

# Challenges and Issues of Recommender System for Big Data Applications



Chandrima Roy and Siddharth Swarup Rautaray

**Abstract** The knowledge overload is a big problem in today's world. In reality, the information overload implies the availability of so much data or knowledge that goes beyond the user's manageable limits and causes a great difficulty in all kinds of decision taking. The main reason we need a recommender system in modern society is that because of the proliferation of the Internet, people have so many options to choose from. A recommender system refers to a system that can predict a user's future preference for a set of items and recommends the top items. It is a knowledge retrieval application that enhances accessibility as well as efficiently and effectively suggests relevant items to users by considering the user interests and preferences. A recommendation framework tries to tackle the problem of overloading information. There are so many other instances such as these, where we have plenty of data, but we can't decide what we want. Even though volume of information has increased, a new problem has arisen as people have had difficulty selecting the items they actually want to see. Recommenders systems have the ability to change the way websites interact with users and allow businesses to optimize their Return on Investment (ROI); based on the information, they can collect on the preferences and purchases of each customer. A traditional recommendation system cannot do its work without sufficient information, and big data offers plenty of user data such as past transactions, browsing history, and reviews for recommendation systems in order to provide accurate and efficient recommendations. In short, even the most advanced recommenders without big data can't be successful. This work primarily discusses and reflects on current issues, challenges, and research gaps in the production of high-quality recommender systems. Such problems and challenges will present new paths for study, and the target can be accomplished for high-quality recommender systems. The entire research is broken into major tasks including the study of state-of-the-art approaches for recommender system and big data applications; overcoming the problem of cold start,

---

C. Roy (✉) · S. S. Rautaray

Kalinga Institute of Industrial Technology (KIIT) Deemed to be University, Bhubaneswar, Orissa, India

e-mail: [chandrima.roy.1914@gmail.com](mailto:chandrima.roy.1914@gmail.com)

S. S. Rautaray

e-mail: [siddharthfcs@kiit.ac.in](mailto:siddharthfcs@kiit.ac.in)

scalability, and building a proactive recommender system; this research considers the process of development for a generic intelligent recommender system that can be work on more than one domain; it also expands the basic recommender program definition.

**Keywords** Recommender system · Big data · Challenges · Cold start · Scalability · Proactive

## 1 Introduction

A recommendation system aims to tackle the problem of information overloading. In reality, the explosion of information implies the presence of a lot of data or information that goes beyond the user's manageable limits and makes it very difficult to make decisions of any sort. This challenge arises mainly when the system is unable to handle and process this large volume of knowledge in a structured manner [1]. These are the standard tools effectively implemented in many commercial surroundings such as Flipkart, Netflix, Trip Advisor, and so on. For instances, the user typically gets a lot of choices in many e-commerce web applications, with very little time to explore it all. There are also other cases like this, where we have a lot of data but we can't really determine what we want. While there has been an improvement in the amount of knowledge available, a new problem has arisen, as it has become difficult for consumers to choose the items they really want. So that's where the system of recommendations comes into play. Big data is the engine of recommendation systems. In short, without big data, even the most advanced recommenders can't be successful. This work addresses and focuses mainly on current problems, difficulties, and research gaps in delivering high-class recommendation framework. These problems and obstacles will present new research paths, and the aim for high-quality recommender systems can be reached [2]. One disadvantage of the new network of recommenders is that they are not portable. A variety of recommendation systems are currently being developed for different industries but they are only adequate to fulfill the information needs of the users on that particular application domain. The systems vary from recommending music, events, diseases, and furniture to dating profiles. User browse to listen to a song from "Spotify," but user can't decide which one. User browse "YouTube" to watch some videos, but user can't decide which one to watch. For example, the number of movies that can be put in a blockbuster store which depends on the store's size. In comparison, the Internet nowadays enables people to access plenteous online resources. For example, "Netflix" has a vast collection of movies.

In reality, the information overload means the existence of so much data or knowledge that goes beyond the user's controllable bounds and causes a great effort in all kinds of decision taking [3]. This problem arises mostly when this huge amount of information cannot be handled and processed by the system in a systematic manner.

For example, a lot of choices are usually given to the user in many e-commerce web applications, but with very little time to discover them all.

A Recommender System (RS) may be described as a collection of software program and techniques that provide users with start updates for certain things that might be of use to them. Such recommendations help users make routine decisions [4]. For example, which film to watch, which songs to listen to, which novel to buy, and what reports to read, etc. Typically, a recommendation system is pictured as a platform for information exploration because it has the ability to create new and expanded recommendations to suit its user's needs. A recommender system mainly offers its facilities to two types of companies involved in e-commerce; one is the service provider whose goals to optimize its revenues; the other is the consumer who really likes its advantages in quality decision-making with the presence of a variety of relevant data based on the knowledge or expectations of others.

### ***1.1 Recommendation System Architecture***

Customers interact with recommendation systems via a web portal or mobile user interface where a profiler extracts consumer information based on feedback from explicit and implied methods; the interest of customers in various products is predicted through ranking algorithms which provide a list of pro-positioned items based on their calculated custom relevance. The data structure of the recommendation system is basically based on a database which stores item description and customer feedback and is constantly updated (Fig. 1). Because of the clustering and filtering features, recommendation systems are commonly used within e-commerce. They direct consumers to find new and unexpected items that are not discovered by themselves [5].

Taking ebay as a case, an Ajax event is triggered every time a user visits the web and clicks on a product. This event will generate an entry database—"When User X viewed Product Y" Where do we get the User details? If he is a user registered, then we get it from HTTP Session or we extract it from the cookie scheme. If it depends on cookies, then the recommendation only works as long as the user visits the site from the same browser. If a customer adds a product to his shopping cart, then orders a product, likes a product on Facebook or shares a summary of a product, identical events are shot. The website now has data obtained for the user, and it knows what he had been viewing, how often he was watching, what brand he may be interested in, what item he really likes, and so on.

### ***1.2 Big Data***

By day-to-day activities and interaction between user and computers, data is increasing exponentially. The generation of data is not restricted to the particular

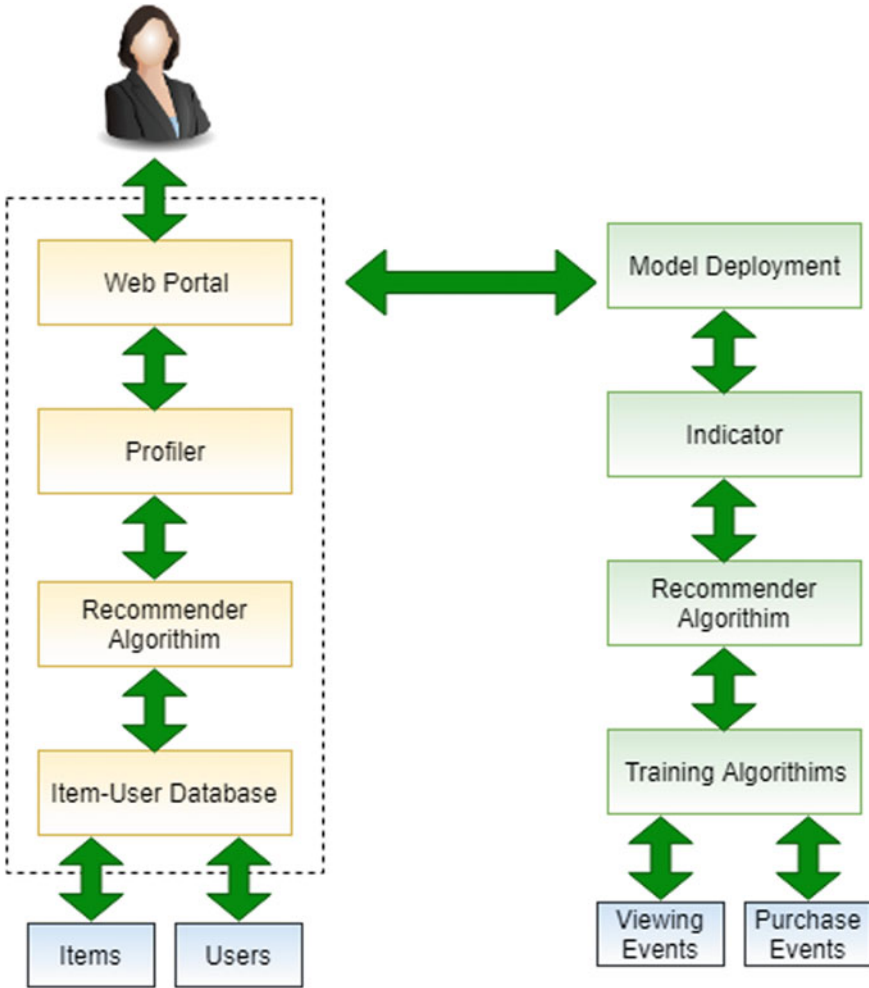
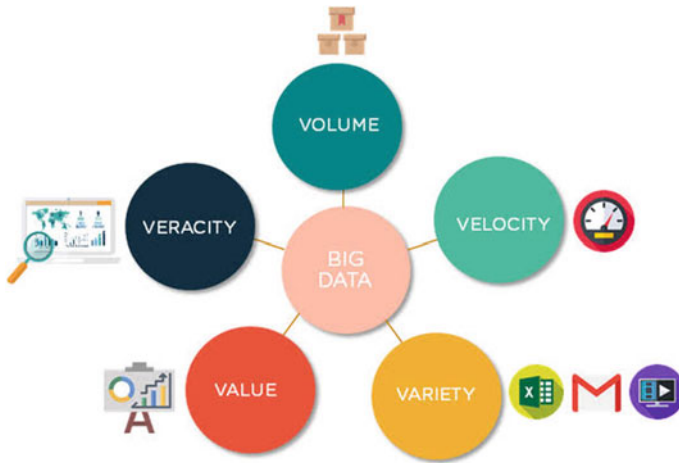


Fig. 1 General design of recommendation system [11]

domain, and any technique with which the data is collected, stored, and retrieved. Big data refers to the growing, complex sets of information that expands at rising levels [6]. This includes the amount of data, the size, or speed at which it is generated and gathered and the variety or probability of the enclosed data points. Big data often starts out from different sources and exists in various formats. The Internet of Things (IoT), the Cloud, Facebook, and Artificial Intelligence (AI) drive data sophistication through various types and sources of data. Big data, for example, comes from computers, video/audio, sensors, transactional software, networks, the online social networking sites—all of it was generated in real time and on a very huge scale [7]. The word “big data” can be defined as data that gets so large that traditional methods



**Fig. 2** 5V's of big data

cannot be used to process it. The amount of data called big data is a continuously changing factor and the continuous development of new methods for handling such big data. It fully changes our planet and does not mean that it is a phenomenon that will vanish in near future at any moment. It is also broken down by five vs. (Velocity, Volumes, Value, Variety, and Veracity) to make sense of the massive amount of data (Fig. 2).

In fact, big data dictates what recommendations do. Without the continuous provision of data, recommendations can't do a thing [8]. But the role of big data goes beyond numbers. The operations require a CPU which can run for hours. It can be worked with Hadoop to do this. The role of big data can therefore be summarized in the provision of appropriate, fast-acting data, and the necessary setup for the quick processing of the data [9]. Current systems aren't designed to handle such huge data volumes quickly. But getting only big data won't be enough to send strong recommendations.

The objectives of this paper can be stated as: (1) providing the general principles and techniques associated with different types of recommender systems (2) identifying and addressing some of the prominent problems and challenges in designing and developing a high-quality recommender system, such as cold start, scalability, and proactive recommender system.

Rest of the paper is structured as Sect. 2 presents general concepts and techniques relevant to cold start problem, Sect. 3 presents problems and challenges of scalability, Sect. 4 presents several approaches and techniques to produce a proactive recommender system, and finally, Sect. 5 concludes our discussion.

## 2 The Cold Start Problem in Recommendation

Cold start means the system doesn't have sufficient knowledge to make recommendations for a new user or a new item [10]. Any recommendation system will face cold start problems with the introduction of new client, product, or software. A new product registry or an introduction of new products makes it impossible to introduce an item to a customer as less information is available. The collective filtering cannot efficiently make recommendations for new user and new object event [11]. The content-based recommendation method is the solution to this problem, since it does not rely on the ranking of products. Another way is to identify visitors as browsers that are only there to search items. Two basic types of cold start problem are.

### 2.1 *New User Cold Start Problem*

The term cold start means that the program does not have sufficient information to suggest a new user or a new object. With the addition of new user, an object or a program, any recommendation program will face cold start problem [12]. Limited number of information is available which makes it problematic to recommend an item to a user. The issue of cold start is linked to the sparseness of information (i.e., for users and items) available in the algorithm recommendation.

#### 2.1.1 **Approaches for Cold Start User Mitigation Technique**

The model below focuses on mitigating the new user cold start problem, where a new user is asking for suggestions, and there is no data about their preferences [13]. Rating is very important, as it indicates a particular user's preferences. Additionally, there are no historical records.

The first step is responsible for identifying the new consumer into a particular category. For the classification, this model will use J48 algorithm represented in Fig. 3. The algorithm generates a decision tree based on a greedy top-down approach in which it considers each attribute that can distinguish all instances [14]. This process continues in a recursive fashion until the classification of all the results. The difference in entropy levels between attributes is used to select the best classification attribute.

During the second step, the algorithm uses a technique to locate the new user's "neighbors." This model will examine the user's important characters, and it will look for other users within the class that matches the new user best. The final result is determined in the third step. It is achieved by introducing statistical methods for predicting the new user's ratings. This model will handle the new user cold start problem.

**Algorithm J48 classification**


---

```

Input : an attribute-valued dataset  $D$ 
1 : Tree = {}
2 : if  $D$  is “pure” OR other stopping criteria met then
3 :     terminate
4 : end if
5 : for all attribute  $a \in D$  do
6 :     Compute information-theoretic criteria if we split on  $a$ 
7 : end for
8 :  $a_{best}$  = Best attribute according to above computed criteria
9 : Tree = Create a decision node that tests  $a_{best}$  in the root
10 :  $D_v$  = Induced sub-datasets from  $D$  based on  $a_{best}$ 
11 : for all  $D_v$  do
12 :     Tree $_v$  = J48( $D_v$ )
13 :     Attach Tree $_v$  to the corresponding branch of Tree
14 : end for
15 : return Tree

```

---

**Fig. 3** Pseudocode for J48 algorithm

The theory is that people with a common culture would most possibly have similar preferences [15]. A model will be implemented from the training data. Figure 4 represents the model for this process. The goal is to find a neighborhood, where the neighbors are users that belong to the same group as the group predicted by the model. After this step, similarity index will be calculated that combines similarity from the neighbors. Finally, the similarity measure and neighbors’ ratings are combined to get predictions.

## 2.2 New Item Cold Start Problem

The issue with the item cold start will cause the new item to miss the opportunity to be recommended and stay “cold” all the time. The proposed model would like to suggest new products to potentially interested users—for which no interests have been expressed so far [16].

The method employs interrelationships mining. It can derive functionality based on a comparison of various attribute values. One of the main concepts of interrelation mining is to reflect similar characteristics based on a comparison of new attributes called interrelated attributes between values of different attributes. First, most similar item will be chosen to include a new item’s neighborhood, and neighborhood rating information will be used to estimate the new item’s rating value. Lastly, highest rating product will be recommended to a target customer.

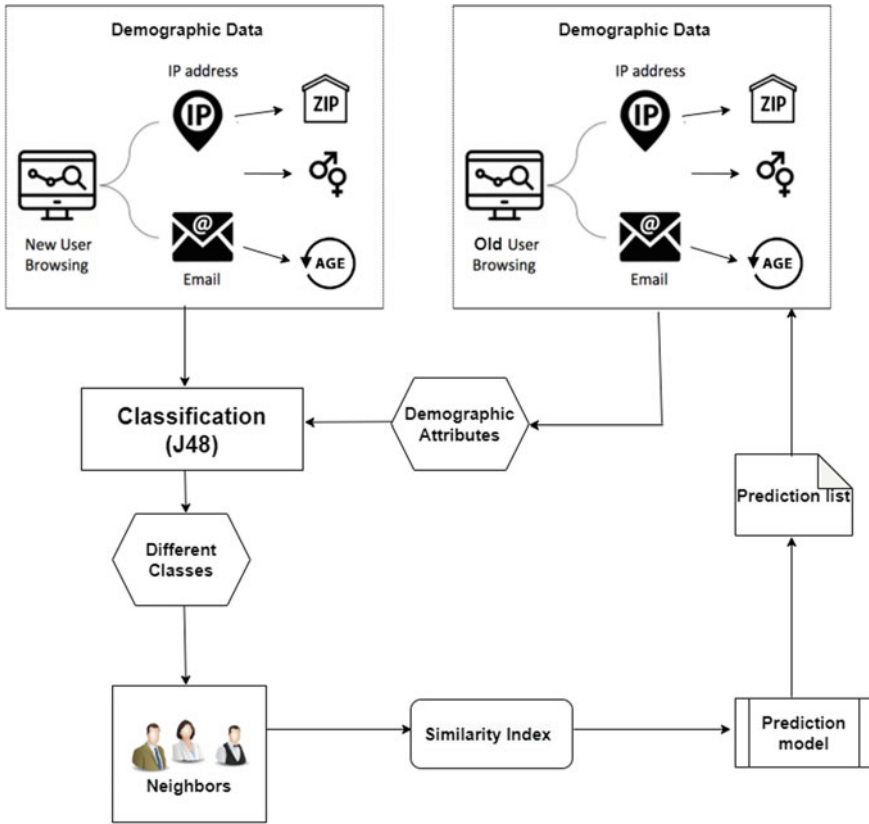


Fig. 4 Model for new user cold start problem

### 2.2.1 Approach for Cold Start Item Mitigation Technique

#### Step 1: Extracting the Relation Between Each Pair of Attributes

The purpose of this task is to extract semantic associations between pairs of co-occurring entities. This approach is mainly organized in two stages. First is defining instances of candidate relationships and then classifying instances of candidates into true and invalid ones. Classifiers are based upon training corpus as a standard classification process in which all relationships and their arguments were manually annotated [17]. The annotated relationships are viewed as positive examples of training, while the majority of the non-labeled co-occurring pairs are considered derogatory examples of training.

#### Step 2: Calculating Similarity Index Between the Attributes

The simplest way to determine linear dependency between two or more attributes is to use correlation. In a large set of attribute values, set of each attribute values can be treated as a vector. The scalar product of two vectors (with subtracted mean value)



and normalized with both norms is a good measure of similarity [18]. When the value is 1, they are equal to a scale factor; when the value is zero, they are orthogonal.

$$R = a.b / (\|a\| \|b\|)$$

where  $R$  is correlation, and  $a, b$  are observed vectors or sets of attributes

$$a = (a_1, a_2, a_3, a_4, \dots, a_n) \quad b = (b_1, b_2, b_3, b_4 \dots b_n)$$

### Step 3: Selection of High Rated Item

At the last stage, the model will predict the top  $N$  high rated items which will be recommended to the user.

## 3 Scalability

In recent years, the rapid growth of customers and goods presents some critical challenges for recommendation systems that produce high-quality recommendations for millions of customers and goods and conducting several recommendations every second. The outcome is a rich archive of information which can be used if managed correctly to boost the user experience and rise their involvement. Both existing large-scale recommender systems in the real world have developed in scalable distributed database systems for data management and distributed and parallel recommendation finding algorithms [19].

### 3.1 Scalable Neighborhood Algorithm

Scalability is a key factor in determining the type of recommender systems that should be used. More complex systems require more people to build/maintain, potentially harder to hire, with greater hardware costs. This can be a long-term investment, and thus, business should realize the incremental business advantage versus the increased expense. Having said that here are some core elements of creating scalable systems. Offline batch processing and serving online. For a large number of users and products, one has to measure offline by lot easily obtainable recommendations.

The recommender framework for YouTube split the modeling process into two phases. During the first step, only user-item interaction data is used to pick hundreds out of millions of candidates. For more selection and ranking, it is then feasible to use more information about candidate videos in the second process. Although softmax or other functions are used for training in the output layer, the probability does not have to be measured during real-time serving time, and the closest-neighbor method can be used on the output from the last hidden layer [9].

### 3.1.1 Scalability in Data Management

Database Management Systems (DBMS) play a key part in social networking applications, where vast volumes of data generated and aggregated at various, geographically separate places need to be stored, extracted, and distributed in an efficient manner [20]. Centralized relational DBMSs are built to ensure uniformity but can scale-up to the needs of global social networks, because of this Twitter and Facebook have been exploring different distributed and NoSQL data storage systems like MongoDB, Megastore, Cassandra [21]. Relational database management systems are “not cloud friendly” because of the connections and interactions between stored data, even when running using cloud structure, NoSQL databases will scale out better. However, they also face many challenges which have to do with data partitioning and replication, and for this reason, middleware solutions have been developed.

### 3.1.2 Scalability Using Deep Learning for Big Data

The use of hybrid filtering is another way of confronting sparsity, cold start, scalability, and other challenges [22]. This approach includes the combination of various recommendation development techniques, seeking to increase prediction accuracy while highlighting shortcomings of the individual methods.

In order to learn parameters, traditional deep neural networks need vast quantities of data, which is a computationally intensive process that takes considerable time to train a model. When the data size grows exponentially and the deep learning models become more complex, more computational power and memory are required, such as high-performance computing (HPC) resources to train a model of accuracy in a timely manner. Given the ongoing efforts to train and infer deep learning models to increase competitiveness, many current deep learning algorithms are notoriously difficult to scale and parallelize due to inherent interdependencies in the computational steps as well as training data. Clustering techniques work by identifying user groups which tend to have similar tastes. By combining the views of the other users within that cluster, predictions can be made for a person once the clusters are formed. Many clustering techniques, with partial participation, represent each consumer in multiple clusters. Hence, the prediction is an average over the clusters, weighted by degree of involvement. Clustering strategies usually generate less personal recommendations than other methods, often resulting in less accuracy than neighboring algorithms [23]. Figure 5 represents the neighborhood formation after clustering.

#### 3.2 Algorithm: Clustered Neighborhood Formation

**Step 1.** Using the training dataset to construct user partitions using the clustering algorithm.

**Step 2.** Determine a given user’s neighborhood.

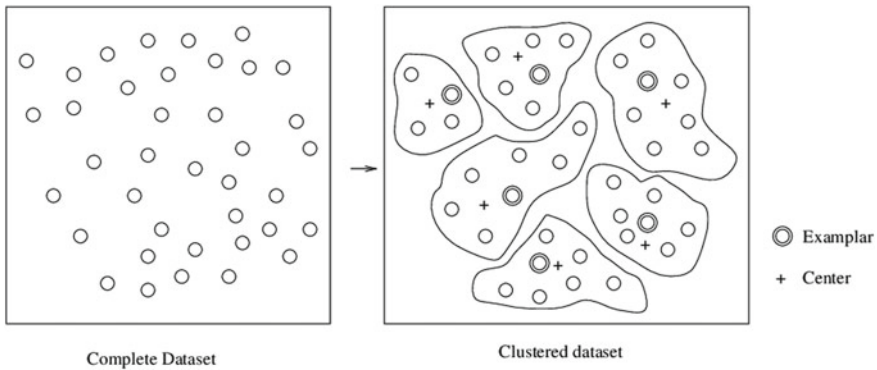


Fig. 5 Neighborhood formation after clustering

**Step 3.** Upon obtaining the neighborhood, the classic collaborative filtering algorithms is used to generate prediction from it.

The approach has two advantages—first, it decreases the sparsity of the data collection, and second, the prediction generation is much quicker due to the reduction of dimensionality and the use of a static precomputed neighborhood.

## 4 Proactive Recommender System

A proactive system would be able to deliver highly relevant information and determine at what point user need any item or information and at what moments information needs are more critical and less disruptive. Recommender systems main aim is to mitigate the effects of overloading information/choice. Normally, recommendations are prepared based on anticipating what’s going to be useful or relevant for users. Therefore, it is vital that they are relevant to consumers, whether in relation to their information needs, current behavior or emotional state. This provides a detailed understanding of the users "context but also knowledge of the past of previous users" interactions within the system. A lot of existing systems need active user participation and focus. Recommendation systems manage the distribution of recommendations without taking into account the users’ needs when, when, or how the recommendations are made. Proactive recommendation systems offer a more proactive approach to recommendations delivery by pre-identifying information needs and working on behalf of users with minimal effort and without interruption [24].

## ***4.1 Proactive Recommendation***

Recommended systems are a well-established and effective solution to reduce the overload of choices as customers face a range of materials, goods, applications, and services, etc. In the ideal scenario, users receive highly relevant and interesting feedback (products) while organizations (system providers) benefit from increased user interaction. Many current recommender systems need active user interaction and attention. Recommender system manages the distribution of recommendations without taking into account the expectations of users on when, where, or how the recommendations are delivered. There is still growing competition for consumer attention with an explosion of applications/services. PS aims to predict consumer needs ahead of time and then take effective action on behalf of them. Actions are typically linked to the delivery at the right time, location, etc., of the appropriate content. Therefore, if appropriate, then proactive recommenders systems should include specific suggestions, without disrupting users in any way possible [25].

Constructive systems can create highly interactive environments in which they can assist and improve user interaction and provide comfort and assistance when needed. Overall, consumers demonstrate a strong interest in implementing and following constructive suggestions. Practical implementation needs to be treated with caution, because some users feel that they will become too dependent on these systems, while others also refrain from using them because of unwanted interruptions and frustration.

## ***4.2 Intelligent Proactive Recommender System***

This strategy incorporates information relating to a user gathered from their social networking sites, along with mobile devices, to proactively suggest relevant information to the consumer. The recommendation process in Fig. 6, therefore, includes a background model that specifies when and when to suggest the relevant information to the consumer (Books, music, news, movies).

User-related information, along with low-level information, is leveraged from the user's mobile device, laptop, or tab/iPad, to infer the user's condition. The approach we are putting forward aims to suggest specific things that suit the situation of a user without waiting for the user to initiate an operation. We consider that the daily routine of the user is defined as a collection of situations and the profile of the user and organized within a knowledge database that represents a particular category of information [26].

After this knowledge, extraction can be done from the collected data about the user. Mostly, three categories of knowledge will be extracted, which are user profile: User information and associated interests; Location: User location taken from GPS coordinates; Time: numeric or intermittent labels (morning, Evening, night). To capture

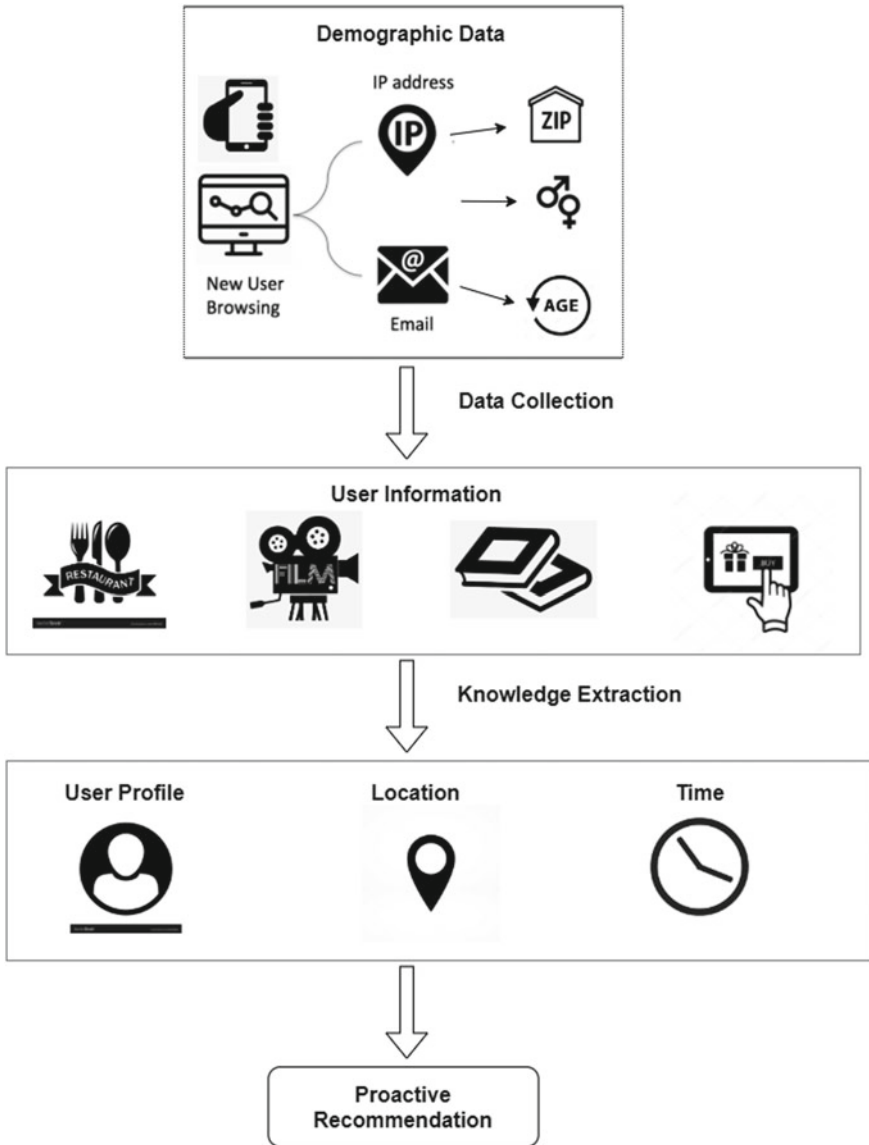


Fig. 6 A general overview of intelligent proactive recommender system

the context, these dimensions are instantiated using the sensors embedded in the mobile device of the user.

## 5 Conclusion

In the past few years, recommender systems have been used to alleviate knowledge and cognitive overload problems by recommending relevant and applicable items to users. Several improvements have been made in this regard to achieve a high quality, finely tuned recommendation system. This study aimed on these critical issues and challenges and proposed three models to mitigate these problems such as cold starting, scalability, and proactive recommendation system. Scaling according to the user's need and yet getting the recommendation device optimized is still an order of the day. For various applications which generate "Big Data," a new recommendation method is proposed in this research that has the ability to process heterogeneous data.

Currently, several recommendation systems are built for various sectors but they are only sufficient to satisfy the users' information needs on that specific application domain. One limitation of current recommendation system is that they are not portable. This research considers the process of development for an adaptive intelligent recommendation system that can be work on more than one domain that also extends the concept of basic recommendation system. To use dynamically auto-updated data from external websites, this recommendation framework can be designed to process complex data at the same time as active user requests arrive.

## References

1. Ramzan, B., I.S. Bajwa, N. Jamil, R.U. Amin, S. Ramzan, F. Mirza, and N. Sarwar. 2019. An intelligent data analysis for recommendation systems using machine learning. *Scientific Programming*.
2. Fu, M., H. Qu, Z. Yi, L. Lu, and Y. Liu. 2018. A novel deep learning-based collaborative filtering model for recommendation system. *IEEE Transactions on Cybernetics* 49 (3): 1084–1096.
3. Zhang, Y. 2016. GroRec: A group-centric intelligent recommender system integrating social, mobile and big data technologies. *IEEE Transactions on Services Computing* 9 (5): 786–795.
4. Sabitha, S., and T. Choudhury, T. (2018). Proposed approach for book recommendation based on user k-NN. In *Advances in Computer and Computational Sciences*, 543–558. Springer: Singapore.
5. Chen, J., H. Zhang, X. He, L. Nie, W. Liu, and T.S. Chua. 2017, August. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 335–344.
6. Roy, C., S.S. Rautaray, and M. Pandey. 2018. Big data optimization techniques: a survey. *International Journal of Information Engineering & Electronic Business*, 10 (4)
7. Shu, J., et al. 2018. A content-based recommendation algorithm for learning resources. *Multimedia Systems* 24 (2): 163–173. <https://doi.org/10.1007/s00530-017-0539-8>.
8. Wang, D., Y. Liang, D. Xu, X. Feng, and R. Guan. 2018. A content-based recommender system for computer science publications. *Knowledge-Based Systems* 157: 1–9.
9. Sarwar, B.M., G. Karypis, J. Konstan, and J. Riedl. 2002, December. Recommender systems for large-scale e-commerce: Scalable neighbourhood formation using clustering. In: *Proceedings of the Fifth International Conference on Computer and Information Technology*, vol. 1, 291–324.

10. Sardianos, C., N. Tsiarakis, and I. Varlamis, 2018. A survey on the scalability of recommender systems for social networks. In *Social Networks Science: Design, Implementation, Security, and Challenges*, 89–110. Cham: Springer.
11. Roy, C., M. Pandey, and S. Swarup Rautaray, 2018, April. A proposal for optimization of data node by horizontal scaling of name node using big data tools. In *2018 3rd International Conference for Convergence in Technology (I2CT)*, 1–6. IEEE.
12. Deng, H. 2019. *Recommender Systems in Practice*. <https://towardsdatascience.com/recommender-systems-in-practice>
13. Bobadilla, J., F. Ortega, A. Hernando, and A. Gutiérrez. 2013. Recommender systems survey. *Knowledge-Based Systems* 46: 109–132.
14. Maqsood, T., O. Khalid, R. Irfan, S. Madani, and S. Khan. 2016. Scalability Issues in Online Social Networks. *ACM Computing Surveys (CSUR)* 49 (2): 40.
15. Peska, L., and H. Trojanova. 2017. Towards recommender systems for Police Photo Lineup. [arXiv:1707.01389](https://arxiv.org/abs/1707.01389).
16. Smirnova, E., and F. Vasile. 2017. Contextual sequence modeling for recommendation with recurrent neural networks. [arXiv:1706.07684](https://arxiv.org/abs/1706.07684).
17. Roy, C., M. Pandey, and S.S. Rautaray. 2018. A proposal for optimization of horizontal scaling in big data environment. In *Advances in Data and Information Sciences*, 223–230. Singapore: Springer
18. Chatzis, S., P. Christodoulou, and A. Andreou. 2017. Recurrent latent variable networks for session-based recommendation. [arXiv:1706.04026](https://arxiv.org/abs/1706.04026).
19. Zanotti, G., M. Horvath, L. Barbosa, V. Immedisetty, and J. Gemmel. 2016. ACM: Infusing collaborative recommenders with distributed representations. In *Conference Proceedings of the 1st Workshop on Deep Learning for Recommender Systems* 35–42.
20. Braunhofer, M., F. Ricci, B. Lamche, and W. Wörndl. 2015, August. A context-aware model for proactive recommender systems in the tourism domain. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, 1070–1075.
21. Melguizo, M.C.P., L. Boves, and O.M. Ramos. 2009. A proactive recommendation system for writing: Helping without disrupting. *International Journal of Industrial Ergonomics* 39 (3): 516–523.
22. Sabic, A., and M. Zanker. 2015. Investigating user's information needs and attitudes towards proactivity in mobile tourist guides. In *Information and Communication Technologies in Tourism 2015*, 493–505. Cham: Springer.
23. Sabic, A. 2016, September. Proactive recommendation delivery. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 459–462.
24. Roy C., K. Barua, S. Agarwal, M. Pandey, S.S. Rautaray. 2019. Horizontal scaling enhancement for optimized big data processing. In: A. Abraham, P. Dutta, J. Mandal, A. Bhattacharya, S. Dutta (Eds.), *Emerging Technologies in Data Mining and Information Security. Advances in Intelligent Systems and Computing*, vol 755. Singapore: Springer. [https://doi.org/10.1007/978-981-13-1951-8\\_58](https://doi.org/10.1007/978-981-13-1951-8_58)
25. Kermi, I. 2017. *A hybrid model for context-aware proactive recommendation* (Doctoral dissertation). University of Toulouse, University of Toulouse III-Paul Sabatier.
26. Motadoo, S. 2018. Resolving cold start problem using user demographics and machine learning techniques for movie recommender systems.