matrix factorization is integrated with the Louvain community detection method there is a drastic decrease in the RMSE value for all the methods and for all the datasets. Thus, we can say that when community information is introduced to the matrix factorization approaches gives the best result.

C. DISCUSSIONS ON MSE RESULTS

Table 7 compares the results of the MSE values for five different datasets for four different MF methods by not using and using the community approach. The table contains an in-depth examination of the MSE values obtained both

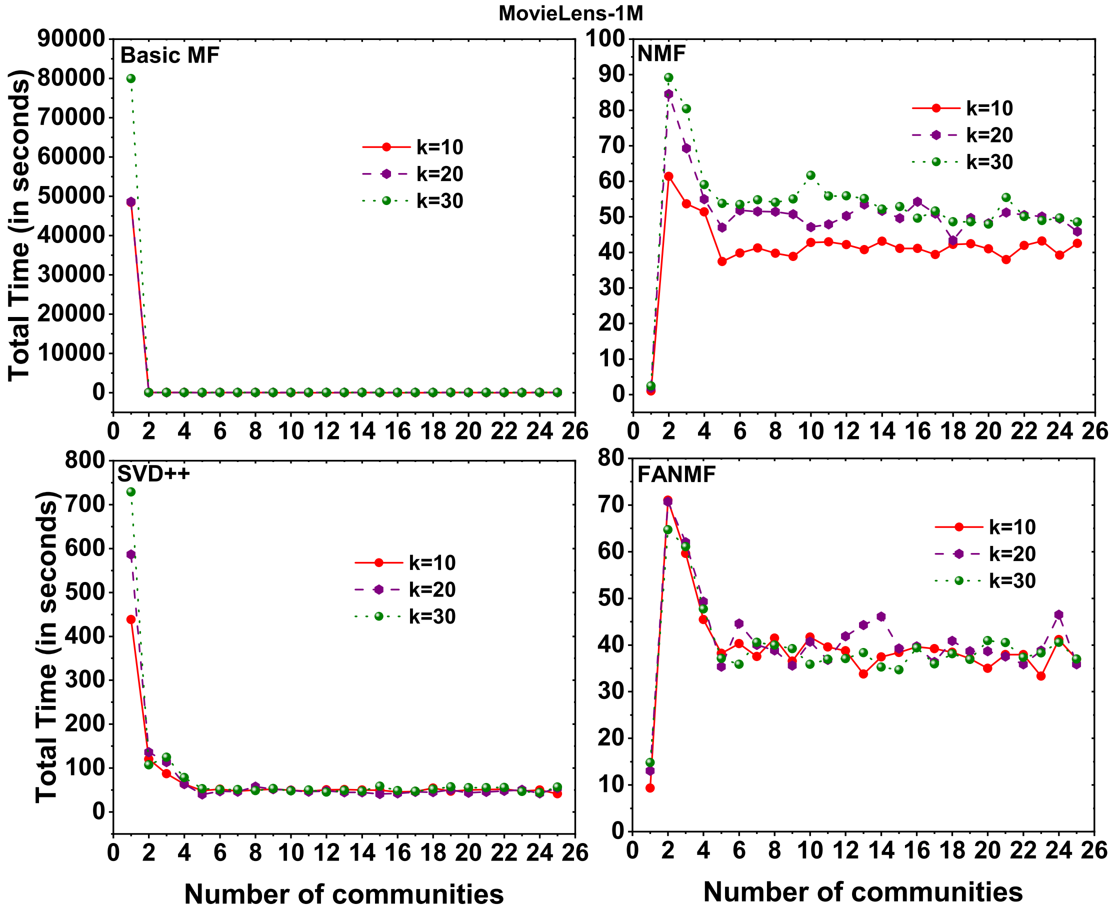

FIGURE 16. Comparing computational time for the Basic MF, NMF, SVD++, and FANMF methods across different latent features and communities for movielens-1M dataset.

TABLE 7. Comparison of MSE values for five 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	36.66	0.44	1.0	0.45	0.01 (18)	0.0002 (6)	0.01 (12)	0.00002 (7)
Book-crossing	59.81	0.07	0.29	0.07	0.02 (20)	0.004 (11)	0.001 (12)	0.0003 (7)
Anime Recommendation	54.91	0.5	1.52	0.5	0.39 (12)	0.003 (9)	0.04 (19)	0.36 (10)
Restaurant Recommendation	11.84	0.03	0.22	0.07	0.11 (16)	0.0004 (8)	0.01 (4)	0.0002 (25)
MovieLens-1M	45.28	0.01	1.16	0.37	0.13 (9)	0.19 (14)	0.0001 (7)	0.0004 (10)

TABLE 8. Comparison of time (in seconds) for five different datasets on different MF methods by integrating Louvain and MF approaches Vs by not integrating Louvain and MF approaches for calculating MSE value.

	without using community				with using community (number of communities)			
MF Method (→) / Dataset (↓)	Basic MF	NMF	SVD++	FANMF	Basic MF	NMF	SVD++	FANMF
Food Recommendation	15.68	0.33	0.94	0.08	0.23 (25)	1.44 (25)	1.21 (25)	0.43 (25)
Book-crossing	7039.27	4.42	86306.34	8.96	11.62 (19)	4.82 (16)	3.71 (20)	8.07 (19)
Anime Recommendation	16773.47	4.35	288.41	19.54	43.55 (23)	25.39 (14)	29.17 (23)	10.80 (19)
Restaurant Recommendation	23.30	0.23	1.45	0.04	0.12 (23)	0.31 (13)	0.16 (21)	0.08 (7)
MovieLens-1M	90578.47	3.30	704.45	16.38	42.64 (9)	42.91 (23)	39.01 (20)	19.36 (25)

without and with the community approach integrated into the MF technique. Using the proposed method, we have shown in brackets the community number at which the MSE value is low. The table shows that when we do not use

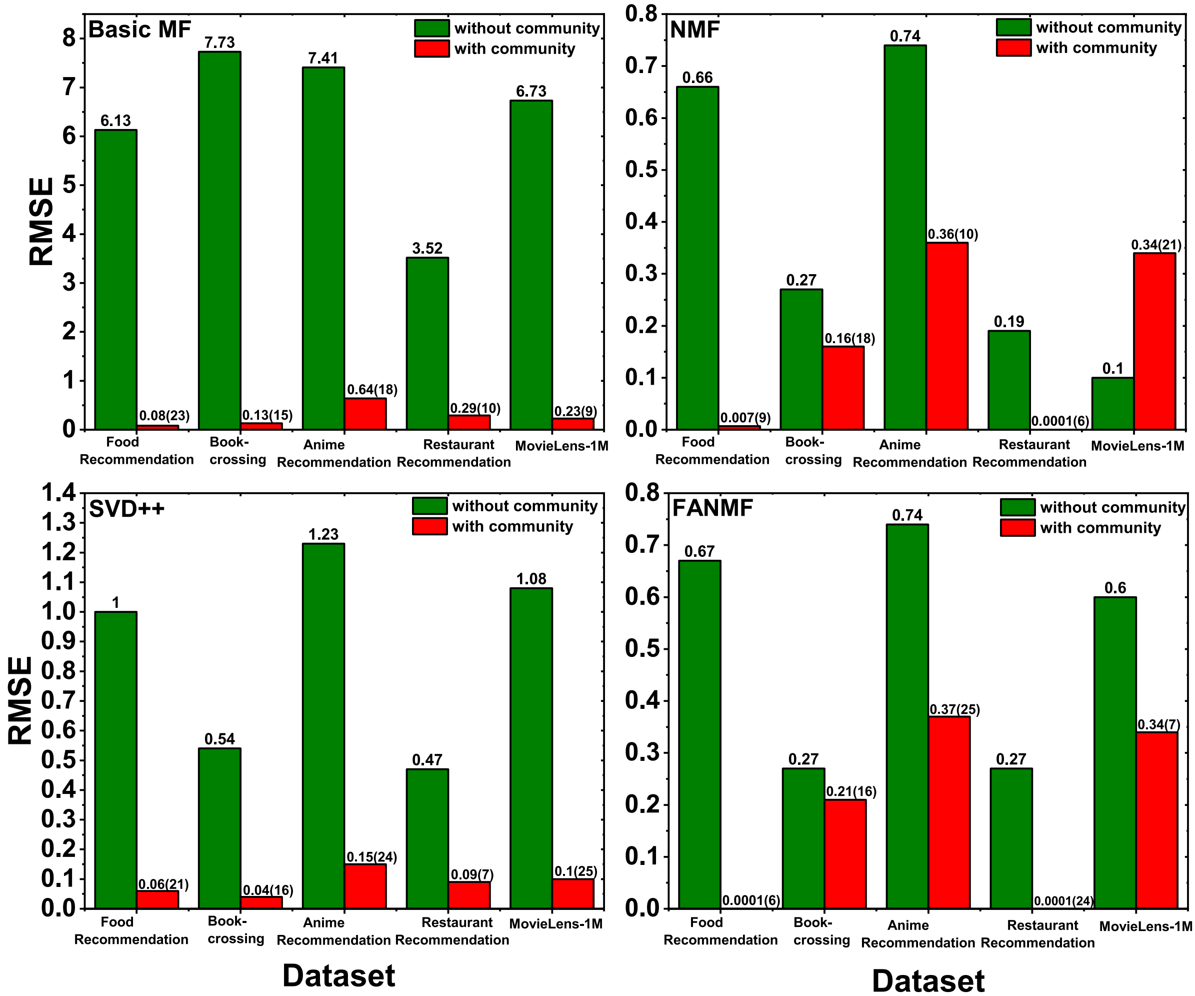

FIGURE 17. Comparative analysis of the RMSE metric for the Basic MF, NMF, SVD++, and FANMF methods for food recommendation, book-crossing, anime recommendation, restaurant recommendation, and movielens-1M datasets using without community and with community showing the best community number in braces.

TABLE 9. Comparison of MAE values for five 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	5.71	0.33	0.16	0.11	0.004 (18)	0.01 (14)	0.002 (17)	0.00001 (6)
Book-crossing	7.63	0.01	0.04	0.01	0.01 (11)	0.0004 (12)	0.0002 (14)	0.0008 (10)
Anime Recommendation	7.19	0.15	0.19	0.15	0.1 (16)	0.0002 (6)	0.0008 (7)	0.0002 (5)
Restaurant Recommendation	3.20	0.3	0.14	0.09	0.04 (10)	0.001 (4)	0.008 (12)	0.0009 (17)
MovieLens-1M	6.5	0.006	0.3	0.21	0.06 (6)	0.0002 (22)	0.0004 (6)	0.0006 (5)

the community approach, the MSE value is huge, however, when we use the Louvain community approach, which integrates with the MF technique, the MSE value is reduced. For instance, we observe that the food recommendation dataset shows a better score of MSE when divided into 18 communities for the basic MF method, 6 communities for the NMF method, 12 communities for the SVD++ method, and 7 communities for the FANMF method. For the book-crossing dataset, we observe a better score when divided into 20 communities for the basic MF method, 11 communities for the NMF method, 12 communities for the SVD++ method, and 7 communities for the FANMF method. In the anime recommendation dataset, it is observed that a better score of MSE value is seen at 12 communities for the basic MF method, 9 communities for the NMF method, 19 communities for the SVD++ method, and at 10 communities for the FANMF method. Similarly, for the restaurant recommendation dataset, the

TABLE 10. Comparison of time (in seconds) for five different datasets on different MF methods by integrating Louvain and MF approaches Vs by not integrating Louvain and MF approaches for calculating MAE value.

	without using community				with using community (number of communities)			
MF Method (→) / Dataset (↓)	Basic MF	NMF	SVD++	FANMF	Basic MF	NMF	SVD++	FANMF
Food Recommendation	20.40	0.05	0.8	0.21	0.46 (25)	0.05 (25)	0.94 (25)	0.36 (25)
Book-crossing	6304.03	0.73	31.79	18.95	12.16 (19)	23.56 (20)	5.47 (24)	10.46 (20)
Anime Recommendation	17422.04	0.84	166.48	34.84	43.44 (12)	24.64 (9)	12.24 (25)	9.92 (9)
Restaurant Recommendation	34.12	0.06	1.67	0.31	0.12 (23)	0.05 (7)	0.17 (17)	0.07 (7)
MovieLens-1M	126925.4	2.53	731.53	57.82	44.02 (20)	5.63 (7)	17.87 (10)	47.01 (10)

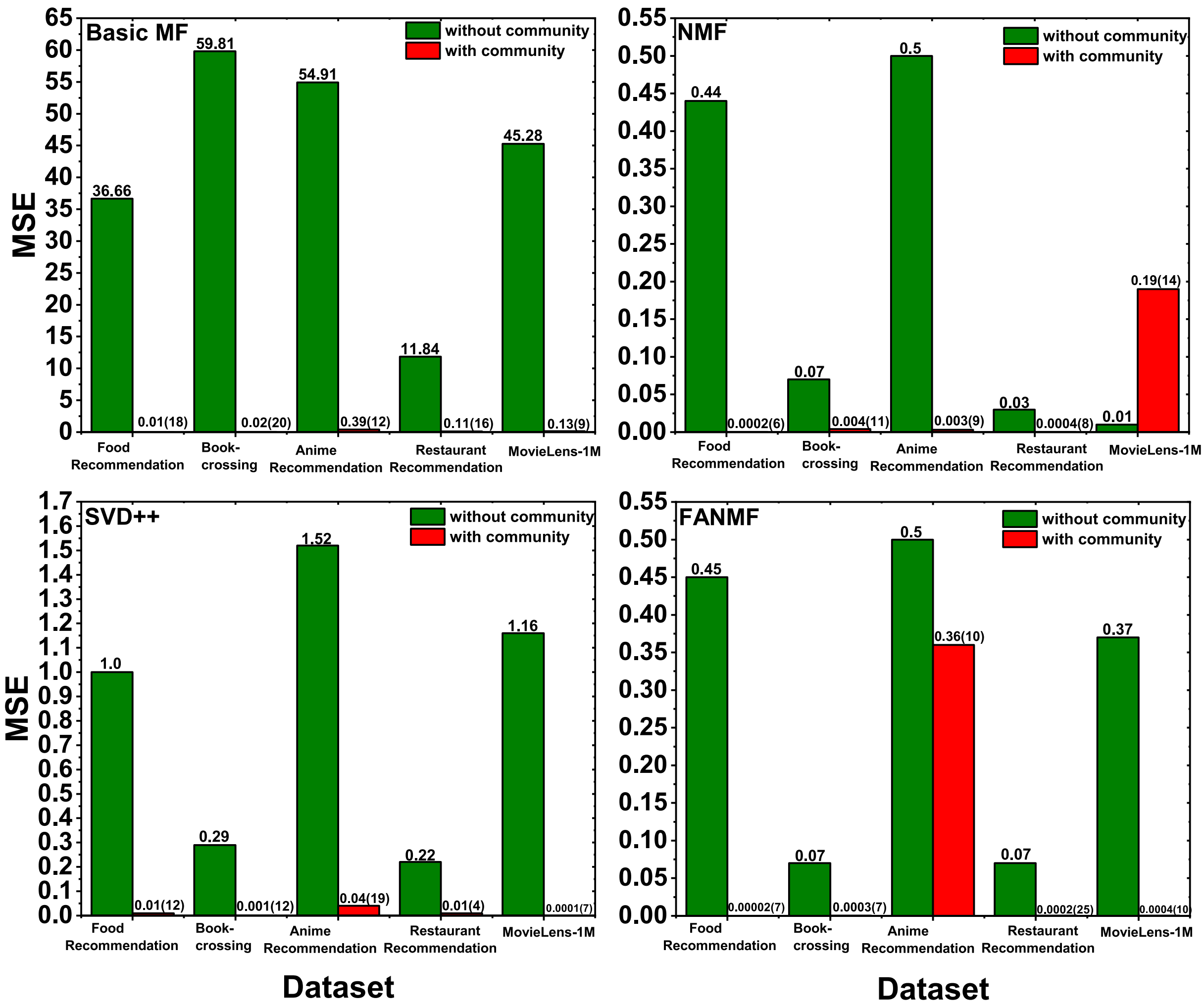

FIGURE 18. Comparative analysis of the MSE metric for the Basic MF, NMF, SVD++, and FANMF methods for food recommendation, book-crossing, anime recommendation, restaurant recommendation, and movielens-1M datasets using without community and with community showing the best community number in braces.

better score of MSE value is observed at 16 communities for the basic MF method, 8 communities for the NMF method, 4 communities for the SVD++ method, and at 25 communities for the FANMF method. For the movielens-1M dataset, we observe a better score when divided into 9 communities for the basic MF method, 14 communities for the NMF method, 7 communities for the SVD++ method, and 10 communities for the FANMF method. From the five 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, it outperforms the non-utilization of the community approach with the MF method. This ensures improving the recommendation accuracy for the users. The graph plotting of MSE performance metric for various matrix factorization techniques

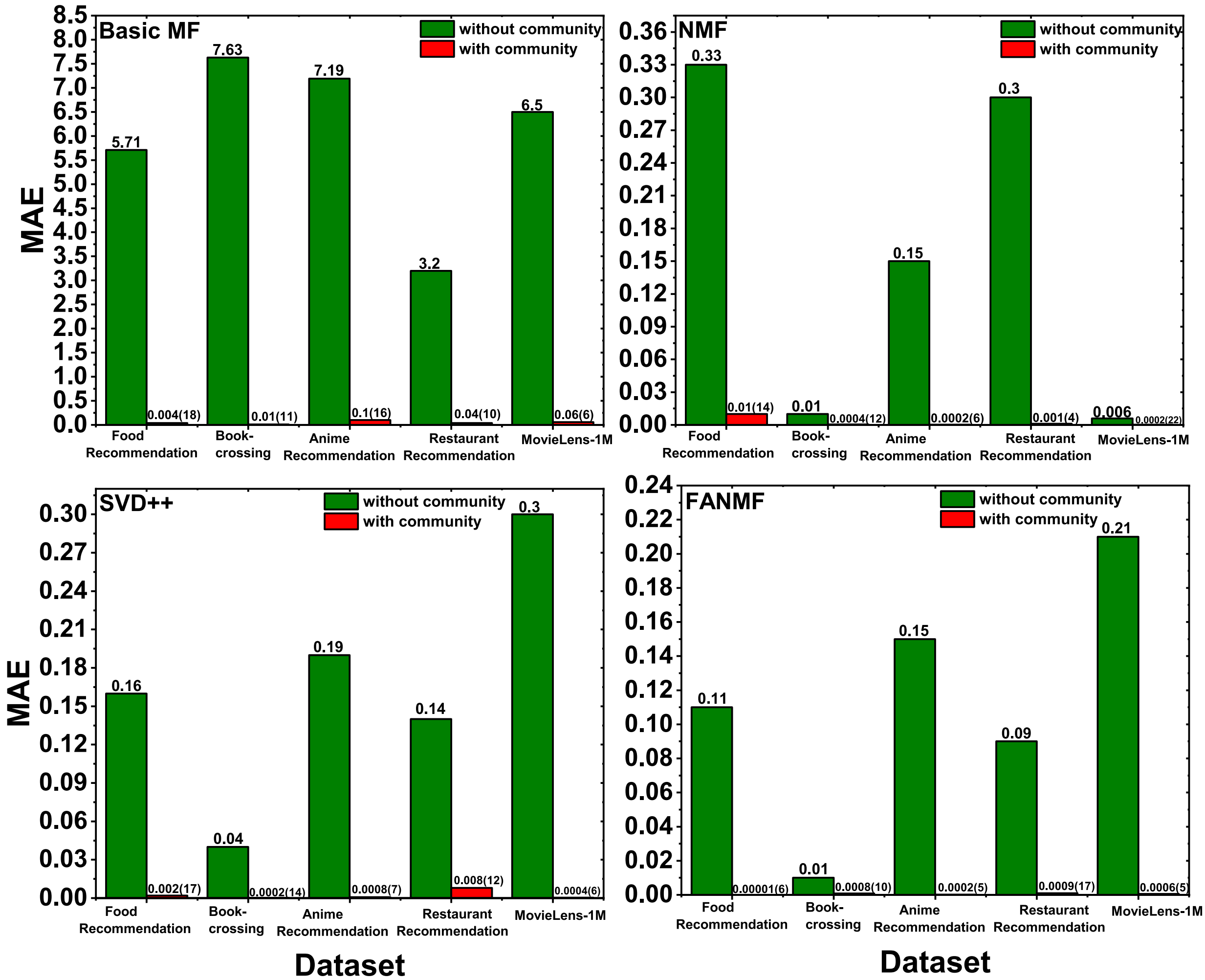

FIGURE 19. Comparative analysis of the MAE metric for the Basic MF, NMF, SVD++, and FANMF methods for food recommendation, book-crossing, anime recommendation, restaurant recommendation, and movielens-1M datasets using without community and with community showing the best community number in braces.

and five different datasets is shown in Supplementary Section.

Table 8 provides the time assessment for the five 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 using the MSE metric. 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 calculation of MSE. 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. Hence, we say that our proposed approach is better for calculating the MSE value within less time.

Fig. 18 depicts the bar plot for the comparative analysis graph of the MSE evaluation metric for different datasets. The green color in the plot indicates without using the community whereas the red color in the plot indicates with community. The particular community number where the MSE value is low is shown in brackets. It is observed from the figure that only using the matrix factorization approaches gives more MSE value. When the matrix factorization is integrated with the Louvain community detection method there is a drastic decrease in the MSE value for all the methods and for all the datasets. Thus, we can say that when community information is introduced to the matrix factorization approaches gives the best result.

D. DISCUSSIONS ON MAE RESULTS

Table 9 provides the comparison of the results of the MAE values for five different datasets for four different MF methods by not using and using the community approach. The table provides a detailed analysis of the MAE 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 MAE 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 MAE value is high, and when by using the Louvain community approach that integrates with the MF method, we observe a less MAE value. For instance, we observe that the food recommendation dataset shows a better score of MAE when divided into 18 communities for the basic MF method, 14 communities for the NMF method, 17 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 11 communities for the basic MF method, 12 communities for the NMF method, 14 communities for the SVD++ method, and 10 communities for the FANMF method. In the anime recommendation dataset, it is observed that a better score of MAE value is seen at 16 communities for the basic MF method, 6 communities for the NMF method, 7 communities for the SVD++ method, and at 5 communities for the FANMF method. Similarly, for the restaurant recommendation dataset, the better score of MAE value is observed at 10 communities for the basic MF method, 4 communities for the NMF method, 12 communities for the SVD++ method, and at 17 communities for the FANMF method. For the movielens-1M dataset, we observe a better score when divided into 6 communities for the basic MF method, 22 communities for the NMF method, 6 communities for the SVD++ method, and 5 communities for the FANMF method. From the five 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, it outperforms the non-utilization of the community approach with the MF method. The result ensures that our proposed approach improves the recommendation accuracy of the users. The graph plotting of MAE performance metric for various matrix factorization techniques and five different datasets is shown in Supplementary Section.

Table 10 provides the time assessment for the five 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 using the MAE metric. 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 calculation of MAE. 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. Hence, we say that our proposed approach is better for calculating the MAE value within less time.

Fig. 19 depicts the bar plot for the comparative analysis graph of the MAE evaluation metric for different datasets. The two colors in the plot indicate the green where the metric value shown is without using the community and the red color indicates with using the community. In the brackets, the value indicates the best community number where the MAE value is low. It is observed from the figure that only using the matrix factorization approaches gives more MAE value. When the matrix factorization is integrated with the Louvain community detection method there is a drastic decrease in the MAE value for all the methods and all the datasets. Thus, we can say that when community information is introduced to the matrix factorization approaches gives the best result.

VI. ADVANTAGES AND LIMITATIONS

The advantages of our proposed approach are as follows: By employing community detection algorithms to cluster users and items tightly within a bipartite network, our proposed approach restricts MF predictions to those clusters. This focuses on reducing the false recommendations and improving the accuracy. Limiting MF predictions to identified communities enhances the relevance of recommendations, ensuring users receive suggestions closely aligned with their interests and consumption patterns within their respective communities. By reducing the number of user-item combinations processed, the approach improves the time complexity of MF methods, enhancing the efficiency and scalability of recommendation generation, especially for large datasets. Our approach can better accommodate diverse user preferences and behaviors, thereby improving overall user satisfaction with the recommendation system.

The two notable limitations of our proposed approach are: If the communities within the data are not well-defined or properly structured, the effectiveness of this approach diminishes. Handling huge datasets can present substantial computational challenges due to the sheer volume of data. Sparse ratings can lead to numerous weakly connected or isolated nodes within the network. In some practical scenarios where our proposed approach has limitations like in considering the contextual factors based on user mood and behavior, the recommendations become irrelevant. In the music platform where if the user is struck with the same genre and style of music lacks the novelty for fresh experiences. In scenarios like finance, and law where strict regulations are made may limit our approach.

VII. 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

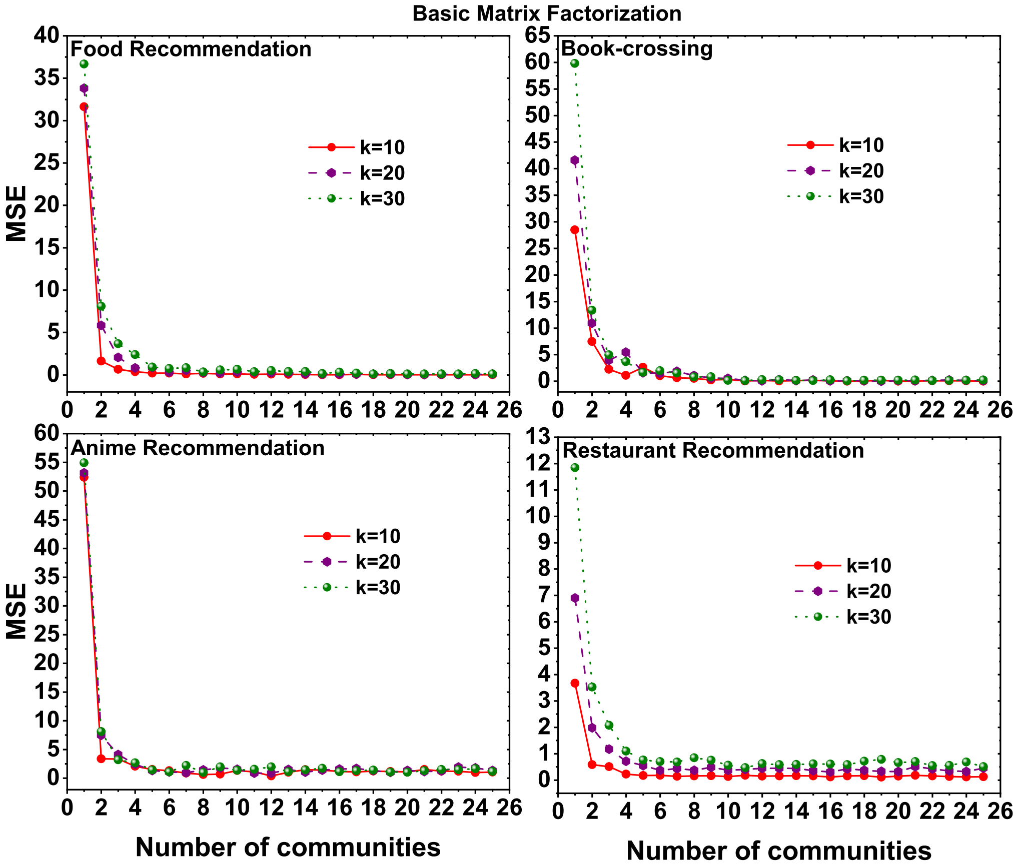

FIGURE 20. Examining the MSE metrics for the basic matrix factorization method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

algorithm in community detection tasks. We have explored computational efficiency in different domains regarding the suggested approach's time and evaluation using different performance metrics like RMSE, MSE, and MAE. The results show that better recommendations can be provided by using our proposed approach. The well-structured communities are also formed 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 user recommendations based on their experience. The primary benefit is emphasizing the 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 address real-world challenges and handle the data 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. We can consider the recommendation system in terms of the classification problem for improving the model's performance and accurate recommendations for the users. In the future, we would like to implement advanced techniques like temporal models, and deep learning to improve our model's performance.

VIII. ABBREVIATIONS

The following are the abbreviations used in this paper:

BG	Bipartite Graph.
RM	Rating Matrix.
$\widetilde{RM}$	Predicted Rating Matrix.
MF	Matrix Factorization.
NMF	Non-Negative Matrix Factorization.
SVD	Singular Value Decomposition.
SVD++	Advanced Singular Value Decomposition.
FANMF	Factorized Asymmetric Non-Negative Matrix Factorization.
RMSE	Root Mean Square Error.
MSE	Mean Square Error.
MAE	Mean Absolute Error.

SUPPLEMENTARY INFORMATION

A. DISCUSSIONS ON MSE RESULTS

Fig. 20 shows the MSE value on four datasets for 25 communities and different latent features for $k = 10$, 20, and

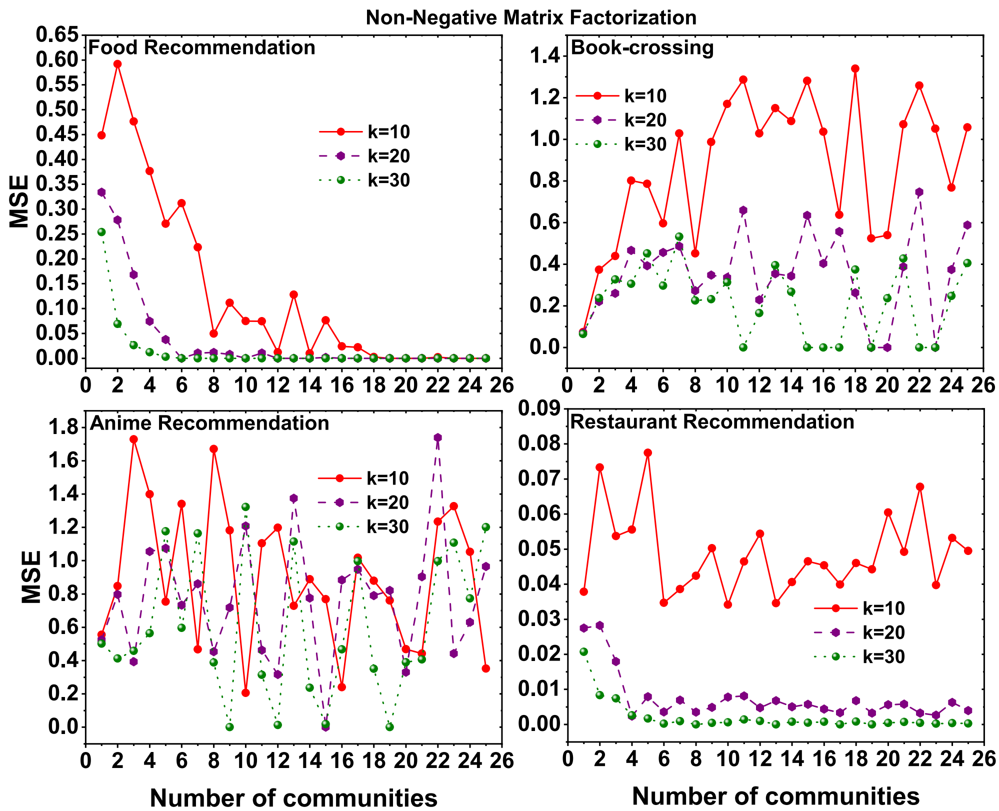

FIGURE 21. Examining the MSE metrics for the NMF method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

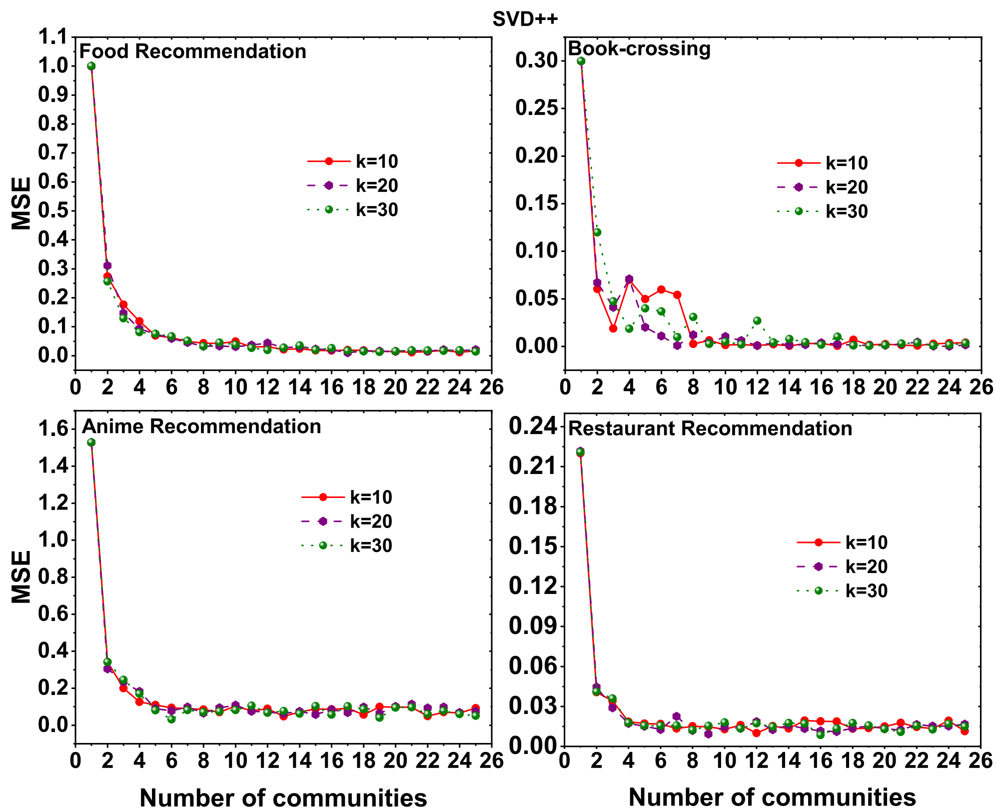

FIGURE 22. Examining the MSE metrics for the SVD++ method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

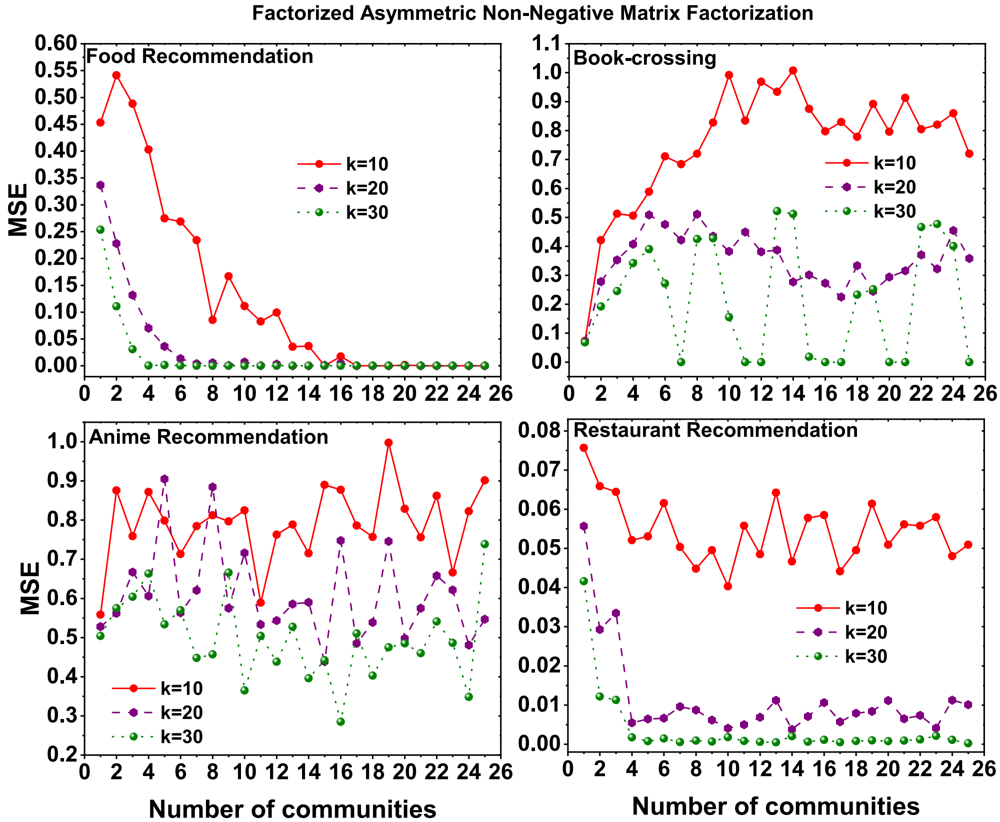

FIGURE 23. Examining the MSE metrics for the FANMF method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

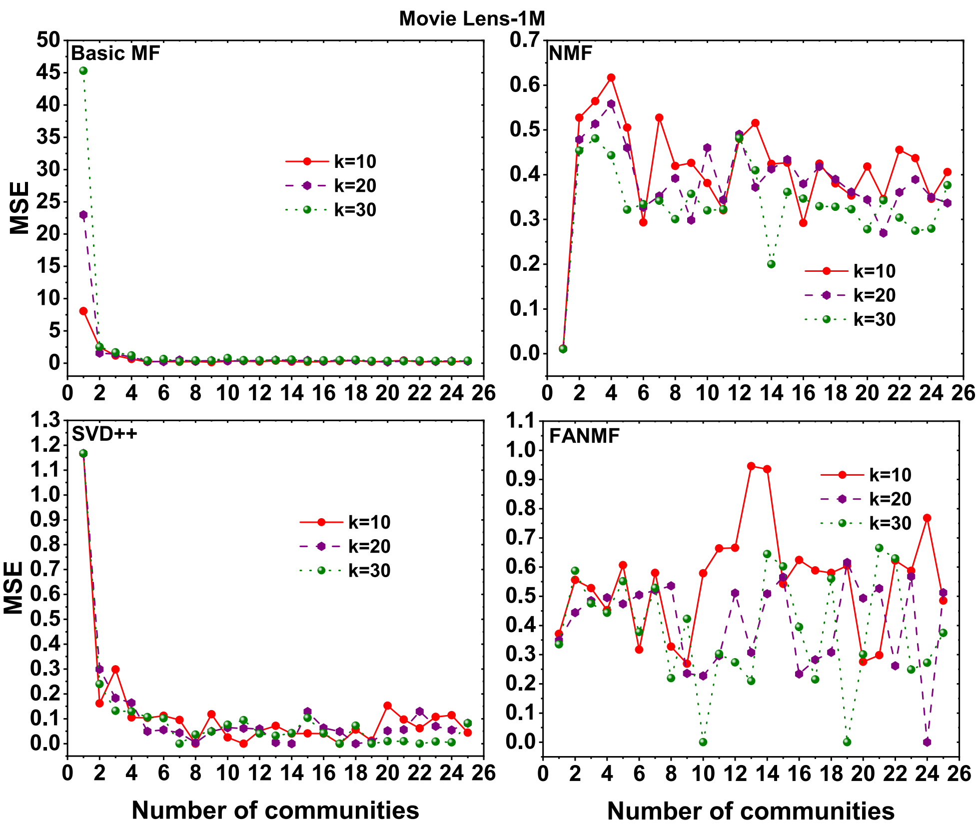

FIGURE 24. Examining the MSE metric for the Basic MF, NMF, SVD++, and FANMF methods across different latent features and communities for movielens-1M dataset.

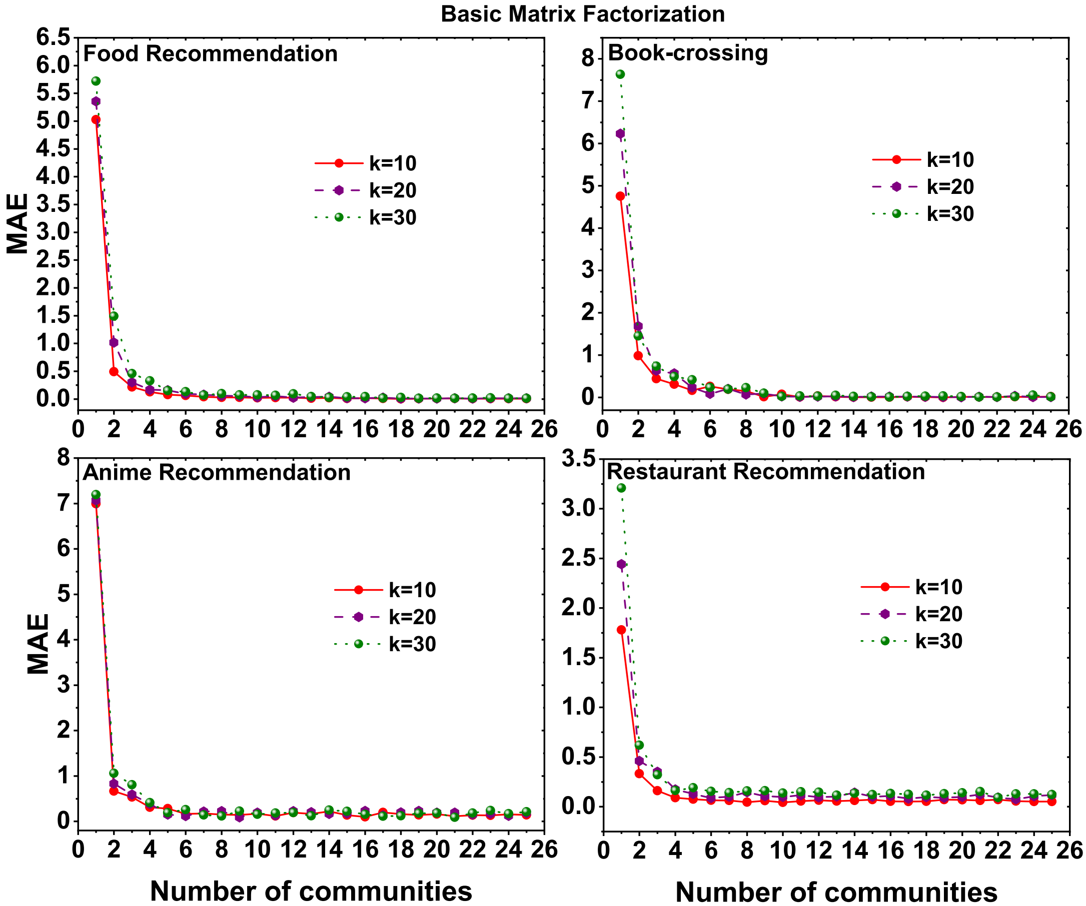

FIGURE 25. Examining the MAE metrics for the basic matrix factorization method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

30 for the basic MF method. It is observed in the figure that for all four datasets without using community detection at $c = 1$ for the basic MF method, the MSE value is high. When applying the community detection method along with the basic MF, there is a decrease in the MSE value as the communities increase. A clear difference is observed when not using community division and by using the community division. As the Louvain community detection method is applied and communities are increased, there is a drastic change in the MSE value. Furthermore, we observe that after a certain number of communities, the MSE value remains constant. This indicates the value of the better community division for the network.

Fig. 21 shows the MSE 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 all the datasets that as the number of latent features is increased, there is a decrease in the MSE value. There are severe ups and falls in the dataset as the number of communities is increased. In the food recommendation dataset, as the number of communities increased, there was a decrease in the MSE value. In the book-crossing, anime, and restaurant datasets, initially, without using community division, the value is low, but in certain points of communities, there is a low MSE value. Thus the community approach outperforms by getting a lower MSE value as the communities are increased.

Fig. 22 shows the MSE 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 without using community division at $c = 1$, the MSE value is high and when the Louvain community detection method is integrated with the SVD++ method when the communities are increased, there is a decrease in the MSE value. We observe that after a certain community division, the MSE value remains constant. This indicates the value of better community division for the network.

Fig. 23 shows the MSE value on four datasets for 25 communities and different latent features for $k = 10$, 20, and 30 for the FANMF method. It is observed in all the datasets that, as the latent features are increased there is a decrease in the MSE value. In the food recommendation dataset, without using community division, the MSE value is high. After community division is integrated with the MF method, there is a decrease in the MSE value as the communities are increased. It is maintained constant after a certain community division, which indicates the better community division of the network. For all the remaining

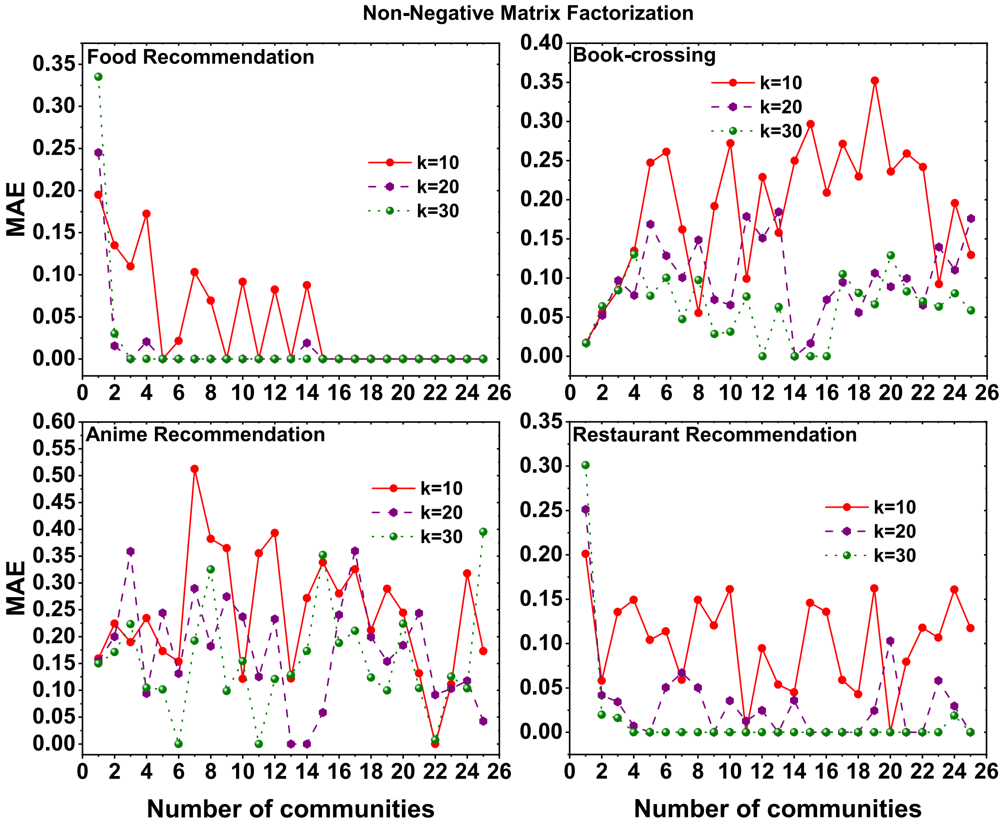

FIGURE 26. Examining the MAE metrics for the NMF method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

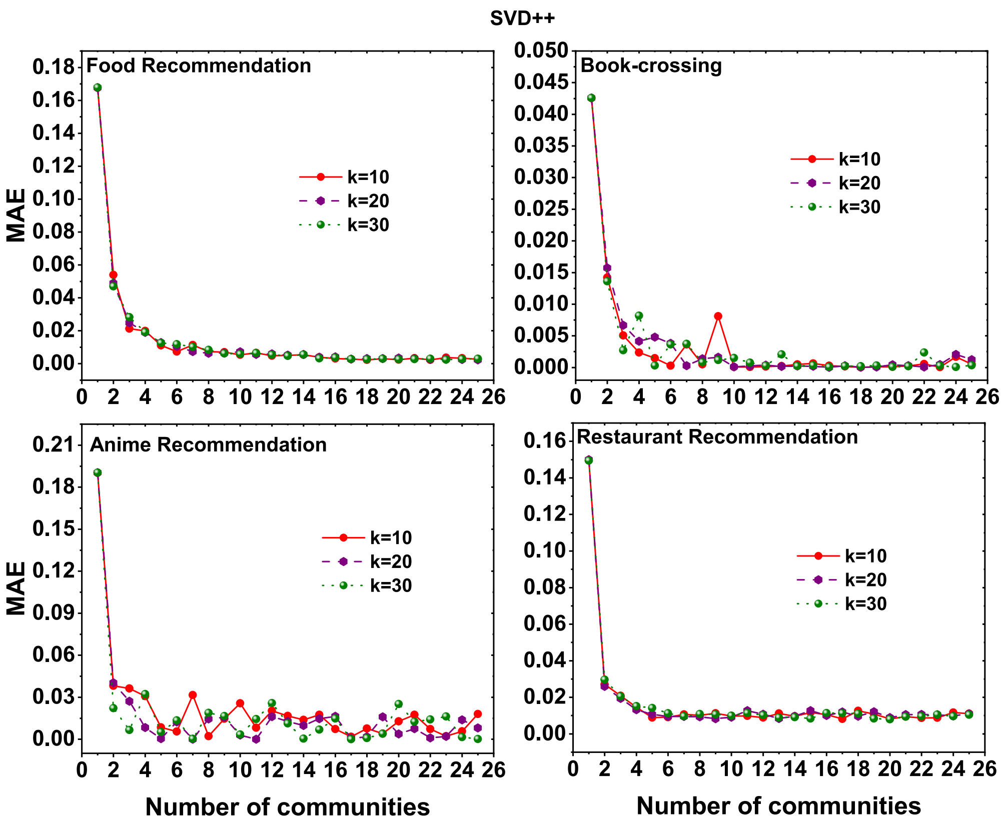

FIGURE 27. Examining the MAE metrics for the SVD++ method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

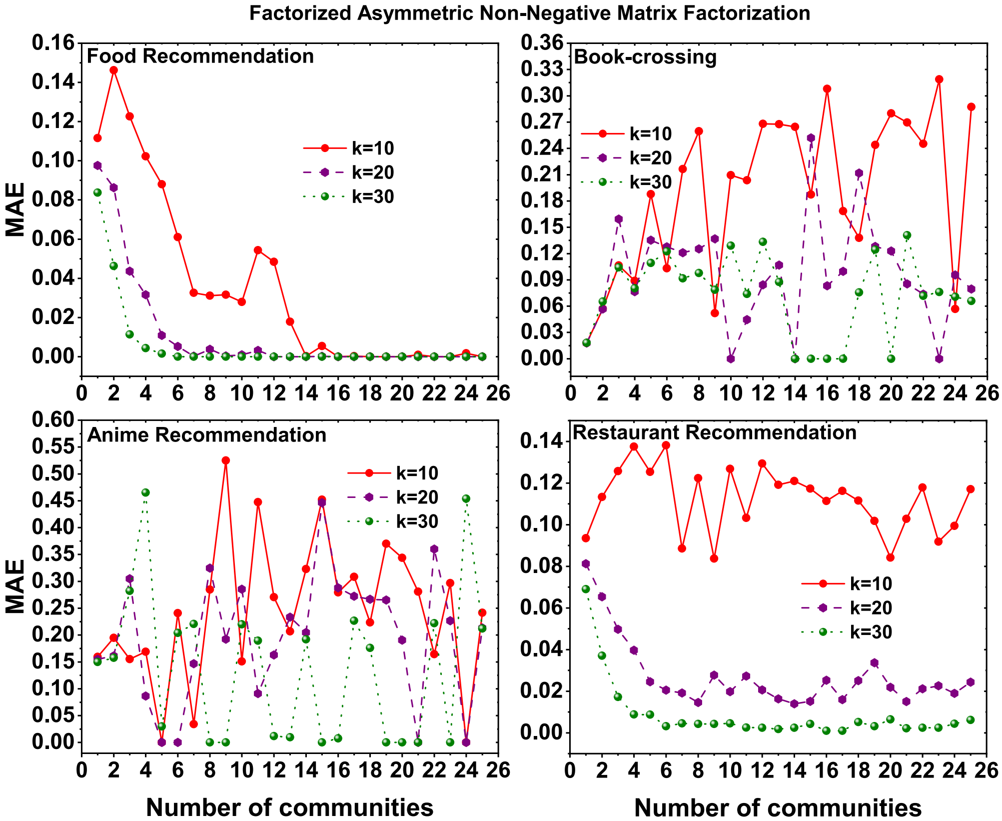

FIGURE 28. Examining the MAE metrics for the FANMF method across different latent features and communities for food recommendation, book-crossing, anime recommendation, and restaurant recommendation datasets.

datasets, it is observed that when the communities are increased, there are severe ups and falls in the MSE value. The MSE value is decreased after certain communities which indicates the better division of communities for the network is detected.

Fig. 24 shows the MSE value on MovieLens-1M dataset for 25 communities and different latent features for $k = 10$, 20, and 30 for the basic MF, NMF, SVD++, and FANMF methods. It is observed in basic MF and SVD++ methods, it is observed that when the $c = 1$, without using the community division, the MSE value is high, and when the community division is applied there is a decrease in the MSE value as the communities are increased. For the NMF and FANMF methods, at $c = 1$, there is a low MSE value, but when the MF method is integrated with the Louvain community division, at a certain point of communities, there is a decrease in the MSE value. As the number of latent features increases, the MSE value decreases for the increase in the number of communities.

B. DISCUSSIONS ON MAE RESULTS

Fig. 25 shows the MAE 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 all four datasets without using community detection at $c = 1$ for the basic MF method, the MAE value is high. When applying the community detection method along with the basic MF, there is a decrease in the MAE value as the communities increase. A clear difference is observed when not using community division and by using the community division. As the Louvain community detection method is applied and communities are increased, there is a drastic change in the MAE value. Furthermore, we observe that after a certain number of communities, the MAE value remains constant. This indicates the value of the better community division for the network.

Fig. 26 shows the MAE 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 all the datasets that as the number of latent features is increased, there is a decrease in the MAE value. There are severe ups and falls in the dataset as the number of communities is increased. In the food and restaurant recommendation datasets, as the number of communities increased, there was a decrease in the MAE value. In the book-crossing and anime datasets, initially, without using community division, the value is low, but in certain points of communities, there is a low MAE value. Thus the community approach outperforms by getting a lower MAE value as the communities are increased.

Fig. 27 shows the MAE 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 without using community division at $c = 1$, the MAE value is high and when the

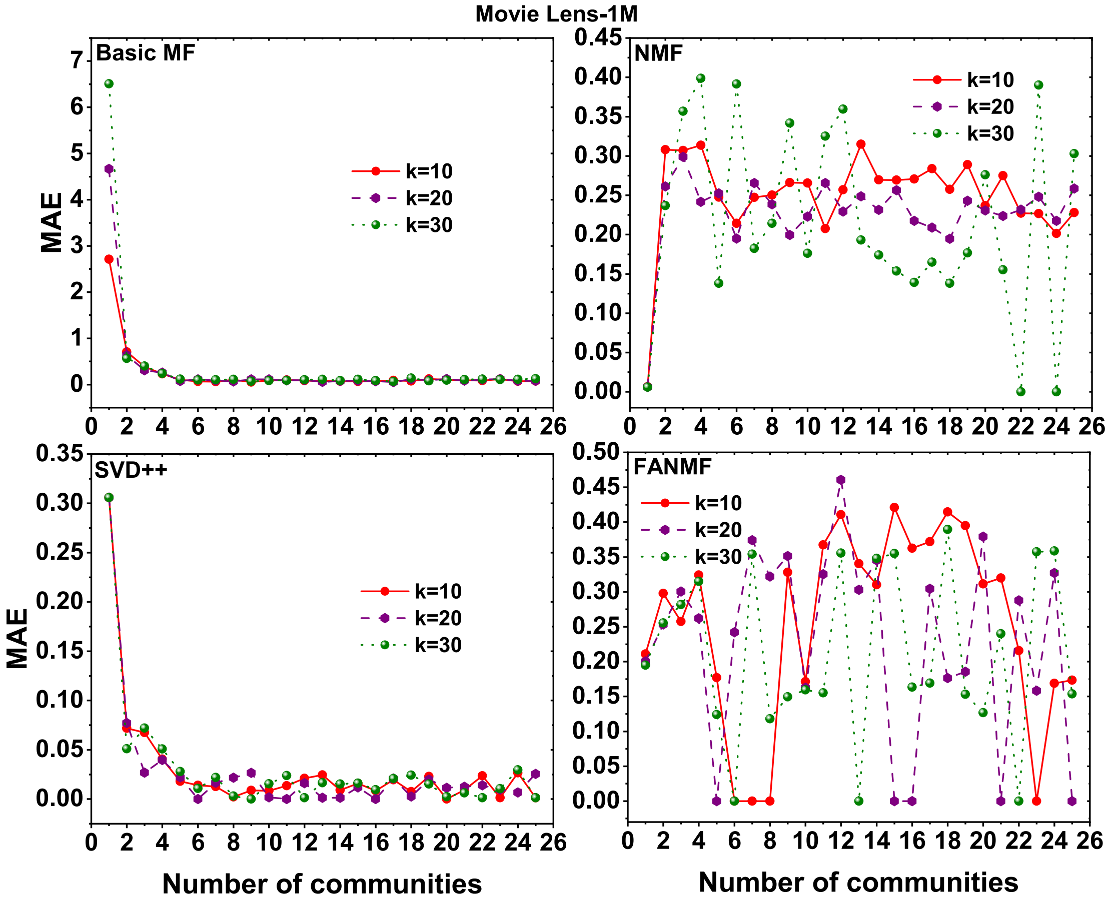

FIGURE 29. Examining the MAE metric for the Basic MF, NMF, SVD++, and FANMF methods across different latent features and communities for movielens-1M dataset.

Louvain community detection method is integrated with the SVD++ method when the communities are increased, there is a decrease in the MAE value. We observe that after a certain community division, the MAE value remains constant. This indicates the value of better community division for the network.

Fig. 28 shows the MAE value on four datasets for 25 communities and different latent features for $k = 10$, 20, and 30 for the FANMF method. It is observed in all the datasets that, as the latent features are increased there is a decrease in the MAE value. In the food recommendation, and restaurant recommendation datasets, without using community division, the MAE value is high. After community division is integrated with the MF method, there is a decrease in the MAE value as the communities are increased. It is maintained constant after a certain community division, which indicates the better community division of the network. For all the remaining datasets, it is observed that when the communities are increased, there are severe ups and falls in the MAE value. The MAE value is decreased after certain communities which indicates the better division of communities for the network is detected.

Fig. 29 shows the MAE value on MovieLens-1M dataset for 25 communities and different latent features for $k = 10$, 20, and 30 for the basic MF, NMF, SVD++, and FANMF methods. It is observed in basic MF and SVD++ methods, it is observed that when the $c = 1$, without using the community division, the MAE value is high, and when the community division is applied there is a decrease in the MAE value as the communities are increased. For the NMF and FANMF methods, at $c = 1$, there is a low MAE value, but when the MF method is integrated with the Louvain community division, at a certain point of communities, there is a decrease in the MAE value. As the number of latent features increases, the MAE value decreases for the increase in the number of communities.

REFERENCES

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl.-Based Syst.*, vol. 46, pp. 109–132, Jul. 2013.
[2] A. Abdul, J. Chen, H.-Y. Liao, and S.-H. Chang, "An emotion-aware personalized music recommendation system using a convolutional neural networks approach," *Appl. Sci.*, vol. 8, no. 7, p. 1103, Jul. 2018.
[3] G. Geetha, M. Safa, C. Fancy, and D. Saranya, "A hybrid approach using collaborative filtering and content based filtering for recommender system," *J. Phys., Conf. Ser.*, vol. 1000, Apr. 2018, Art. no. 012101.

[4] J. Chen and A. Abdul, "A session-based customer preference learning method by using the gated recurrent units with attention function," *IEEE Access*, vol. 7, pp. 17750–17759, 2019.

[5] R. Van Meteren and M. Van Someren, "Using content-based filtering for recommendation," in *Proc. Mach. Learn. New Inf. Age*, vol. 30, 2000, pp. 47–56.

[6] Q. Liu, N. Chen, T. Sakai, and X.-M. Wu, "ONCE: Boosting content-based recommendation with both Open- and closed-source large language models," in *Proc. 17th ACM Int. Conf. Web Search Data Mining*, Mar. 2024, pp. 452–461.

[7] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The Adaptive Web: Methods and Strategies of Web Personalization*. Cham, Switzerland: Springer, 2007, pp. 291–324.

[8] D. Bokde, S. Girase, and D. Mukhopadhyay, "Matrix factorization model in collaborative filtering algorithms: A survey," *Proc. Comput. Sci.*, vol. 49, pp. 136–146, Jan. 2015.

[9] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[10] R. Mehta and K. Rana, "A review on matrix factorization techniques in recommender systems," in *Proc. 2nd Int. Conf. Commun. Syst., Comput. IT Appl. (CSCITA)*, Apr. 2017, pp. 269–274.

[11] S. Marinetti, L. Finesso, and E. Marsilio, "Matrix factorization methods: Application to thermal NDT/E," *NDT & E Int.*, vol. 39, no. 8, pp. 611–616, Dec. 2006.

[12] B. Barney, "Introduction to parallel computing," *Lawrence Livermore Nat. Lab.*, vol. 6, no. 13, p. 10, 2010.

[13] K. Asanovic, R. Bodik, J. Demmel, T. Keaveny, K. Keutzer, J. Kubiatowicz, N. Morgan, D. Patterson, K. Sen, and J. Wawrzynek, "A view of the parallel computing landscape," *Commun. ACM*, vol. 52, no. 10, pp. 56–67, 2009.

[14] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, no. 6825, pp. 268–276, Mar. 2001.

[15] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Phys. Rep.*, vol. 659, pp. 1–44, Nov. 2016.

[16] B. M. Maggs, L. R. Matheson, and R. E. Tarjan, "Models of parallel computation: A survey and synthesis," in *Proc. 28th Hawaii Int. Conf. Syst. Sci.*, 1995, pp. 61–70.

[17] S. Tokala, M. K. Enduri, and T. J. Lakshmi, "Unleashing the power of SVD and Louvain community detection for enhanced recommendations," in *Proc. IEEE 15th Int. Conf. Comput. Intell. Commun. Netw. (CICN)*, Dec. 2023, pp. 807–811.

[18] X. Que, F. Checconi, F. Petrini, and J. A. Gunnels, "Scalable community detection with the Louvain algorithm," in *Proc. IEEE Int. Parallel Distrib. Process. Symp.*, May 2015, pp. 28–37.

[19] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, Feb. 2010.

[20] S. Sahu, "DF Louvain: Fast incrementally expanding approach for community detection on dynamic graphs," 2024, *arXiv:2404.19634*.

[21] P. Chunaev, "Community detection in node-attributed social networks: A survey," *Comput. Sci. Rev.*, vol. 37, Aug. 2020, Art. no. 100286.

[22] J. Yang and J. Leskovec, "Overlapping community detection at scale: A nonnegative matrix factorization approach," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, Feb. 2013, pp. 587–596.

[23] F. Ye, C. Chen, and Z. Zheng, "Deep autoencoder-like nonnegative matrix factorization for community detection," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 1393–1402.

[24] M. A. Javed, M. S. Younis, S. Latif, J. Qadir, and A. Baig, "Community detection in networks: A multidisciplinary review," *J. Netw. Comput. Appl.*, vol. 108, pp. 87–111, Apr. 2018.

[25] D. Huang, D. Mu, L. Yang, and X. Cai, "CoDetect: Financial fraud detection with anomaly feature detection," *IEEE Access*, vol. 6, pp. 19161–19174, 2018.

[26] H. Tong and C.-Y. Lin, "Non-negative residual matrix factorization with application to graph anomaly detection," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2011, pp. 143–153.

[27] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst.*, Sep. 2010, pp. 135–142.

[28] V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joye, N. Taft, and D. Boneh, "Privacy-preserving matrix factorization," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 801–812.

[29] Y. Chen and X. Chen, "Research on knowledge graph application technology," *J. Phys., Conf. Ser.*, vol. 1187, no. 4, Apr. 2019, Art. no. 042083.

[30] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He, "A survey on knowledge graph-based recommender systems," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3549–3568, Aug. 2022.

[31] J. Ben Schafer, J. A. Konstan, and J. Riedl, "E-commerce recommendation applications," *Data Mining Knowl. Discovery*, vol. 5, no. 1, pp. 115–153, Jan. 2001.

[32] R. Widayanti, "Improving recommender systems using hybrid techniques of collaborative filtering and content-based filtering," *J. Appl. Data Sci.*, vol. 4, no. 3, pp. 289–302, Sep. 2023.

[33] H. Papadakis, A. Papagrigoriou, C. Panagiotakis, E. Kosmas, and P. Fragopoulou, "Collaborative filtering recommender systems taxonomy," *Knowl. Inf. Syst.*, vol. 64, no. 1, pp. 35–74, Jan. 2022.

[34] J. Hintz, "Matrix factorization for collaborative filtering recommender systems," Univ. Texas, Austin, Tech. Rep., 2015.

[35] E. Zangerle and C. Bauer, "Evaluating recommender systems: Survey and framework," *ACM Comput. Surv.*, vol. 55, no. 8, pp. 1–38, Aug. 2023.

[36] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in *Proc. 5th ACM Conf. Recommender Syst.*, Oct. 2011, pp. 301–304.

[37] N. Khaledian and F. Mardukhi, "CFMT: A collaborative filtering approach based on the nonnegative matrix factorization technique and trust relationships," *J. Ambient Intell. Humanized Comput.*, vol. 13, no. 5, pp. 2667–2683, May 2022.

[38] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2003, pp. 267–273.

[39] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2006, pp. 549–553.

[40] M. N. Schmidt, O. Winther, and L. K. Hansen, "Bayesian non-negative matrix factorization," in *Proc. Int. Conf. Independ. Compon. Anal. Signal Separat.*, Paraty, Brazil. Cham, Switzerland: Springer, 2009, pp. 540–547.

[41] X. Luo, Z. Liu, L. Jin, Y. Zhou, and M. Zhou, "Symmetric nonnegative matrix factorization-based community detection models and their convergence analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 1203–1215, Mar. 2022.

[42] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 263–272.

[43] S. Jiang, J. Li, and W. Zhou, "An application of SVD++ method in collaborative filtering," in *Proc. 17th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process. (ICCWAMTIP)*, Dec. 2020, pp. 192–197.

[44] W. Shi, L. Wang, and J. Qin, "User embedding for rating prediction in SVD++-based collaborative filtering," *Symmetry*, vol. 12, no. 1, p. 121, Jan. 2020.

[45] A. M. A. Al-Sabaawi, H. Karacan, and Y. E. Yenice, "SVD++ and clustering approaches to alleviating the cold-start problem for recommendation systems," *Int. J. Innov. Comput., Inf. Control*, vol. 17, no. 2, pp. 383–396, 2021.

[46] M. Hosseinzadeh Aghdam and M. Daryaie Zanjani, "A novel regularized asymmetric non-negative matrix factorization for text clustering," *Inf. Process. Manage.*, vol. 58, no. 6, Nov. 2021, Art. no. 102694.

[47] 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, Sep. 2023.

[48] 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," *Phys. A, Stat. Mech. Appl.*, vol. 506, pp. 975–986, Sep. 2018.

[49] S. Ghosh, M. Halappanavar, A. Tumeo, A. Kalyanaraman, H. Lu, D. Chavarrià-Miranda, A. Khan, and A. Gebremedhin, "Distributed Louvain algorithm for graph community detection," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2018, pp. 885–895.

[50] M. Mohammadi, M. Fazlali, and M. Hosseinzadeh, "Accelerating Louvain community detection algorithm on graphic processing unit," *J. Supercomput.*, vol. 77, no. 6, pp. 6056–6077, Jun. 2021.

[51] J. Zhang, J. Fei, X. Song, and J. Feng, "An improved Louvain algorithm for community detection," *Math. Problems Eng.*, vol. 2021, pp. 1–14, Nov. 2021.

[52] W. Zhang, "Improving commuting zones using the Louvain community detection algorithm," *Econ. Lett.*, vol. 219, Oct. 2022, Art. no. 110827.

[53] B. Hallinan and T. Striphas, "Recommended for you: The Netflix prize and the production of algorithmic culture," *New Media Soc.*, vol. 18, no. 1, pp. 117–137, Jan. 2016.
[54] R. Duan, C. Jiang, and H. K. Jain, "Combining review-based collaborative filtering and matrix factorization: A solution to rating's sparsity problem," *Decis. Support Syst.*, vol. 156, May 2022, Art. no. 113748.
[55] S. Venkatesan, "A recommender system based on matrix factorization techniques using collaborative filtering algorithm," *Neuroquantology*, vol. 21, no. 5, p. 864, 2023.
[56] J. Ivarsson and M. Lindgren, "Movie recommendations using matrix factorization," KTH Roy. Inst. Technol., Stockholm, Sweden, Tech. Rep., 2016.
[57] F. Esposito, "A review on initialization methods for nonnegative matrix factorization: Towards omics data experiments," *Mathematics*, vol. 9, no. 9, p. 1006, Apr. 2021.
[58] 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, Jun. 1994.
[59] 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, Oct. 1999.
[60] Y.-T. Guo, Q.-Q. Li, and C.-S. Liang, "The rise of nonnegative matrix factorization: Algorithms and applications," *Inf. Syst.*, vol. 123, Jul. 2024, Art. no. 102379.
[61] R. Kumar, B. K. Verma, and S. Sunder Rastogi, "Social popularity based SVD++ recommender system," *Int. J. Comput. Appl.*, vol. 87, no. 14, pp. 33–37, Feb. 2014.
[62] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 995–1000.
[63] A. Tosyali, J. Kim, J. Choi, and M. K. Jeong, "Regularized asymmetric nonnegative matrix factorization for clustering in directed networks," *Pattern Recognit. Lett.*, vol. 125, pp. 750–757, Jul. 2019.
[64] V. Blondel, J.-L. Guillaume, and R. Lambiotte, "Fast unfolding of communities in large networks: 15 years later," 2023, *arXiv:2311.06047*.
[65] N. S. Sattar and S. Arifuzzaman, "Scalable distributed Louvain algorithm for community detection in large graphs," *J. Supercomput.*, vol. 78, no. 7, pp. 10275–10309, May 2022.
[66] M. Seifikar, S. Farzi, and M. Barati, "C-blondel: An efficient louvain-based dynamic community detection algorithm," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 2, pp. 308–318, Apr. 2020.
[67] M. Fazlali, E. Moradi, and H. Tabatabaee Malazi, "Adaptive parallel Louvain community detection on a multicore platform," *Microprocessors Microsyst.*, vol. 54, pp. 26–34, Oct. 2017.
[68] M. E. Eddin, M. Massaoudi, H. Abu-Rub, M. Shadmand, and M. Abdallah, "Novel functional community detection in networked smart grid systems-based improved Louvain algorithm," in *Proc. IEEE Texas Power Energy Conf. (TPEC)*, Feb. 2023, pp. 1–6.
[69] C. D. G. Linhares, J. R. Ponciano, F. S. F. Pereira, L. E. C. Rocha, J. G. S. Paiva, and B. A. N. Travençolo, "Visual analysis for evaluation of community detection algorithms," *Multimedia Tools Appl.*, vol. 79, nos. 25–26, pp. 17645–17667, Jul. 2020.
[70] K. Varsha and K. K. Patil, "An overview of community detection algorithms in social networks," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, Feb. 2020, pp. 121–126.
[71] S. Sahu, "An approach for addressing internally-disconnected communities in Louvain algorithm," 2024, *arXiv:2402.11454*.
[72] *Kaggle*. Accessed: Jul. 31, 2023. [Online]. Available: https://www.kaggle.com/datasets?search=recommender+systems

SRILATHA TOKALA received the B.Tech. and M.Tech. degrees in computer science and engineering from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India, in 2013 and 2015, respectively. She is currently pursuing the Ph.D. degree in computer science and engineering with SRM University-AP, Amaravati, India. Her research interests include recommender systems, algorithms, and complex networks.

MURALI KRISHNA ENDURI (Member, IEEE) received the M.Sc. degree in mathematics from Acharya Nagarjuna University, Andhra Pradesh, India, in 2008, the M.Tech. degree in systems analysis and computer applications from the National Institute of Technology Karnataka, India, in 2011, and the Ph.D. degree in computer science and engineering from Indian Institute of Technology Gandhinagar, Gujarat, India, in 2018. He has research experience/postdoctoral research with Indian Institute of Technology Madras, Tamil Nadu, India, in 2018. He is currently an Assistant Professor with the Department of Computer Science and Engineering, SRM University-AP, Amaravati, India. His research interests include 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.

T. JAYA LAKSHMI (Member, IEEE) received the Ph.D. degree from the School of Computer and Information Sciences, University of Hyderabad, India, in 2019, with a focus on "Link Prediction in Heterogeneous Social Networks." She is currently an Assistant Professor with the Department of Computer Science and Engineering, SRM University-AP, Amaravati, Andhra Pradesh, India. She has an overall teaching experience of 22 years. Her research interests include graph mining, recommender systems, natural language processing, and security analytics. She is a reviewer of reputed international journals.

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

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 the Chairperson. He is currently the Section Head of technology foresight of the 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 is a Senior Editor of *Cogent Engineering*. He also served as a Reviewer for many famous academic journals, which are organized by ACM, Elsevier, IEEE, and Springer.