Decision Support Systems 118 (2019) 46–57

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

# A noise correction-based approach to support a recommender system in a highly sparse rating environment

Sujoy Bag[a], Susanta Kumar[a], Anjali Awasthi[b], Manoj Kumar Tiwari[a,*]

[a] Department of Industrial and Systems Engineering, Indian Institute of Technology Kharagpur, India
[b] Concordia Institute for Information Systems Engineering, Concordia University Montreal, Canada

ARTICLE INFO

*Keywords:*
Decision support systems
Recommender system
Collaborative filtering
Natural noise
Sparsity

ABSTRACT

Recommender systems support consumers in decision-making for selecting desired products or services from an overloaded search space. However, this decision support system faces difficulties while dealing with sparse and noisy rating data. Therefore, this research re-classifies users and items of a system into three classes, namely strong, average and weak to identify and correct noise ratings. Later, the Bhattacharya coefficient, a well-performing similarity measure for a sparse dataset, is integrated with the proposed re-classification method to predict unrated items from the obtained noise-free sparse dataset and recommend preferred products to consumers. Furthermore, the effectiveness of the proposed model is validated on two sparse and noisy datasets and compared with various published methods in terms of the mean absolute error (MAE), root mean square error (RMSE), F1-measure, precision, and recall values. The obtained results confirm that the proposed model performs better than other published relevant methods.

## 1. Introduction

Amazon, a widely known e-commerce company, has revealed that they sold more than 100 million products on their Prime Day, held on July 2018 [1]. Similarly, Flipkart, another popular Indian e-commerce company, sold one million units of smartphones in the very first hour of their big billion days, held on October 2018 [2]. This extensive growth of online shopping is catalyzed by its various aspects, such as preferred product recommendations, available feedback data, and trustworthiness of e-retailers etc. Positive reviews and ratings on online platforms help a seller attract more customers and boost sales [3]. According to Wallace's (2017) report, 96% of Americans have made an online purchase in their life, where 80%, 30%, and 5% have bought products as recently as the past month, week, and day, respectively [4]. In a typical e-commerce setting, customers receive an enormous amount of product-related information [5] and confuse to select suitable items. A recommender system helps visitors cope with information overload and providing them personalized services [6–8]. Users benefit from this system by minimizing online search cost, having appropriate purchase decision from optimal product evaluation and exploring new items. Online traders benefit by winning the trust and loyalty of customers, providing better-personalized services, retaining customers for a longer time, promoting their newly launched products, converting visitors to potential buyers and ultimately holding their position in online competitions and boosting their profits. Various organizations (i.e., Elsevier, LinkedIn, Netflix and YouTube) have implemented this system to provide worthy services to professional and digital societies.

Collaborative filtering, a widely used recommendation generation technique in both literature and industry, utilizes ratings of users to build user-user or item-item similarity metrics and identifies neighborhoods of users or items to generate recommendations [9]. However, malicious and non-malicious noises in the rating data [10] distort the performance of collaborative filtering. Malicious users intend to manipulate the collaborative filtering (CF) by shilling or profile injection attacks [11,12]. Various supervised and unsupervised learning approaches [13,14] have been reported in the literature to detect harmful attacks and improve the performance of collaborative filtering. Yang et al. analyzed statistical properties of diverse attack models to extract well-designed features from user profiles and applied the re-scaled AdaBoost [13] to improve the performance of the harmful attack detection. Furthermore, Zhang et al. proposed an unsupervised machine learning approach using hierarchical clustering and hidden Markov model [14] to detect shilling attacks properly.

In another situation, recommender systems have a misleading presumption that all genuine users provide an accurate rating on items. But, the inconsistent user behavior in a system generates another type of noise (natural noise) in a rating matrix [15–17]. Usually, this noise is caused when a user provides a rating absent-mindedly or has the

* Corresponding author at: Department of Industrial and Systems Engineering, Indian Institute of Technology, Kharagpur, West Bengal 721302, India.
*E-mail addresses:* sujoy.bag@iitkgp.ac.in (S. Bag), anjali.awasthi@concordia.ca (A. Awasthi), mktiwari9@iem.iitkgp.ernet.in (M.K. Tiwari).

https://doi.org/10.1016/j.dss.2019.01.001
Received 16 August 2018; Received in revised form 5 January 2019; Accepted 5 January 2019
Available online 11 January 2019

unwillingness to rate the item. Disruptive nature of natural noise in developing any statistical distribution hinders the identification of this noise by considering a particular attack strategy. Many researchers have proposed methods to remove the malicious noise in a recommender system model. However, non-malicious noise (i.e., natural noise) has been largely unnoticed. Only, a limited number of research studies related to natural noise have been conducted in the recommendation system domain [16]. To address the non-malicious noise, O'Mahony et al. removed the noisy ratings from the user-item rating matrix before applying the collaborative filtering [10] to predict ratings of unrated items. However, sparsity (a user-item rating matrix where less than or equals to 1% of the ratings are available) is another major challenge in recommendation generation. Thus, the approach of removing natural noise amplifies the issue of data sparsity. This sparse and biased rating-matrix generates an unreliable traditional similarity score, which forms poor recommendation generation. In this scenario, limited numbers of co-rated items are presented, and furthermore, in the case of natural noise, certain ratings are biased. To some extent, Toledo et al. were successful in identifying natural noise, and to correct it accurately, they applied a re-prediction approach using Pearson's correlation coefficient in a relatively denser dataset [18]. Later, Yera et al. incorporated a fuzzy model with the re-prediction method for managing the natural noise [17]. Furthermore, re-prediction, fuzzy profiling, global and local noise management approaches are performed in a group recommender system [15].

However, this re-prediction approach has only been tested in dense datasets. Moreover, in the sparse dataset, Pearson's correlation coefficient suffers from fewer co-rated problems and it shows low (high) similarity regardless of similar (difference) in ratings [19,20]. Hence, in the sparse data set, formations of Pearson's correlation coefficient and neighborhoods are inefficient and inaccurate, and hence, re-prediction will add to the problem of noisiness. Therefore, instead of re-prediction, the proposed method of noise correction is to re-classify users and items to weak, average and strong classes using the concept of self-contradiction nature of users and items. In addition to the correction of noisy ratings, the proposed model has used the noise-free data to predict ratings of unrated items effectively and has generated recommendations aptly. Moreover, in this research, the Bhattacharya coefficient, a recently developed similarity model that performs well in sparse datasets, is integrated with the proposed re-classification method to perform appropriate recommendation generation from noisy and sparse datasets. The key features of this study are two folded such as developing a unique re-classification method and building an integrated model with the help of Bhattacharya similarity model to solve the issues of natural noise and sparsity in the recommender system. The proposed model is validated based on two randomly generated datasets from MovieLens. Further, in this study, the results of the proposed method are compared with four relevant approaches. The results show that the proposed approach outperforms the recently published methods in terms of accuracy of prediction.

The remaining of this paper is organized as follows. The literature review of this study has been discussed in Section 2 and the proposed methodology is described in Section 3. Further, experimental setups and results of this study are presented in Section 4, and Section 5, respectively. Finally, the work is concluded in Section 6 with providing limitations and future research directions of the proposed study.

## 2. Literature review

In this digital era, the aptness of the recommender system motivates millions of consumers to go for online shopping and entertainment at a lower searching cost [21–23]. In addition to goods and multimedia recommendations, applications of this system have been noted in various business areas (i.e., e-government, e-health, e-tourism, e-library, e-resource services, and e-learning) [24,25]. Moreover, the recommendations of e-services fulfill the customer's satisfaction, which leads to benefiting different digital companies in terms of enlarging their sales, revenue, and loyalty [26]. Recently, e-commerce [27–29] has become the largest online marketplace for a variety of products, where recommendation techniques are used significantly. However, most of the recommender systems suffer major challenges from sparse and noise rating data.

This section presents various recommender system methods where researchers have addressed the issues of noise and sparsity. Moreover, the conceptual background of collaborative filtering is described along with two similarity metrics, such as the Pearson's correlation and Bhattacharya coefficient. Also, the literature gap and contribution of this study have been presented at the end of this section.

### 2.1. Natural noise in recommender systems

The major focus of a recommender system is to improve the prediction accuracy, while natural noise moderates the performance of the system. The ground assumption of the user-item rating matrix is often violated due to inconsistent user behavior in providing ratings on similar items. In this regard, Cosley et al. asked users to rerate 40 movies of the MovieLens dataset and noticed that 40% of users are inconsistent in their rating [30]. Similarly, Amatriain et al. instructed users to rerate items and compared the difference with a predetermined threshold value to decide whether the rating should keep or drop [31]. In the same scenario, Amatriain et al. observed that mild opinions are more inconsistent than extreme ratings [32]. According to O'Mahony et al. removing the natural noise leads to improving the performance of the recommendation system [10]. Thus, Li et al. proposed a real-time quadric optimization algorithm for the identification and removal of noisy but non-malicious users (NNMUs) [33]. However, dropping rating is an inappropriate method for handling sparse datasets as it increases the sparsity of the dataset. Thus, Toledo et al. considered the global rating of the user-item matrix and classified each user and item as being in weak, average and strong classes [18]. Their assumption of the natural noisy rating is that if both the user and item belong to the same class and the user makes a rating that does not belong to the same class, then the rating is to be treated as a probable noisy rating. The noisy ratings are then re-predicted using Pearson's correlation coefficient. Again, Pearson's correlation coefficient performs poorly while dealing with the sparse dataset. Therefore, this research has been proposed a re-classification approach instead of using the re-prediction method to correct the natural noise. Furthermore, the Bhattacharya coefficient instead of Pearson's correlation coefficient has been used to perform the recommender system well in noise and sparse datasets.

### 2.2. Sparsity of recommender systems

There is a huge number of users and items available in the online marketplace. However, only a limited number of items get rated by a limited number of users, which leads to a sparse user-item rating matrix [20]. Predicting a rating from a sparse rating matrix is a challenging task in recommender systems. Sparsity refers to a scenario in which most of the items are unrated, typically more than 99%. The sparsity of a dataset is calculated as.

$$Sparsity = 1 - \frac{R}{U * I} \tag{0.1}$$

where, $R$ is the total number of available ratings, $U$ is the total number of users in the system and $I$ is the total number of items.

Various techniques have been proposed in the literature to overcome this data sparsity issue. Luo et al. introduced a global similarity whose value depends on locally connected neighbors [34]. Further, Anand & Bharadwaj extended the work to automatize the parameter alpha depending on the sparsity of the entire dataset as well as the sparsity of the individual user and item (alpha is used to adjust the assigned weights to a global and local neighbor) [35]. In addition,

social trust performed well in the sparsity scenario to predict ratings of newly added users or items [36]. In an extension, Patra et al. used global similarity and a novel local similarity to address the issue of sparsity [20]. They used the Bhattacharya coefficient as a measure of global similarity, which utilizes all of the ratings available in the database to compute user-user or item-item similarities. Besides, multi-type auxiliary implicit feedback has been used to address the sparsity issue [37]. Moreover, adjective features embedded in user reviews are employed to resolve the problem of sparsity and transparency in recommender systems [38]. However, natural noise and sparsity issues have not been addressed in the literature together.

### 2.3. Collaborative filtering

The two most popular recommender system techniques are a content-based recommendation and collaborative filtering. In this research scenario, collaborative filtering is explored for further improvement so that the upgraded version of this algorithm can be easily applied and re-implemented in various application domains of the recommender system. Similarity metrics are the backbone of collaborative filtering that helps to find *k-nearest neighbors* of a targeted user and predict ratings of unrated items. Hence, this study considers two most convenient similarity metrics, namely, the Pearson's correlation and Bhattacharya coefficient, for handling sparse ratings in the recommender system.

#### 2.3.1. Pearson's correlation similarity: The Pearson's correlation coefficient represents the linear relationship between two estimated vectors [39]

$$Sim(u,v) = \frac{\sum_{i\in I(u,v)} (R(u,i) - \overline{R(u)})\cdot(R(v,i) - \overline{R(v)})}{\sqrt{\sum_{i\in I(u,v)} (R(u,i) - \overline{R(u)})^2}\cdot\sqrt{\sum_{i\in I(u,v)} (R(v,i) - \overline{R(v)})^2}} \quad (0.2)$$

where $R(u,i)$ is the rating of item $i$ by user $u$, $\overline{R(u)}$ is the average rating of user $u$ for all of the co-rated items, and $I(u,v)$ represents the number of co-rated items by both users $u$ and $v$.

#### 2.3.2. Bhattacharya coefficient (BC) for the global similarity

The Bhattacharya coefficient (BC) is one of the most popular and widely used similarity measures to retrieve value-added information from sparse datasets. Patra et al. have introduced the BC as a measure of global similarity between the item pairs in recommender systems [20]. As is known, a histogram is a diagram that is plotted between the frequencies of certain occurrences vs. the bins of different frequency counts. Basically, BC measures the similarity between two histograms. In the recommender system scenario, items $i$ and $j$ are considered as histograms to calculate the BC for global similarity as,

$$Sim(i,j)_{BC} = BC(i,j) = BC(p_i', p_j') = \sum_{h=1}^{m} \sqrt{(p_{ih}')*(p_{jh}')} \quad (0.3)$$

where $p_{ih}' = \frac{\#h}{\#i}$, m is the number of bins, $\#i$ is the number of users that rated item $I$, and $\#h$ is the number of users that rated item $I$ with the rating value of $h$. Let us consider the following example of the user-item rating matrix in Table 1.

Here, item 1 has obtained a total of 6 ratings. Of these, one rating is 1, three ratings are 3, one rating is 4, and one rating is 5. Item 2 has also obtained a total of 6 ratings, and the frequency of the ratings at each discrete level is the same as for item 1.

Hence, the BC between $I_1$ and $I_2$ is calculated as,

$$BC(I_1,I_2) = \sqrt{\frac{1}{6}*\frac{1}{6}} + \sqrt{\frac{3}{6}*\frac{3}{6}} + \sqrt{\frac{1}{6}*\frac{1}{6}} + \sqrt{\frac{1}{6}*\frac{1}{6}} = 1 \quad (0.4)$$

Similarly, the BC between $I_1$ and $I_3$ is computed as,

$$BC(I_1,I_3) = \sqrt{\frac{1}{6}*\frac{1}{4}} + \sqrt{\frac{3}{6}*\frac{1}{4}} + \sqrt{\frac{1}{6}*\frac{1}{4}} + \sqrt{\frac{1}{6}*\frac{1}{4}} = 0.966 \quad (0.5)$$

Bhattacharya similarity between two users can also be calculated in the same way.

### 2.4. Literature gap and contribution

Recently published relevant research papers are presented in Table 2. A limited number of study have been performed to address the natural noise [18]. Toledo et al. have identified noisy ratings and proposed a re-prediction approach using Pearson's correlation coefficient to replace the noisy ratings with the predicted ratings [18]. Furthermore, re-prediction method is incorporated with the fuzzy model and also performed in group recommender systems [15,40]. However, existing models of re-prediction approach are tested only in dense datasets. Moreover, Pearson's correlation coefficient suffers in case of fewer co-rated items and shows low (high) similarity regardless of similar (difference) in the ratings [20]. Thus, Pearson's correlation coefficient measures inaccurate similarity value and identifies insignificant neighborhoods in the sparse data set. Hence, re-prediction doesn't create an optimal solution for correcting natural noise in sparse datasets. Patra et al. proposed a Bhattacharya coefficient to address the issue of sparsity [20], but the natural noise is not considered in their model. Therefore, in this study, ratings of natural noise are identified [18] and further proposed a re-classification method instead of using the re-prediction approach to correct the natural noise. Furthermore, the Bhattacharya coefficient [20] is integrated with the proposed re-classification method to perform the recommender system efficiently even in sparse and noisy datasets.

**Table 1**
User-item rating matrix.

| | $U_1$ | $U_2$ | $U_3$ | $U_4$ | $U_5$ | $U_6$ | $U_7$ | $U_8$ | $U_9$ | $U_{10}$ | $U_{11}$ | $U_{12}$ | $U_{13}$ | $U_{14}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $I_1$ | 3 | 0 | 1 | 0 | 3 | 0 | 0 | 5 | 0 | 3 | 0 | 0 | 0 | 4 |
| $I_2$ | 0 | 3 | 0 | 3 | 0 | 0 | 0 | 0 | 1 | 0 | 5 | 3 | 4 | 0 |
| $I_3$ | 0 | 3 | 1 | 0 | 0 | 4 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 2**
Recently published relevant research papers.

| Authors | Problem | Work |
|---|---|---|
| (Toledo et al., 2015) [18] | Natural noise | Identify natural noise and correct it by using a re-prediction approach in a relatively denser dataset |
| (Yera et al., 2016) [17] | Natural noise | A fuzzy model has been incorporated with re-prediction method |
| (Castro et al., 2017; Castro et al., 2018) [15,40] | Natural noise | Fuzzy profiling, re-prediction, global and local noise management methods are implemented in group recommender system |
| (Patra et al., 2015) [20] | Data sparsity | Develop the Bhattacharya coefficient to address the issue of sparsity |

**Table 3**
Criteria to assign users to different classes.

| Classes for users | Condition |
|---|---|
| Weak user ($U_w$) | $\|U_w\| \geq \|U_a\| + \|U_s\|$ |
| Average user ($U_a$) | $\|U_a\| \geq \|U_w\| + \|U_s\|$ |
| Strong user ($U_s$) | $\|U_s\| \geq \|U_a\| + \|U_w\|$ |
| Variable user ($U_v$) | When the above three conditions are unsatisfied |

**Table 4**
Criteria to assign items to different classes.

| Classes for items | Condition |
|---|---|
| Weak item ($I_w$) | $\|I_w\| \geq \|I_a\| + \|I_s\|$ |
| Average item ($I_a$) | $\|I_a\| \geq \|I_w\| + \|I_s\|$ |
| Strong item ($I_s$) | $\|I_s\| \geq \|I_a\| + \|I_w\|$ |
| Variable item ($I_v$) | When the above three conditions are unsatisfied |

## 3. Methodology

### 3.1. Noise identification and correction

The first step of the proposed approach is to identify the natural noise in the system and correct it. To do so, users and items are classified into weak, average, strong and uncertain classes based on conditions presented in Tables 3 and 4. For the process of classification, Toledo et al. set a threshold to segregate weak, average and strong ratings [18]. In addition, they introduced four parameters, namely, $k_u$, $v_u$, $k_i$, *and* $v_i$, where $k_u$ and $v_u$ are the threshold boundaries of user-classes between weak-average and average-strong, respectively. Suppose that if a user rating is less than $k_u$, it is added to the set of weak class, and if it is greater than or equal to $k_u$ but less than $v_u$, then the rating is considered to be an average rating. Similarly, if a rating is greater than or equal to $v_u$, then the rating is considered to be a strong rating. In this research, the threshold values of $k_u$ and $v_u$ are pre-determined fixed thresholds (i.e. 3 and 4) for user-item classifications of all users and items. This study is performed on the rating scale of 1 to 5 where the rating value of 1 and 2 are considered as weak ratings, rating 3 is average rating and the ratings of 4 and 5 are strong ratings. The cardinality of total ratings of a user is taken into consideration when classifying the user into a particular class. If the cardinality of weak ratings of the user exceeds his average and strong ratings together, then his ratings are negatively skewed in the discrete rating scale to the lower ends, and as a result, the user is considered to be a weak user $U_w$. If the cardinality of average ratings exceeds the cardinality of his strong and weak ratings, then the user is considered to be an average user $U_a$, because his ratings are symmetric and dense around the mean. Similarly, items are also classified into weak, average and strong classes depending on their ratings and threshold boundaries $k_i$ (the weak-average threshold for items) and $v_i$ (the average-strong threshold for items).

The classification procedures of users and items are conspicuously presented in Algorithms 1 and 2 [18]. Initially, users and items are separated based on the threshold boundaries of weak-average and average-strong classes. Then, the cardinality of rating metrics is used to classify ratings in weak, average, strong and variable classes. Finally, the classified noise ratings are corrected based on the proposed method. The detailed procedure of the noise rating correction is presented in Table 5 and Algorithm 3.

**Algorithm 1.** Classify the user's classes of a highly sparse and noisy dataset.

***Input :*** *User - item rating metric of a sparse and noise dataset*
***Output :*** *Set of classified user's classes*
***1 :*** $U_W = (\ ), U_A = (\ ), U_S = (\ ), U_V = (\ )$
***2 : for*** *each user u* ***do***
***3 : for*** *each item i* ***do***
/* $k_u$ *and* $v_u$ *are the threshold boundaries of* user's *classes between weak – average and average – strong respectively* */
***4 : if*** $r(u,i) < k_u$ ***then***
***5 :*** *add* $r(u,i)$ *to the set of* $U_W$
***6 : else if*** $r(u,i) \geq k_u$ *and* $r(u,i) < v_u$ ***then***
***7 :*** *add* $r(u,i)$ *to the set of* $U_A$
***8 : else***
***9 :*** *add* $r(u,i)$ *to the set of* $U_S$
***10 : end if***
***11 : end for***
***12 : end for***
***13 : for*** *each user u* ***do***
***14 : if*** $\|U_W\| \geq \|U_A\| + \|U_S\|$ ***then***
***15 :*** $U_W$ *belongs to weak user's class*
***16 : else if*** $\|U_A\| \geq \|U_W\| + \|U_S\|$ ***then***
***17 :*** $U_A$ *belongs to average user's class*
***18 : else if*** $\|U_S\| \geq \|U_A\| + \|U_W\|$ ***then***
***19 :*** $U_S$ *belongs to strong user's class*
***20 : else***
***21 :*** $U_V$ *belongs to variable user's class*
***22 : end if***
***23 : end for***

**Algorithm 2.** Classify the item's classes of a highly sparse and noisy dataset.

***Input :*** *User - item rating metric of a sparse and noise dataset*
***Output :*** *Set of classified items - classes*
***1 :*** $I_W = (\ ), I_A = (\ ), I_S = (\ ), I_V = (\ )$
***3 : for*** *each item i* ***do***
***2 : for*** *each user u* ***do***
/* $k_i$ *and* $v_i$ *are the threshold boundaries of* item's *classes between weak – average and average – strong respectively* */
***4 : if*** $r(u,i) < k_i$ ***then***
***5 :*** *add* $r(u,i)$ *to the set of* $I_W$
***6 : else if*** $r(u,i) \geq k_i$ *and* $r(u,i) < v_i$ ***then***
***7 :*** *add* $r(u,i)$ *to the set of* $I_A$
***8 : else***
***9 :*** *add* $r(u,i)$ *to the set of* $I_S$
***10 : end if***
***11 : end for***
***12 : end for***
***13 : for*** *each user i* ***do***
***14 : if*** $\|I_W\| \geq \|I_A\| + \|I_S\|$ ***then***
***15 :*** $U_W$ *belongs to weak item's class*
***16 : else if*** $\|I_A\| \geq \|I_W\| + \|I_S\|$ ***then***
***17 :*** $U_A$ *belongs to average item's class*
***18 : else if*** $\|I_S\| \geq \|I_A\| + \|I_W\|$ ***then***
***19 :*** $I_S$ *belongs to strong item's class*
***20 : else***
***21 :*** $I_V$ *belongs to variable item's class*
***22 : end if***
***23 : end for***

**Table 5**
Proposed approach to correcting the noise.

| | $I_w$ | $I_a$ | $I_s$ |
|---|---|---|---|
| $U_w$ | Modify user rating with $k$ | No modification | No modification |
| $U_a$ | No modification | Modify user rating with $(k + v)/2$ | No modification |
| $U_s$ | No modification | No modification | Modify user rating with $v$ |

**Algorithm 3.** Correct the natural noise of a highly sparse dataset.

***Input :*** *User - item rating matrix of a sparse dataset including natural noise, classified user - item rating matrix*
***Output :*** *Updated rating matrix after correcting the natural noise*
***1 :*** $U_w = (\ ), I_w = (\ ), U_a = (\ ), I_a = (\ ), U_s = (\ ), I_s = (\ )$
***2 : for*** *each rating r(u, i)* ***do***
***3 :*** *if both u and i belongs to weak class & r(u, i) ≥ k* ***then***
***4 :*** replace r(u,i) with k
***5 :*** ***else if*** *both u and i belongs to average class, and r(u, i) < k or r(u, i) ≥ v* ***then***
***6 :*** replace r(u,i) with (k+u)/2
***7 :*** ***else if*** *both u and i belongs to strong class and r(u, i) < v* ***then***
***8 :*** replace r(u,i) with v
***9 :*** ***else***
***10 :*** *unchanged*
***11 :*** ***end if***
***12 :*** ***end for***

#### 3.1.1. Illustration with examples

First, a user-item rating matrix is considered in Table 6. The user $U_1$ has given high ratings on three items and low ratings on two items. The cardinality of the user's strong rating is higher than the cardinalities of the user's low and neutral rating. Hence, the user is considered to be a strong user. Similarly, for the item $I_1$, the cardinality of the item's weak rating is higher than the cardinalities of the item's neutral and low rating. Hence, the item is classified as a weak item. The classifications of all users and items are shown in Table 6. If both the user and item belong to a particular class and the user makes a self-contradiction by providing a rating in some other class, in that case, the noise rating is modified with a rating that belongs to a user-item class. For example, if both the user and item belong to the strong class and the user rating is very low, then the rating is modified with the strong class rating (i.e., if a user belongs to the strong class and gives the rating 5/5 for Interstellar movie, 5/5 for Fight Club, and 4/5 for The Dark Knight, but the user unwillingly gives a rating of 1/5 for The Matrix (which is a highly rated movie, and both the user and movie belong to the same class - strong)). If a user provides such a type of class rating, then the rating is treated as noise and is modified to the prior rating of the same class.

In the user-item rating matrix presented in Table 6, $(U_3, I_1)$ belongs to the weak class, but the rating given to it belongs to a strong class, which is a class contradiction by the user. Hence, the rating is modified to a weak class and, thus, assigned a score of 1 to replace the 5. In another instance, $(U_1, I_2)$ belongs to the strong class, but the rating assigned to it belongs to the weak class, and therefore, the rating value of 1 is replaced with the rating value of 4.

**Table 6**
Assigning users and items for different classes.

| | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ | $I_6$ | | W | A | S | Class | | W | A | S | Class |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $U_1$ | 2 | 1 | 5 | 0 | 4 | 5 | $U_1$ | 2 | 0 | 3 | Strong | $I_1$ | 5 | 0 | 1 | Weak |
| $U_2$ | 1 | 4 | 3 | 3 | 0 | 3 | $U_2$ | 1 | 3 | 1 | Average | $I_2$ | 1 | 0 | 5 | Strong |
| $U_3$ | 5 | 0 | 1 | 1 | 3 | 1 | $U_3$ | 3 | 1 | 1 | Weak | $I_3$ | 2 | 3 | 1 | Average |
| $U_4$ | 1 | 4 | 1 | 0 | 2 | 1 | $U_4$ | 4 | 0 | 1 | Weak | $I_4$ | 1 | 2 | 0 | Average |
| $U_5$ | 1 | 5 | 3 | 3 | 3 | 3 | $U_5$ | 1 | 4 | 1 | Average | $I_5$ | 1 | 2 | 1 | Average |
| $U_6$ | 2 | 5 | 3 | 0 | 0 | 0 | $U_6$ | 1 | 1 | 1 | Uncertain | $I_6$ | 2 | 2 | 1 | Uncertain |

### 3.2. Framework of the proposed study

Bhattacharya coefficient (BC) is a well-performed similarity measure for sparse datasets, whereas, the proposed re-classification method is developed to correct the natural noise. Therefore, the Bhattacharya coefficient (BC) similarity is integrated in this research with the noise identification and the proposed noise correction method to solve both problems of natural noise and sparsity in the recommender system together. The flow chart of the proposed noise corrected BC method is presented in Fig. 1(a). The published approaches of re-predicted PC and noise BC are shown in Fig. 1(b) and (c), respectively. Further to identify the best alternative approach for noisy and sparse datasets, the proposed noise corrected BC and existing methods are mixed to generate another two methods. The mixed methods of noise corrected PC, and re-predicted BC have been presented in Fig. 1(d) and (e), respectively. Thereafter, the rating prediction of unrated items using the proposed noise corrected BC method is presented in Algorithm 4.

**Algorithm 4.** Rating prediction of unrated items using the proposed noise corrected BC method.

***Input :*** *Noise corrected user - item rating matrix*
***Output :*** *Value of predicted ratings*
***1 :*** $R^*(u,i) = [\ ]$
***## Prediction of un - rated items using user - based KNN CF algorithm***
***2 : for*** *the user u = 1 to |U|* ***do***
***3 :*** ***for*** *the item i = 1 to |I|* ***do***
***4 :*** ***if*** $R(u,i) == 0$ ***then***
***5 :*** $$R^*(u,i) = \overline{R(u)} + \frac{\sum_{v \in \overline{NN}_k} Sim(u,v)_{BC} \cdot \left(R(v,i) - \overline{R(v)}\right)}{\sum_{v \in \overline{NN}_k} |Sim(u,v)_{BC}|}$$
***6 :*** ***end if***
***7 :*** ***end for***
***8 :*** ***end for***

*# Whereas, $R^*(u,i)$ is the predicted rating of the targeted user u for item i*
*# $\overline{R(u)}$ is the mean rating of user u.*
*# $NN_k$ is the k numbers of other top similar users who have also rated item i.*
*# $R(v,i)$ is rating of the nearest neighbor v for item i.*
*# $\overline{R(v)}$ is the mean rating of the nearest neighbor v of the user u.*
*# $Sim(u,v)_{BC}$ is the Bhattachrya coefficient (BC) similarity between the user u and its nearest neighbor v.*

## 4. Experimental setup

This section explains the experimental setup including details of the used dataset and evaluation metrics.

### 4.1. Datasets

In this research, MovieLens, a standard dataset in recommendation system research, is collected from the GroupLens data store to validate the proposed model [41]. The dataset consists of 943 users and 1682 items and 1 million ratings in the discrete scale of [1–5], where a rating of 1 indicates the user dislikes the movie, a rating of 2 means the user weakly prefers the movie, a rating of 3 indicates that the user is neutral about the movie, and a rating of 4 means that the user strongly prefers the movie and 5 means the user very strongly prefers the movie. The percentage of available ratings in the original dataset is approximately 6.3, which is quite dense. As the proposed model is required to test on different levels of highly sparse datasets, thus, two sample data sets are randomly generated with considering 300 random users and 400

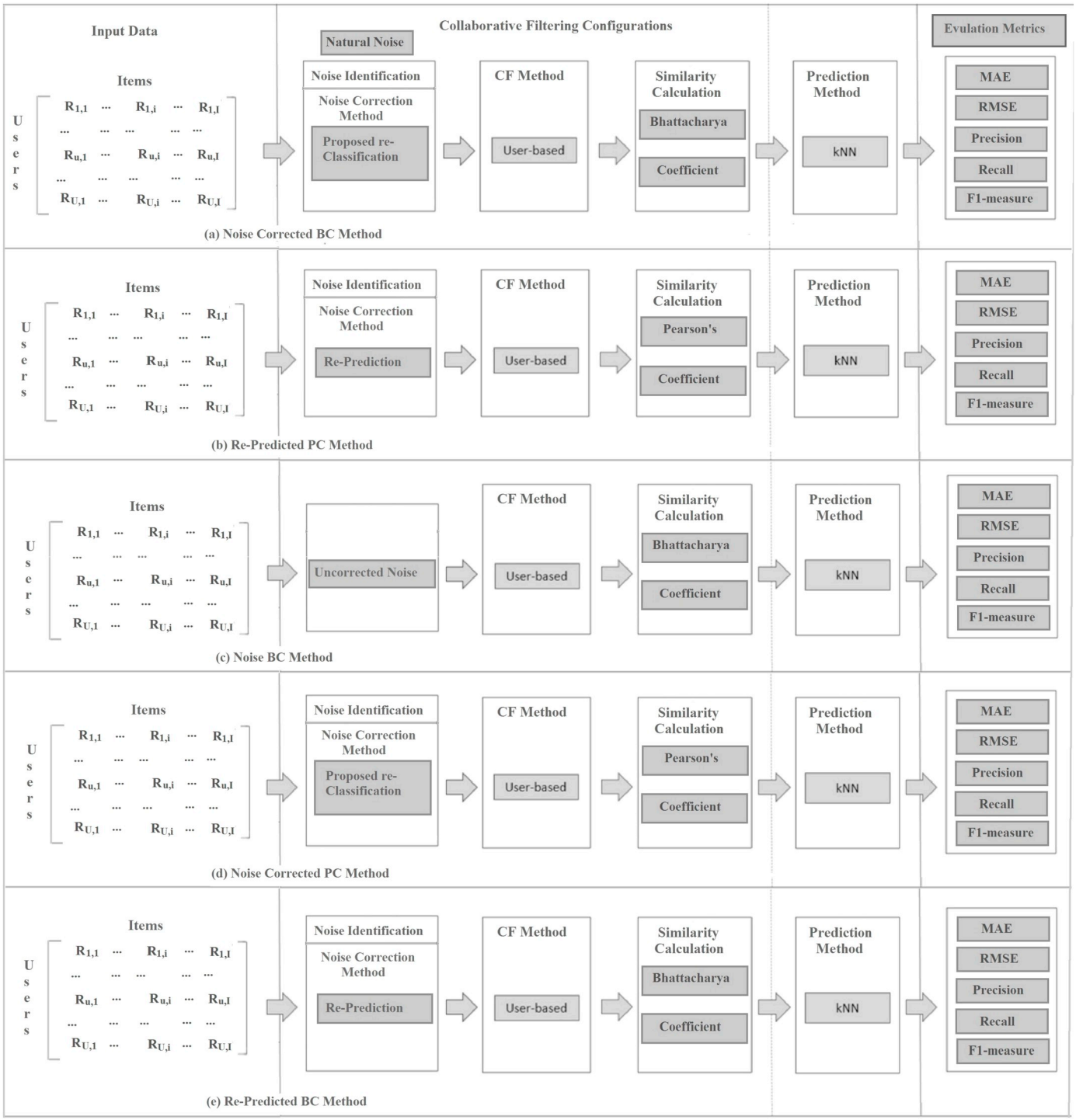


**Fig. 1.** Flowchart of the proposed and published relevant approaches.

random movies from the MovieLens dataset. Initially, the number of available ratings for both generated dataset-1 and dataset-2 are same (i.e. 8551), where the sparsity is approximately 0.9287. Further, different percentages of available ratings are randomly altered to zero to form two unique and sparse datasets.

In dataset-1, 90% of the available ratings are altered to zero randomly, and thus the sparsity of the dataset becomes 0.9929 where the total number of available ratings are 855. Then, the dataset is split into 70%–30% for the purpose of training and testing respectively. Further, 85% of the available ratings are replaced with zero in data set-2 to reach a sparsity of approximately 0.9893 (total available ratings are 1283), and then, the dataset is divided into 80%–20% for the purpose of training and testing. In this analysis, 10% and 15% of rating data of original dataset are chosen as because of testing the performance consistency for all approaches in removing 5% rating data in sparse dataset. The entire work is performed in R-studio 4.2.3.

### 4.2. Evaluation metrics

The performance of the proposed algorithm is evaluated by both 1) the predictive accuracy and 2) the classification accuracy. The mean absolute error (MAE) and root mean square error (RMSE) are

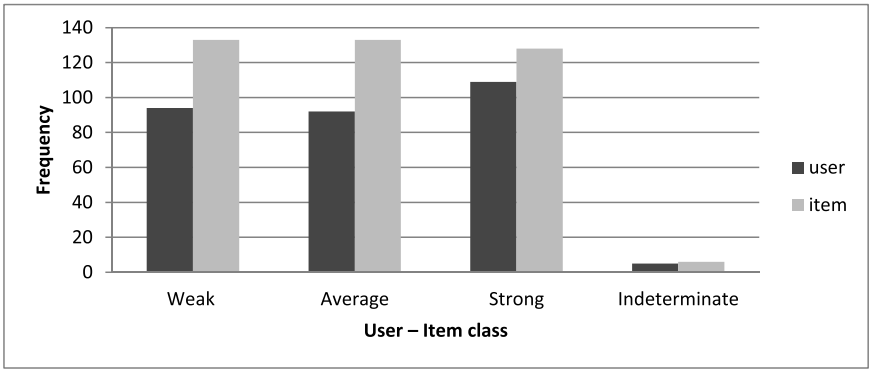


**Fig. 2.** Classification of users and items to different classes for dataset-1.

commonly used to evaluate the predictive accuracy, while the Precision, Recall, and F1-measure are used to measure the classification accuracy [20]. In the computation of MAE, the first absolute sum of the difference between the actual rating and predicted rating is calculated over all of the predicted ratings, and then, it is divided by a total number of predicted ratings. Hence, a smaller value of MAE indicates a better accuracy of prediction.

$$MAE = \frac{\sum_{i=1}^{n} |r_i - p_i|}{n} \tag{0.6}$$

The mean square error (MSE) is calculated by dividing the sum of squares of the differences of the actual and predicted ratings by the total number of ratings on which the predictions are made. The RMSE is obtained by taking the square root of the MSE.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (r_i - p_i)^2}{n}} \tag{0.7}$$

On the other hand, precision and recall values are calculated based on four possible indicators of predicted rating, namely true positive (TP), false positive (FP), true negative (TN), and false negative (FN) [25,42].

$$\textbf{\textit{Precision}} = \frac{TP}{TP + FP} \tag{0.8}$$

$$\textbf{\textit{Recall}} = \frac{FN}{TP + FP} \tag{0.9}$$

With an increase in the number of neighbors, the precision value decreases, while the recall value increases. Hence, the harmonic mean of both the precision and recall is accounted for evaluating the F1 measure. A higher F1 value signifies a high quality of prediction.

$$F1\,measure = \frac{precision + recall}{2 \times precision \times recall} \tag{0.10}$$

## 5. Results

This section shows rating frequencies of different classification classes and the number of modified ratings for applying the proposed re-classification method. Furthermore, the performance of the proposed algorithm is compared with relevant approaches and presented in the form of various accuracy metrics.

### 5.1. Classification

Initially, users and items are classified into weak, average, strong and indeterminate classes to identify noisy ratings, and then, noise ratings are corrected based on the proposed model.

#### 5.1.1. User-item classification

Users and items of dataset-1 and dataset-2 are classified based on the algorithm provided by Toledo et al. [18] and presented in Figs. 2 and 3, respectively. In dataset-1, 94 users are identified as weak users, 92 users as average users and 109 users as strong users. Similarly, 133 items are classified as weak, 133 items as average and 128 items as strong items. A total of 5 users and 6 items are unclassified into any classes. Similarly, 121 weak, 87 average and 89 strong users are presented in dataset-2. The classified items of weak, average and strong are 163, 129 and 106, respectively. However, 3 users and 2 items are unclassified into any classes.

#### 5.1.2. User-item re-classification

Corrected ratings of users using the proposed rating correction

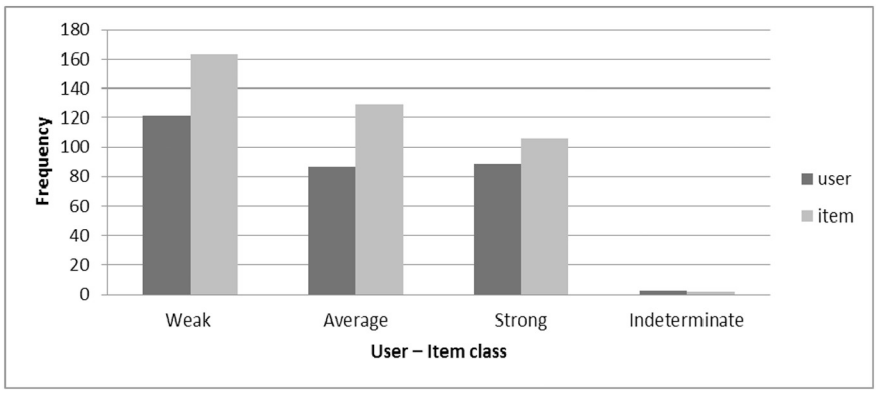


**Fig. 3.** Classification of users and items to different classes for dataset-2.


**Fig. 4.** Corrected ratings of users using the proposed rating correction method for dataset-1.

method for dataset-1 and dataset-2 are presented in Figs. 4 and 5, respectively. In the noise identification process, 61 ratings are identified as a noisy rating in dataset-1 out of 598 available ratings (i.e., 10.2% of the ratings are noisy). In the re-classification time, ratings are modified to be 9 ratings from average to strong, 3 ratings from weak to strong, 19 ratings from strong to average and 30 ratings from weak to average, using the noise modification algorithm. Similarly, in dataset-2, 88 ratings have been identified as a noisy rating out of 897 ratings (i.e., 9.8% of the ratings are natural noise ratings). In this case, 24 ratings are modified from average to strong, 22 from strong to average, and 42 ratings from weak to average.

### 5.2. Performance of the recommender system

After re-classifying users and items, the proposed method is identified and corrected inconsistency rating patterns of users and items, and then performed the KNN algorithm using Bhattacharya's coefficient for predicting unrated items. The proposed noise corrected Bhattacharya's coefficient (BC) method is validated based on two randomly generated datasets. The obtained results from the proposed method is compared with four relevant approaches. It is hard to perform collaborative filtering in a small number of nearest neighbors for the sparse dataset. Thus, the rating prediction in this study has been performed based on nearest neighbors 20 to 100.

#### 5.2.1. Results for dataset-1

Results of this study in the form of evaluation metrics of precision, recall, and F1-measure are presented in Figs. 6, 7, and 8, respectively, for dataset-1. It is clearly visible that in the sparse scenario, the proposed noise correction method performs better than the noise re-prediction approach. Moreover, the results of the proposed noise corrected BC leads all of the existing state-of-the-art methods (i.e. noise BC and


**Fig. 5.** Corrected ratings of users using the proposed rating correction method for dataset-2.

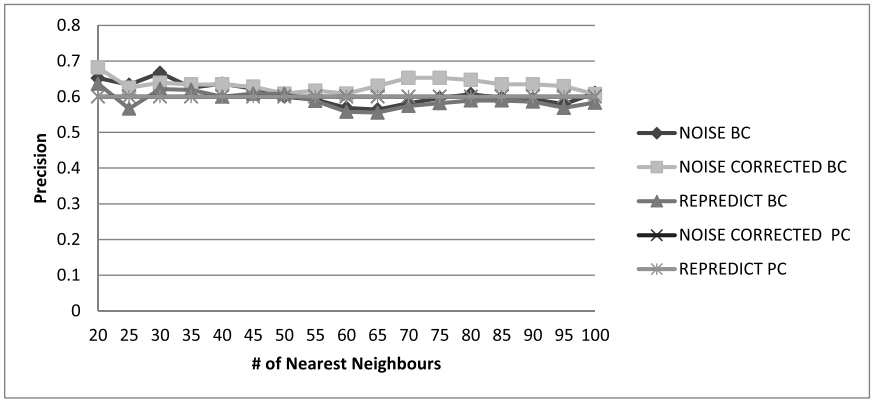


**Fig. 6.** Precision value comparison of the different methods for dataset-1.

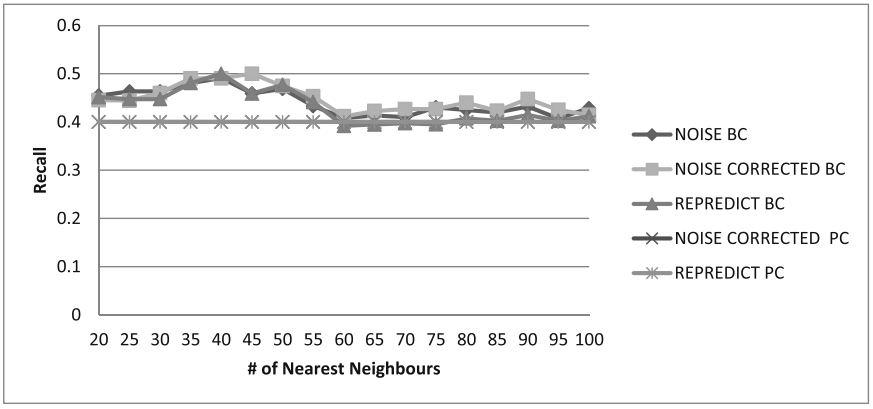


**Fig. 7.** Recall value comparison of the different methods for dataset-1.

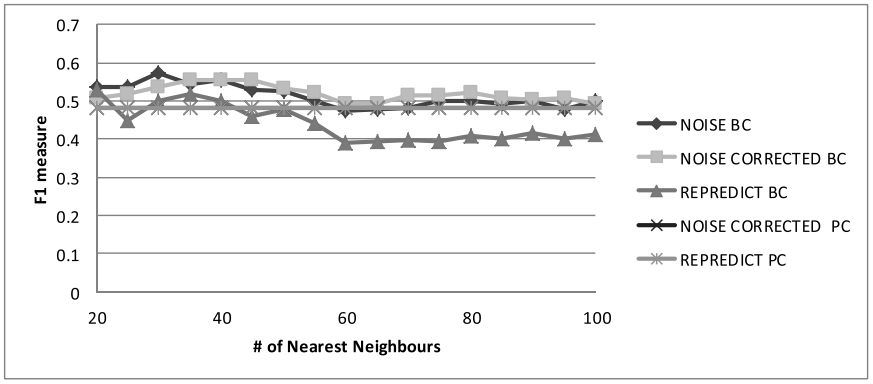


**Fig. 8.** F1 value comparison of the different methods for dataset-1.

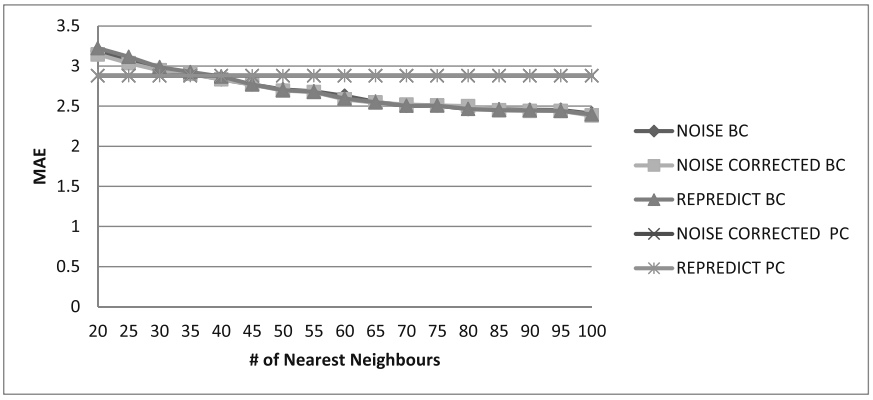


**Fig. 9.** MAE value comparison of the different methods for dataset-1.

re-predict PC) and mixed approaches (i.e. re-predict BC and noise corrected PC).

Furthermore, the results in the form of the accuracy metrics MAE and RMSE are presented in Figs. 9 and 10 to validate the correctness of the prediction model. The results show that BC has a lower value of MAE and RMSE (hence a better predictive accuracy) than PC in all of the three scenarios – noisy, noise correction, and noise re-prediction.

*5.2.2. Results for dataset-2*

Similar to dataset-1, results of the proposed model for dataset-2 in the form of precision, recall, and F1-measure are compared with relevant methods and shown in Figs. 11, 12, and 13 respectively. It can be clearly noted that the proposed noise corrected BC method outperformed all of the others existing and mixed approaches.

Because a linear pattern in the MAE and RMSE value has been observed while analyzing the accuracy of the proposed models in dataset-1, the computation of the MAE and RMSE values are overlooked for dataset-2.

## 6. Conclusions and future extensions

The basic objective of this work is to improve the efficiency of recommendation generation in noisy and sparse scenarios. Dataset-1 contains approximately 10.2% natural noise and sparsity of 0.004 ratings, while dataset-2 contains approximately 9.8% natural noise and sparsity of approximately 0.008 ratings. Thus, correcting the noisy rating is a crucial pre-processing step before starting any other recommendation generation tasks, such as building a similarity matrix,

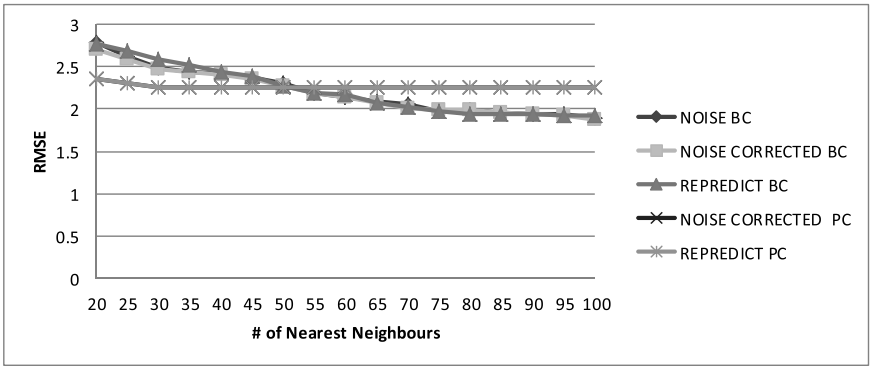


**Fig. 10.** RMSE value comparison of the different methods for dataset-1.

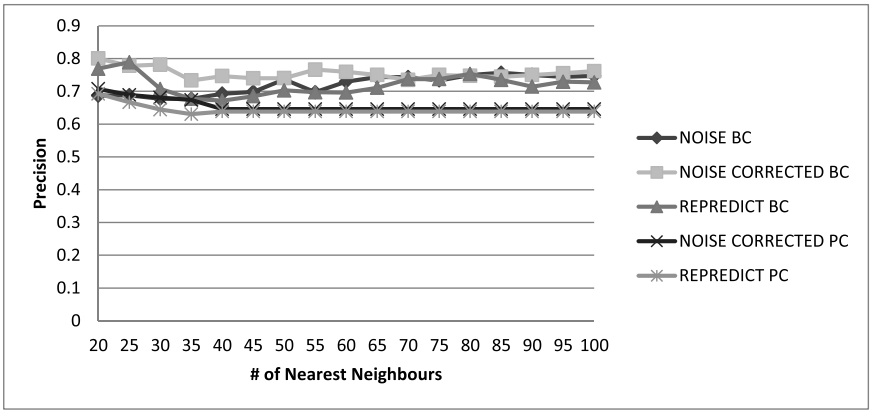


**Fig. 11.** Precision value comparison of the different methods for dataset-2.

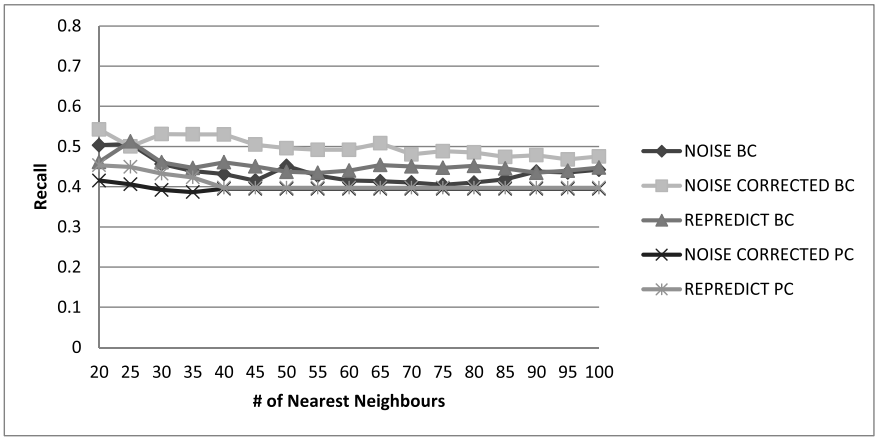


**Fig. 12.** Recall value comparison of the different methods for dataset-2.

forming nearest neighbors or rating predictions of the unrated items. Moreover, sparsity is another major issue in recent recommender systems. In the literature, the issues of sparsity and natural noise are handled separately by using the Bhattacharya coefficient and re-prediction of noisy ratings with Pearson's coefficient, respectively. Additionally, the approach of re-prediction of noisy ratings yields very poor predictions for sparse datasets. Therefore, in this study, first the users and items are classified into weak, average and strong classes, and then, the natural noise is identified and corrected in the dataset based on the proposed user-item rating re-classification method. In addition, as BC forms effective similarity metrics in a sparse rating matrix, it is thus further applied on a noise-free rating matrix to address the deficiencies of the traditional similarity measures in sparse datasets. The experimental results show that the proposed method improves the performance of the recommendation system in terms of having better precision, recall and F1 values and lower MAE and RMSE values. Moreover, the proposed method is advance in terms of the time complexity because it dropped the step of re-prediction. This study is helpful for mitigating bias and irrelevant recommendation generation in the recent scenario of sparse and natural noise rating metrics. Apart from goods and multimedia recommendations, the proposed model is applicable to various business and administrative areas of e-government, e-health, e-tourism, e-library, e-resource services, and e-learning etc.

Although, this research has a significant contribution in this digital era, still, there are some limitations in the proposed model. Thus, in the

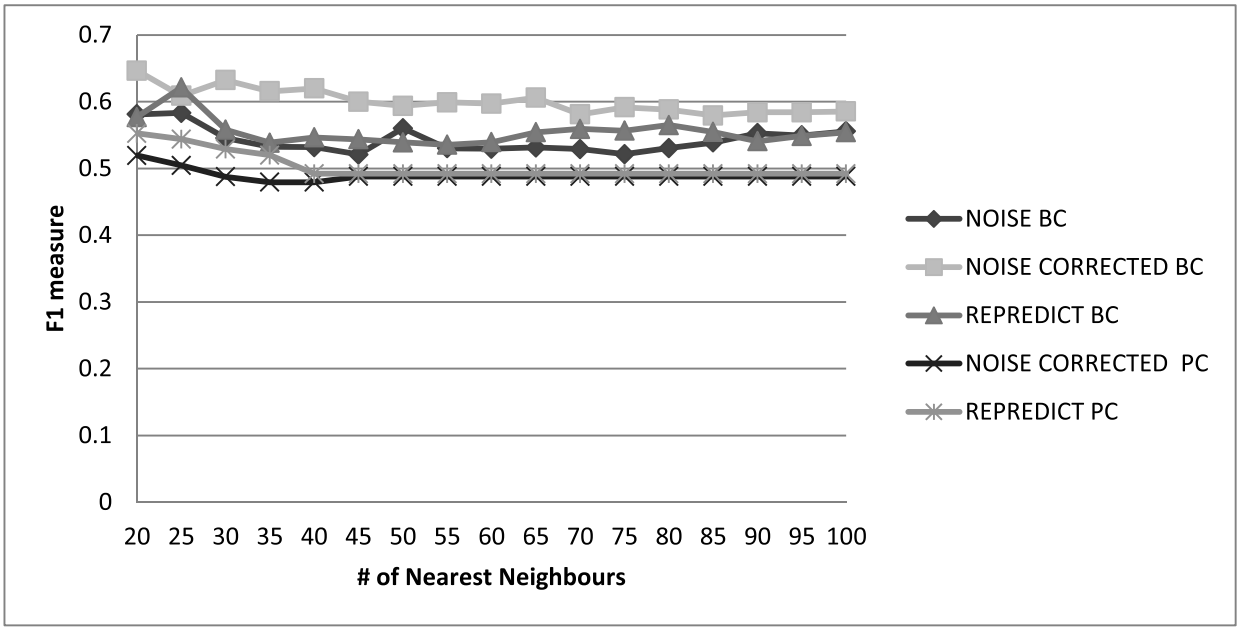


**Fig. 13.** F1 value comparison of the different methods for dataset-2.

future, this work can be extended in many directions. The proposed model is used for a pre-determined and fixed threshold for user-item classifications of all users and items. In the future, the statistical distribution of ratings for all users and items can be considered to set different threshold boundaries for different users and items. Further, the proposed model is implemented only on the global similarity of BC, which has quadratic time complexity; however, it ignores the local similarity of BC because it has a high time complexity of $O(n^4)$. In the future, an efficient local similarity can be designed to improve the efficiency and accuracy of the prediction model. Due to the limitations of high computing resources, a pilot study has been performed in this research by only considering two datasets of 300 users and 400 items. In the future, big data analysis can be involved or parallel processing to test the proposed model for big datasets. Specifically, Apache Spark, an open-source cluster computing framework, can be used for the proposed model to handle the huge amount of user-generated ratings. Moreover, a high-performance extreme learning machine or quantum machine learning can be applied instead of a traditional recommendation system prediction model to handle big datasets. In this study, only numeric rating values are considered for improving the efficiency of the recommender system. In the future, various side information of online consumers such as reviews data, search pattern, eye movement etc. can be utilized in the recommender system model to improve its performance. Moreover, aggregate diversity is an important factor for analyzing the performance of a recommender system. However, in this study, only, various accuracy metrics are considered and improved for better recommendation. In the future, aggregate diversity can be considered and improved for noisy and sparse datasets.

## Acknowledgments

The work has been supported by the project of E-Business Center of Excellence and funded by Ministry of Human Resource and Development (MHRD), Government of India under the scheme of Center for Training and Research in Frontier Areas of Science and Technology (FAST), Grant No. F.No.5-5/2014-TS.VII.

## References

[1] K. Holdefehr, This was the best-selling product on Amazon prime day 2018, https://www.realsimple.com/home-organizing/amazon-prime-day-2018, (2018) , Accessed date: 30 December 2018.
[2] M. Variyar, V. Bansal, One million smartphones sold on Flipkart in first hour of sales, https://economictimes.indiatimes.com/small-biz/startups/newsbuzz/one-million-smartphones-sold-on-flipkart-in-first-hour-of-sales/articleshow/66172971.cms, (2018) , Accessed date: 30 December 2018.
[3] J. Sarkar, E-commerce sellers seek interoperability of ratings, reviews, https://timesofindia.indiatimes.com/business/india-business/e-comm-sellers-seek-porting-of-their-ratings/articleshow/65285454.cms, (2018) , Accessed date: 30 December 2018.
[4] T. Wallace, Ecommerce trends: 141 stats revealing how modern customers shop in 2017, https://www.bigcommerce.com/blog/ecommerce-trends/, (2017) , Accessed date: 30 December 2018.
[5] J. Chen, L. Teng, Y. Yu, X. Yu, The effect of online information sources on purchase intentions between consumers with high and low susceptibility to informational influence, Journal of Business Research 69 (2016) 467–475, https://doi.org/10.1016/j.jbusres.2015.05.003.
[6] S. Bag, M.K. Tiwari, F.T.S. Chan, Predicting the consumer's purchase intention of durable goods: an attribute-level analysis, Journal of Business Research 94 (2019) 408–419, https://doi.org/10.1016/j.jbusres.2017.11.031.
[7] R.E. Hostler, V.Y. Yoon, Z. Guo, T. Guimaraes, G. Forgionne, Assessing the impact of recommender agents on on-line consumer unplanned purchase behavior, Information and Management 48 (2011) 336–343, https://doi.org/10.1016/j.im.2011.08.002.
[8] J.A. Morente-Molinera, G. Kou, I.J. Pérez, K. Samuylov, A. Selamat, E. Herrera-Viedma, A group decision making support system for the Web: how to work in environments with a high number of participants and alternatives, Applied Soft Computing Journal 68 (2018) 191–201, https://doi.org/10.1016/j.asoc.2018.03.047.
[9] D. Valcarce, J. Parapar, Á. Barreiro, Finding and analysing good neighbourhoods to improve collaborative filtering, Knowledge-Based Systems 159 (2018) 193–202, https://doi.org/10.1016/j.knosys.2018.06.030.
[10] M.P. O'Mahony, N.J. Hurley, G. Silvestre, Detecting noise in recommender system databases, Proceedings of the 11th International Conference on Intelligent User Interfaces, 2006, pp. 109–115, , https://doi.org/10.1145/1111449.1111477.
[11] C.Y. Chung, P.Y. Hsu, S.H. Huang, βp: a novel approach to filter out malicious rating profiles from recommender systems, Decision Support Systems 55 (2013) 314–325, https://doi.org/10.1016/j.dss.2013.01.020.
[12] J.S. Lee, D. Zhu, Shilling attack detection - a new approach for a trustworthy recommender system, INFORMS Journal on Computing 24 (2012) 117–131, https://doi.org/10.1287/ijoc.1100.0440.
[13] Z. Yang, L. Xu, Z. Cai, Z. Xu, Re-scale AdaBoost for attack detection in collaborative filtering recommender systems, Knowledge-Based Systems 100 (2016) 74–88, https://doi.org/10.1016/j.knosys.2016.02.008.
[14] F. Zhang, Z. Zhang, P. Zhang, S. Wang, UD-HMM: an unsupervised method for shilling attack detection based on hidden Markov model and hierarchical clustering, Knowledge-Based Systems 148 (2018) 146–166, https://doi.org/10.1016/j.knosys.2018.02.032.
[15] J. Castro, R. Yera, L. Martínez, A fuzzy approach for natural noise management in group recommender systems, Expert Systems with Applications 94 (2018) 237–249, https://doi.org/10.1016/j.eswa.2017.10.060.
[16] L. Martinez, J. Castro, R. Yera, Managing natural noise in recommender systems, Theory and Practice of Natural Computing: 5th International Conference, TPNC

2016, Sendai, Japan, December 12–13, 2016, Proceedings 5, 2016, pp. 3–17, , https://doi.org/10.1007/978-3-319-49001-4_1.
[17] R. Yera, J. Castro, L. Martínez, A fuzzy model for managing natural noise in recommender systems, Applied Soft Computing 40 (2016) 187–198, https://doi.org/10.1016/j.asoc.2015.10.060.
[18] R.Y. Toledo, Y.C. Mota, L. Martínez, Correcting noisy ratings in collaborative recommender systems, Knowledge-Based Systems 76 (2015) 96–108, https://doi.org/10.1016/j.knosys.2014.12.011.
[19] S. Bag, S.K. Kumar, M.K. Tiwari, An efficient recommendation generation using relevant Jaccard similarity, Information Sciences (2019), https://doi.org/10.1016/j.ins.2019.01.023 In Press, online available on 11 January 2019.
[20] B.K. Patra, R. Launonen, V. Ollikainen, S. Nandi, A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data, Knowledge-Based Systems 82 (2015) 163–177, https://doi.org/10.1016/j.knosys.2015.03.001.
[21] S. Geuens, K. Coussement, K.W. De Bock, A framework for configuring collaborative filtering-based recommendations derived from purchase data, European Journal of Operational Research 265 (2018) 208–218, https://doi.org/10.1016/j.ejor.2017.07.005.
[22] S.L. Huang, Designing utility-based recommender systems for e-commerce: evaluation of preference-elicitation methods, Electronic Commerce Research and Applications 10 (2011) 398–407, https://doi.org/10.1016/j.elerap.2010.11.003.
[23] S. Bhattacharjee, R.D. Gopal, J.R. Marsden, R. Sankaranarayanan, Re-tuning the music industry: can they re-attain business resonance? Communications of the ACM 52 (2009) 136–140, https://doi.org/10.1145/1516046.1516081.
[24] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: a survey, Decision Support Systems 74 (2015) 12–32, https://doi.org/10.1016/j.dss.2015.03.008.
[25] C. Li, Z. Wang, S. Cao, L. He, WLRRS: a new recommendation system based on weighted linear regression models, Computers and Electrical Engineering 66 (2018) 40–47, https://doi.org/10.1016/j.compeleceng.2018.02.005.
[26] G.I. Doukidis, K. Pramatari, G. Lekakos, OR and the management of electronic services, European Journal of Operational Research 187 (2008) 1296–1309, https://doi.org/10.1016/j.ejor.2006.09.014.
[27] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowledge-Based Systems 46 (2013) 109–132, https://doi.org/10.1016/j.knosys.2013.03.012.
[28] G. Jiang, P.R. Tadikamalla, J. Shang, L. Zhao, Impacts of knowledge on online brand success: an agent-based model for online market share enhancement, European Journal of Operational Research 248 (2016) 1093–1103, https://doi.org/10.1016/j.ejor.2015.07.051.
[29] Y. Yan, R. Zhao, Z. Liu, Strategic introduction of the marketplace channel under spillovers from online to offline sales, European Journal of Operational Research (2017), https://doi.org/10.1016/j.ejor.2017.11.011.
[30] D. Cosley, S.K. Lam, I. Albert, J.A. Konstan, J. Riedl, Is seeing believing?: how recommender system interfaces affect users' opinions, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2003, pp. 585–592, , https://doi.org/10.1145/642611.642713.
[31] X. Amatriain, J.M. Pujol, N. Tintarev, N. Oliver, Rate it again: increasing recommendation accuracy by user re-rating, Proceedings of the Third ACM Conference on Recommender Systems, 2009, pp. 173–180, , https://doi.org/10.1145/1639714.1639744.
[32] X. Amatriain, J.M. Pujol, N. Oliver, I like it... i like it not: Evaluating user ratings noise in recommender systems, International Conference on User Modeling, Adaptation, and Personalization, 2009, pp. 247–258, , https://doi.org/10.1007/978-3-642-02247-0_24.
[33] B. Li, L. Chen, X. Zhu, C. Zhang, Noisy but non-malicious user detection in social recommender systems, World Wide Web 16 (2013) 677–699, https://doi.org/10.1007/s11280-012-0161-9.
[34] H. Luo, C. Niu, R. Shen, C. Ullrich, A collaborative filtering framework based on both local user similarity and global user similarity, Machine Learning 72 (2008) 231–245, https://doi.org/10.1007/s10994-008-5068-4.
[35] D. Anand, K.K. Bharadwaj, Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities, Expert Systems with Applications 38 (2011) 5101–5109, https://doi.org/10.1016/j.eswa.2010.09.141.
[36] G. Pitsilis, S.J. Knapskog, Social trust as a solution to address sparsity-inherent problems of recommender systems, arXiv Preprint arXiv:1208.1004, 2012. https://arxiv.org/abs/1208.1004v1.
[37] G. Guo, H. Qiu, Z. Tan, Y. Liu, J. Ma, X. Wang, Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems, Knowledge-Based Systems 138 (2017) 202–207, https://doi.org/10.1016/j.knosys.2017.10.005.
[38] X. Xu, K. Dutta, C. Ge, Do adjective features from user reviews address sparsity and transparency in recommender systems? Electronic Commerce Research and Applications 29 (2018) 113–123, https://doi.org/10.1016/j.elerap.2018.04.002.
[39] H.J. Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, Information Sciences 178 (2008) 37–51, https://doi.org/10.1016/j.ins.2007.07.024.
[40] J. Castro, R. Yera, L. Martínez, An empirical study of natural noise management in group recommendation systems, Decision Support Systems 94 (2017) 1–11, https://doi.org/10.1016/j.dss.2016.09.020.
[41] GroupLens, MovieLens datasets, https://grouplens.org/datasets/movielens, (2017) , Accessed date: 30 December 2018.
[42] S. Singh, S. Bag, M. Jenamani, Relative similarity based approach for improving aggregate recommendation diversity, In Proc. 12th Ieee Indicon Conf, 2015, https://doi.org/10.1109/INDICON.2015.7443856.

**Sujoy Bag** is an MS Scholar in the Department of Industrial and Systems Engineering, IIT Kharagpur, India and a former visiting scholar of Heriot-Watt University Edinburgh, UK. He has graduated from the Department of Computer Science and Engineering. His broad research area: Data Science, *E*-Business, Sentiment analysis, Digital marketing, and Recommender system etc. He is an Ad-Hoc reviewer in Journal of Business Research and active reviewer in several international journals including Computers & Industrial Engineering (Elsevier), Information Sciences (Elsevier), Journal of the Operational Research Society (Taylor & Francis Online), and Neurocomputing (Elsevier) etc.

**Susanta Kumar** is completed his dual degree (B-Tech + M-Tech) from the Department of Industrial and Systems Engineering, Indian Institute of Technology Kharagpur, India. His research interests include Supply Chain Management, Knowledge Management, Machine Learning and Optimization.

**Anjali Awasthi** is an Associate Professor at Concordia Institute for Information Systems Engineering (CIISE), Concordia University Montreal, Canada. She received her PhD in Industrial engineering and automation from INRIA Rocquencourt & University of Metz, France. Prior to Concordia, Dr. Anjali worked at Centre for Operations Excellence (COE), Sauder School of Business at University of British Columbia where she was involved in several projects on industrial applications of operations research. At University of Laval, she worked on wood supply chains and in France; she was involved in several European projects aimed at improving urban mobility in cities, city logistics and on cybernetic transportation systems. Dr. Awasthi has several years of industry and research experience in areas of automated transportation, sustainable mobility, sustainable supply chain management, quality assurance in supply chains, sustainable city logistic planning, and applied operations research. She is the author of several journal and conference papers on these topics.

**Manoj Kumar Tiwari** is a Professor in the Department of Industrial and Systems Engineering, Indian Institute of Technology Kharagpur, India. He has authored 300+ articles in peer-reviewed academic journals. He is holding the position of associate editor and the editorial board member in 20+ international journals, including the Computers and Industrial Engineering, Engineering Applications of Artificial Intelligence, Information Sciences, International Journal of Production Research, IEEE- System, Man, and Cybernetics: Systems, Journal of Intelligent Manufacturing, and Neurocomputing etc. His research interest includes Decision Support Models, Planning, Scheduling and Control Problems of Manufacturing System, Supply Chain Network, AI Applications in Manufacturing, Intelligent Search Technique, Nature Inspired Algorithm, Agent Technology and its Applications, Planning Problem of Assembly/Disassembly, Continuous Improvement Techniques. Prof Tiwari has led several projects and consultancy with prominent industry and government organization in India, such as Indian Air Force Repair Depot, DRDO, Indian Air Force, and ACC Cements etc. He is also actively involved in UK-India collaboration Next Generation supply chain project funded by EPSRC-UK. He is listed among top 20 most productive authors in the broad area of Production and Operations Management in the last 50 years (Published in IJPE, 2009), 2nd among many researchers working in Logistics and Supply Chain Management in India (Analysis of the Logistics Research in India-White paper published in TU Dortmund University, Dortmund Germany in 2012) and also Fellow of IISE, and INAE. He has also been placed in the scientific committee, keynote speaker and session chair of several international conferences of repute.