

Received April 18, 2020, accepted May 7, 2020, date of publication May 12, 2020, date of current version May 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2994056

A Hybrid Action-Related K-Nearest Neighbour (HAR-KNN) Approach for Recommendation Systems

SUNKURU GOPAL KRISHNA PATRO¹, BROJO KISHORE MISHRA¹, SANJAYA KUMAR PANDA², RAGHVENDRA KUMAR¹, HOANG VIET LONG^{3,4}, DAVID TANIAK⁵, AND ISHAANI PRIYADARSHINI⁶

¹Department of Computer Science and Engineering, GIET University, Gunupur 765022, India

²National Institute of Technology, Warangal 506004, India

³Division of Computational Mathematics and Engineering, Institute for Computational Science, Ton Duc Thang University, Ho Chi Minh City 758307, Vietnam

⁴Faculty of Mathematics and Statistics, Ton Duc Thang University, Ho Chi Minh City 758307, Vietnam

⁵Faculty of Information Technology, Monash University, Melbourne, VIC 3800, Australia

⁶Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716, USA

Corresponding author: Hoang Viet Long (hoangvietlong@tdtu.edu.vn)

ABSTRACT Recommendation System (RS) has been broadly utilized in various areas and discovers product recommendations during an active user interaction in E-Commerce sites. Tremendous growth of users and products in recent years has faced some key challenges. There are numerous online sites that present many decisions to the user at once, which is strenuous. Moreover, finding active user or right product is an important task in RS. Existing works have been proposed to recommend a product by considering user inclination and socio-demographic behaviour. In this paper, we propose a Hybrid Action-Related K-Nearest Neighbour similarity (HAR-KNN) recommender that consolidates the simplicity of hybrid filtering to enrich user behaviour matrix with formation of the vector of features. It will classify the features using race classifiers from both quality and quantity aspects. The proposed approach also addresses the problems of the previous methods to efficiently evaluate user preference on products and balance feature analysis. The K-NN classification method has been qualified online and real-time to find user behaviour data coordinating to a specific user group containing the relationship between the similarity of many users and target users from a huge amount of data. The proposed experimental result is evaluated based on measures such as Mean Absolute Error (MAE), Mean Square Error (MSE) and Root Mean Squared Error (RMSE) with the lowest error of 0.7165, 0.7201 and 0.7322 separately. High predictive measures like Precision (P), Recall (R) and F1 are found to have values 0.8501, 0.2201 and 0.3507 respectively.

INDEX TERMS Recommendation system (RS), user behaviour data, hybrid filtering, K-NN, behavioural matrix.

I. INTRODUCTION

ONLINE commercial centres make their very own benefit dependent on their notices while business metric has the business enthusiasm to rank higher on suggestions to draw in more users [1]. Traditional retail can present only popular products, but online can present a variety of products. Due to the enormous information available across the web, it is difficult for users to comprehend whether the items presented by recommender frameworks are accurate or not [2]. A recommender system extracts user's interest

The associate editor coordinating the review of this manuscript and approving it for publication was Cesar Vargas-Rosales .

from related dataset and provides quality recommendations. Recommender systems are a significant part of E-Commerce that use machine learning [3] and data mining techniques [4] to filter the unseen information and predict whether the user would like a particular item. An intelligent system is a special type of recommender system used to exploit the historical user ratings on data that comes from mined-relevant data through data mining process [5]. Therefore, to achieve the same proposed framework, methodologies like content-based (CB) [6], collaborative based (CF) [7], and hybrid filtering (HF) [8] techniques are required. Furthermore, a lot of factors in learner qualities make a distinction based upon information, learning style and configurations of consecutive

learning where methodologies like CF, CB individually are not sufficient or reasonable to detect the distinction [9]. For accuracy of recommendations, additional information needs to be incorporated into the recommendation process along with the ratings. Then again, the RS framework utilizes SPM for prediction [10]. However, recommendations for E-Commerce resources differ from other domains since learners have different characteristics such as knowledge level and learning styles which influence the user preferences as well as personalization of learner recommendations. [11] Proposed a hybrid CF and SPM framework is used for recommending products. However, the research does not consider extra learner attributes in the recommendation process.

We propose HAR-KNN, an integrated approach of the K-NN algorithm with user behaviour matrix with the following motivations to overcome some problems in existing works: each user rates only a small subset of available products, so most of the cells in the rating matrix are empty. Finding similarity among nearer related users and items is a challenging one. Since the number of items is generated only for a user, hence it is user-specific. It might be difficult to comprehend the taste of the users. This leads to a problem of sparsity. The proposed HAR-KNN method can combine both collaborative and content-based features that categorize user goals and analyze the attributes of many users to form a larger target to accurately predict recommendations. The hybrid filtering solutions help to avoid scalability problems that exist in CB and CF.

The motivation of the design goals is researched as follows,

- With the growth of number of users, the system needs more resources to process information and to give recommendations. So, we propose recommendation tasks that categorize user's goals to recommend accurate predictions using Hybrid Action-Related K-NN Similarity (HAR-KNN) recommendation system.
- Using the K-NN algorithm, vector of features generates behaviour records for each user and samples a similar purchase intention to analyze the attributes of many users to form a larger target and accurately predict recommendations.
- Different permutations and combinations of the above approach were applied to improve the recommendation accuracy for generating recommendations that match user preference better.
- We demonstrate the effectiveness of the proposed framework and the importance of multi-attribute behaviour data from a real E-Commerce site.

The rest of the paper is organized as follows: The general introduction of RS based filtering techniques is discussed in section I. Section II presents the recent works and reviews related to configuration-based RS. Section III represents the proposed recommendation scheme that describes phases for the recommendation service process. Section IV presents the experimental evaluation achieved by the performance measures along with other existing methods. A case study in which the extracted dataset discussion is presented in

section V. Lastly, the overall conclusion of the work is presented in section VI.

II. RELATED WORK

The existing work carried out in this section is based on the recommender system, including recommendation techniques and attentional models in RS.

Lv *et al.* [14] establish a similarity model in which the component module includes information sharing units among various modalities to give the suggestion model dependent on multimodal information source named Multimodal Interest-Related Item Similarity model (Multimodal IRIS). The combination of visual representation and textual information might be able to consider the effects of different modalities at the same time. This multimodal feature can model the interest relevance between the different historical items and target items. Also, the expansion of multimodal data and the diverse intrigue importance of recorded items upgrade the suggestion list.

Pereira *et al.* [15] proposed a new personalized RS approach for configuring the product line process. The reusable features of product lines are intended to evaluate the platform across multiple products. In this, large feature models lead to a problem of dimensionality. Consequently, this approach agrees with 6 recommender algorithms to configure the context of the product line. Through this approach, a slightly relevant feature set decides the makers to organize the product. 3 of the 6 current recommender algorithms identify the relevant features. Hence, this approach may assist users in understanding their feature preferences. The experimental results depict that the proposed approach demonstrates better efficiency and quality-related on two real-world datasets and selects efficient features to support the decision-makers.

Hwangbo *et al.* [16] proposed a K-RecSys framework which expands the collaborative filtering suggestion framework by prescribing the space attributes. This structure merges the online product click data and disengaged product bargain data weighted to reflect the plan items according to the customer inclinations. The customer liked to invest less energy while the suggestions are exact. Hence, the user wishes to buy the item through the item data is as of now liked. To confirm the exhibition, a real working framework is to contrast the K-RecSys with the existing collaborative filtering system. Regardless these outcomes, the experiments are limited and cannot notify the use of online and offline failure time.

Bandyopadhyay *et al.* [17] mentioned a recommended approach for satisfying customer needs using Artificial Neural Networks (ANN). In this approach, student pattern buying has been considered and similar types of products are recommended to other student groups. To satisfy the needs of every customer is the main concern of this approach. The time-series data are predicted to appropriate the data using NN and it is possible to estimate the prediction error. Utilizing historical data, the prediction and forecasting data

have been easily generating product prediction. The collected data can be directed through different online reviews. Then, the unwanted data has been pre-processed. After that, the routine of the model has been trained using feed-forward architecture. Finally, the model can be evaluated by training and testing related to MSE. Thus, the result shows that the predicted output deserves an accurate target using various datasets.

Chen and Wang [18] mentioned a structure-based RS for considering the suitable item to the target user named Structural Balance Theory-based Recommendation (SBT-Rec). While given the sparsity of gigantic proportion of rating data in electronic business, similar things maybe both are absent from the site page. For managing the particular proposals, the SBT-Rec incorporates both user-based collaborative filtering (CF) and item-based CF suggestion in the presentation proportions of item suggestions. Besides, the customized necessities from various users can't coordinate in a specific time.

Ji *et al.* [19] proposed a Hybrid Recommendation model dependent on customer ratings, reviews and social data. This model comprises feature generation, review transformation, model training, community detection, feature blending, and prediction and evaluation. The experimental model is utilized to distinguish and break down the audit content to decide the social networks with convolutional suggestion models like network-based models. This shows the suggestion exactness can be precisely improved dependent on review texts and social networks.

Yang *et al.* [20] proposed a stepwise discovery strategy to identify the spot irregular evaluations that have been created for the combination of collaborative suggestion methods to analyze the similarity and feature extraction between the products. A set of samples is gathered from the user profiles for constructing a rating matrix. It is presumed that the items can be caught by extensively breaking down both the dispersions of mean and prescient errors of items and users. For huge-scale data, the proposed model detects the rating distribution, rating intension and time series analysis.

Previous researches in personalized RS consider a few attributes of structured and unstructured data which is not informative enough to produce any recommendation list. RS system architecture is constrained by multi-attribute information as input matrices which manages the service process. The service process has to access the real e-commerce data including not only products, but also multi-attribute behaviour data to generate predicted ratings. The integration of rating quality will be improved dramatically using multi-attribute information. The existing methods have these gaps in satisfying the abovementioned needs by the recommendation service process.

III. HYBRID ACTION-RELATED K-NN SIMILARITY RECOMMENDER SYSTEM

In this section, the proposed hybrid recommendation process takes a collection of behaviour-related data as input.

This hybrid technique is applied over the dataset to assess the gathered information as input and processes the explicit feedback to frame sorted out data. The data involving the attributes of the user information is now integrated into user behaviour rankings to various products. All the information is valid in the original database that the records are extracted and the large volume of significant data is pre-processed. The hybrid approach combines the simplicity of finding the users that have similar interests with active users. Then, the behaviour pattern extracted from the behaviour prediction and inclination is collected from hybrids filtering to choose target products as recommendations. It likewise chooses the K-NN of the active user that is allocated into a feature characterization that improves the association among users and their favoured products by a classification method. This process provides recommendations according to changing user behaviour. It constructs the pattern information for all products in the behaviour knowledge about the distribution of the data. The classification is based on taking the closest neighbours for determining the vector features of neighbour and product. Thus, the similarity is measured between the number of users and the target user. The outcomes can be combined with data that refers to user activities and purchase history. The proposed recommendation model is shown in Fig 1.

A. RELATION FOR RECOMMENDER SYSTEMS

In this system, the recommendation model for the behaviour scenario could define the browsing user and recommender. We describe the relation in the information system that can be achieved by making assumptions.

Consider, User $U = \{u_1, u_2, \dots, u_i\}$ and product $P = \{p_1, p_2, \dots, p_i\}$ be the set of products. To retrieve the space, the user and product information are stored in the database. The U and P relation can be represented as $R \subseteq U \times P$.

B. USERS RECOMMENDER BEHAVIOUR

The user recommender behaviour is a shared activity among the login user and RS. By logging onto the system, the first user logins to suggest one or more various products. The user chooses a product as his/her decision or ends the connection. Naturally, the user's behaviour logs certain behaviour conduct that happens more with lower interest and generally with higher interest. According to their behaviour information, if the user just specifies the clickstream or any other data to product, it will be called behaviour-based data and denoted by 3 tuples,

$$X : < U, L, \text{ and } M > \quad (1)$$

where X is the sequence of user actions ordered by time, L is a purchased product ID and M is purchased product.

Table 1. Shows the user's behaviour matrix mined from the user's behaviour file.

Let U_k labels the user behaviour and $n_{i,k}$ denotes the average number of U_k for user i . The user behavioural matrix

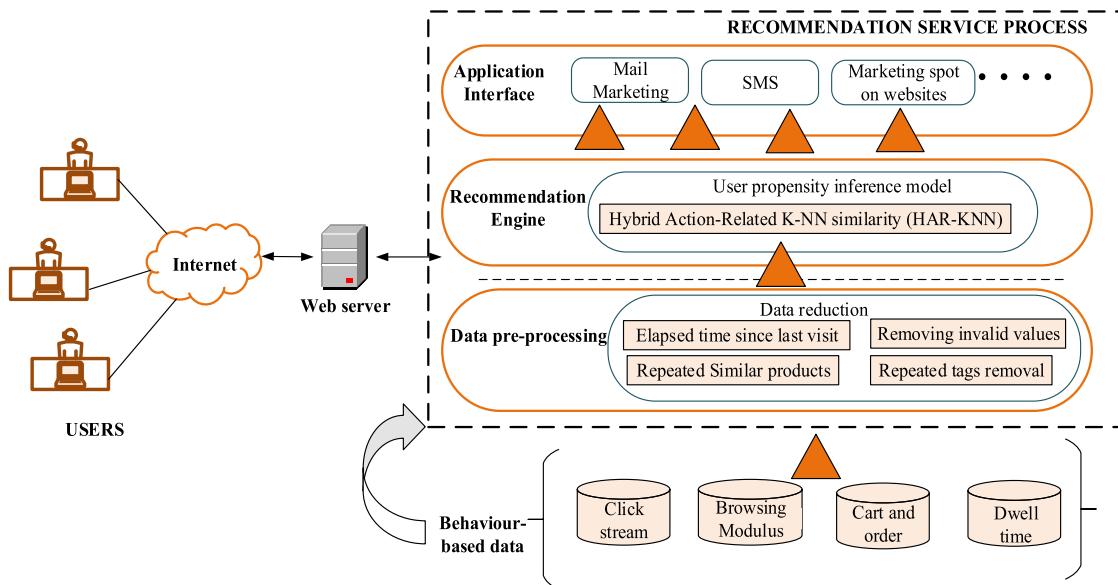


FIGURE 1. Proposed recommendation model.

TABLE 1. Users behaviour matrix.

User	u_1	u_2	...	u_m
u_1	$n_{1,1}$	$n_{1,2}$	$n_{1,3}$	$n_{1,m}$
u_2	$n_{2,1}$	$n_{2,2}$	$n_{2,3}$	$n_{2,m}$
...
u_i	$n_{i,1}$	$n_{i,2}$	$n_{i,3}$	$n_{i,m}$

is shown in Table 1 by equation (1),

$$W_k = \frac{\sum_k^m n_{U_k} / n_{U_i}}{\sum_i^m \sum_k^m n_{U_k} / n_{U_i}} \quad (2)$$

where n_{U_k} represents the average number of U_k and W_k describes the weight of U_k .

C. RECOMMENDATION SERVICE PROCESS

In this process, there are three main phases: Data Collection, Data Pre-processing and Recommendation Engine.

1) DATA COLLECTION

In this component, the behaviour-based data of users has been obtained from the Amazon dataset [14]. From the collected dataset, the principle records consist of the user's behavioural file and the product list file. The user's behavioural file consists of personal background such as clickstream, dwell time, browsing modules of the user. Data type collected from the E-Commerce site: User ID, product name (electronics, books, dresses, etc,), session ID, purchases, number of clicks, product ID, category and time. Based on user data, the collected data is scheduled repeatedly within the user requests.

2) DATA PRE-PROCESSING

In this component, there is a pre-processing module to remove the unnecessary and redundant data that are removed from the log files. It has unwanted files like elapsed time since the last visit, removing invalid values, repeated similar products, and repeated tags. Naturally, the rating matrix is used to perform the work in which the ratings represent users' feedback. In our behaviour database, we use user behaviour matrix for different behaviour types to serve as each user's implicit ratings towards different products. The scarcity (S) of this matrix is measured as,

$$S = \frac{N_{non-zero} * N}{N_{total}} \quad (3)$$

where $N_{non-zero}$ denotes the number of non-zero values, N represents the needed value of recommendation system and N_{total} is the total count values in the matrix.

3) RECOMMENDATION ENGINE

This section presents the approach used to build the RS based on user behaviour relations, combining the approaches of hybrid filtering and the K-NN algorithm. The K-NN identifies the number of similar users through calculating the cosine value. Each neighbour recommends a certain number of products. The category represents neighbour purchased products in different categories of users. Next, the feature represents the vector that will be used to compute the Euclidean distance and finds the nearest neighbours of the user. Finally, the vector features are classified to estimate the similarity between the input instances and k-nearest instances using a weighted K-NN algorithm. To develop the HAR-KNN recommendation approach, the following modules were created.

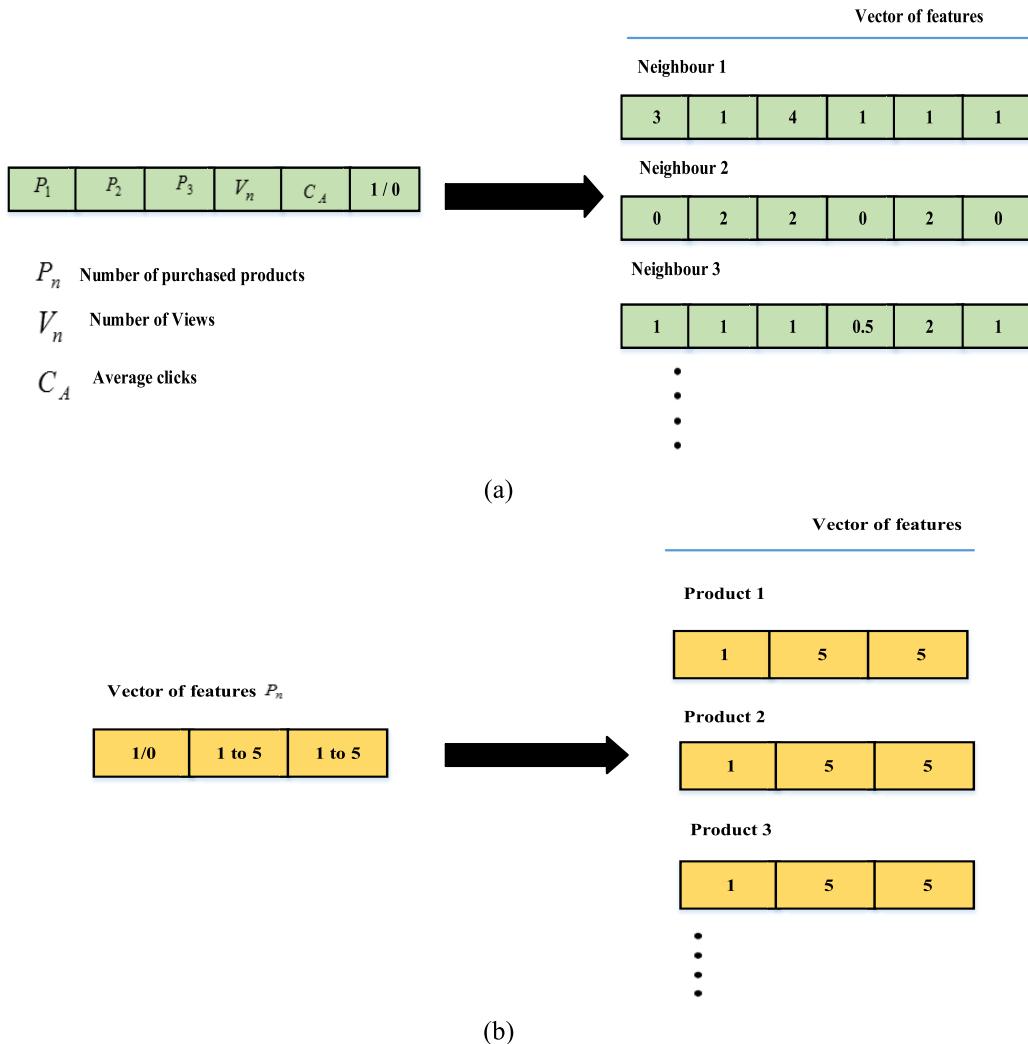


FIGURE 2. (a) and (b): Formation of vector of features to each neighbour and products.

Module 1 (User Behaviour Based Recommendation): After registration, the user behaviour form is introduced for each user. This form is used where there are no similar products and the user logs in for the first time, it is found to be a specific user.

Module 2 (Formation of the Vector of Features to Each Neighbour and Products): In this, the active user has purchased products from the list of categories like several products in electronics, dresses and so on. The formation of the vector of features consists of main steps:

- The list of products is listed in 3 columns that the neighbour vector of features concerns the number of products that each neighbour has purchased in each category.
- The fourth column recognizes the number of views (1 to views, 0 to non-views and 0.5 when it is preposterous to expect to find the new viewers).
- The fifth column is the average clicks and the last column distinguishes if a neighbour has acquired products in separate classifications from the user, which means 1 assuming true and 0 assuming false.

• Our objective with the last column is to offer a need to the neighbours who have been obtained in different classifications. In the wake of making every one of the neighbour vectors includes and normalizing the segments in a scope of 0 and 1. We ascertain the Euclidean distance among the last one of them and the user vector feature to find the closest neighbours to the user. Formation of Vector of Features to Each Neighbour and products is shown in Fig 2 (a) and (b).

When the closest Neighbours are defined, it is important to see if the ID of the products acquired is new to the active user. Although the Products identify with neighbours, for every product, we make a vector of highlights with three columns (see Figure 2(b)).

- The first column will be given the estimation of 1 if the product brand is between the user max price and min price, and 0 if not. Other than that, we can likewise observe if the individual neighbour has given a general ranking of 5, which is the most extreme estimation of a general ranking. The most 5 common behaviour data

utilized in every one of the revisions of the product are in the list of few user behaviour data that are being considered within the separate category given to the revisions of ranking 4 and 5.

- Next, the second column recognizes the general ranking given by the individual neighbour, which can be an incentive somewhere in the range of 1 and 5.
- A third column is a number somewhere in the range of 0 and 5.
- Lastly, after normalizing all the columns with an incentive range of 0 and 1 figures the Euclidean separation between the products vector of features and the vector of features P1. The framework will prescribe the number of products with the most minimal Euclidean distance esteems to the user.

The Euclidean distance (D) between the two data points x and y as,

$$D(x, y) = \sqrt{\left(\sum_{s=1}^M (x_s - y_s)^2\right)} \quad (4)$$

where x and y denote the input and feature instances.

Module 3 (Race Classifier): After forming the vector of features, we can classify the features using race classifier. The classifier is a combination of an improved Gaussian based-KNN algorithm. It can be expressed as a data analysis technique for producing models which can extract features from dataset. This classification is used to estimate the class value based on the known feature variables. A class label for each feature counts the input instances that belong to each class using this algorithm. The maximum number of considered instances is the classification output. All encountered K instances are shown to have equal divisions. The model tuples as in Eqn. (1) consider the set of n tuple training instances as $T_i = \{(X_1, L_1), (X_2, L_2) \dots (X_n, L_n)\}$. The estimated class variables for a given tuples is

$$C(x) = f(x, T_i, \text{parameters}) \quad (5)$$

where, the function $f()$ takes the arguments $\arg()$ of the parameters. These are commonly set by the user U .

For training instances, the classification performance is improved to employ a similarity method using weighted K-NN algorithm is defined as,

$$C(x) = \arg \max_{c \in \{c(1), c(2), \dots, c(m)\}} \sum_{x_i \in p(x, k)} E(c(i), c) \quad (6)$$

Also

$$E(c(i), c) = \begin{cases} 1, & \text{if } c(i) = c \\ 0, & \text{else} \end{cases} \quad (7)$$

where, c denotes the class label of each feature and then class represents the set of products that have common characteristics. $C(x)$ denotes the predicted class for a given training instances, m represents the number of classes in the data and $p(x, k)$ is the set of k - nearest neighbour of x .

Then, the probability of j^{th} class in the set of neighbourhood $p(C(j)p(x, k))$ is defined as,

$$p(C(j)p(x, k)) = \sum_{x_i \in p(x, k)} \frac{E(c(i), c)}{k} \quad (8)$$

The re-defined weighted K-NN algorithm is defined as,

$$\overline{c(x)} = \arg \max \{p(c(1)_{(x, k)}, p(c(2)_{(x, k)}, \dots, p(c(m)_{(x, k)})\} \quad (9)$$

In this process, the mean value of extracted data is computed and passed to K-NN classifier. It computes the Euclidean distance of the feature vector and all samples inside the database; Where K is set to 10. Then, it sorts the distance into ascending order. Although, it selects the closest distance to compute the Gaussian value from vector based on the distance and computes each label value score individually. The above module computes the mean value of the similarities and if the value is higher than 60%, then the algorithm recommends the user products to the target user. In this, the user behaviour information is integrated with the hybrid filtering model. On the creation of vector of features of each product and neighbours and classifies the features that have the equivalent ranking to increase same inclination. The similarity is computed as (10), shown at the bottom of this page, where $S_{target-j}$ and S_{i-j} denotes the target user ratings and a number of user ratings on product item P_{item} , $sim(U_{target}, U_i)$ denotes the similarity of U_{target} and number of users u_i . I denotes the common product items, S_{target} and S_i represents the average rating scores of all product. The computed classification is based on similarity values between the target and the existing user. The recommended level of the product p_i for the user U is computed as follows,

$$y(p)_{u_n, X, k} = \frac{\sum_{\varphi \in k} W_k \cdot \phi_{u_i, X, k}}{\sum_{\varphi \in k} W_k} \quad (11)$$

where $\phi_{u_i, X, k}$ denotes the normalized factor for the parameter of u_i , X , k .

The Complexity Analysis of Algorithm: In our recommendation algorithm, we represent the number of users and products belong to the recommender as $R \subseteq U \times P$. The similarity model used in existing approach defines that its time complexity of user computational time is $O(n)$, where n represents the number of users. Here, the analysis of time complexity of the HAR-KNN algorithm involves the similarity model that is agreed to determine each user nearest neighbours. Based on this algorithm, factors which affect the

$$Sim(U_{target}, u_i) = \frac{\sum_{P_{item} \in I} (S_{target_i} - \overline{S_{target}}) * (S_{ij} - \overline{S_1})}{\sqrt{\sum_{P_{item} \in I} (S_{target_j} - \overline{S_{target}})^2} * \sqrt{\sum_{P_{item} \in I} (S_{i-j} - \overline{S_1})^2}} \quad (10)$$

time complexity algorithm are the size of the user set u_i and product set p_i , whereas computing the similarity. Consider the set of user and product size be $|U|$ and $|P|$. The HAR-KNN chooses K nearest neighbour to compute the user's similarity as $|k|$. Thus, the time complexity to identify all the neighbours is $O(UP)$. On the others hand, the calculation of time complexity integrated with similarity measurement as $O(|U| \cdot |P|)$. In the given dataset, the maximum numbers of unrated products are predicted by HAR-KNN model. Hence, the whole time complexity for the given dataset is $O(U^2 P^2)$. This method eases the time complexity and retains the high quality of recommended products.

The pseudo code for HAR-KNN is shown as follows:

IV. EXPERIMENTS AND EVALUATION

This section explains the dataset, metrics and estimated results of the proposed RS. The effectiveness of the approach is estimated in terms of metrics such as Mean Squared Error/ Mean Square Deviation (MSE/MSD), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Precision, F1 measure, and Recall; and the correlation is made with the baseline models, such as CF-based, and CB filtering methods.

We used the Amazon product dataset. The Amazon product dataset [14] consists of 18,501 product reviews (Item, Item ID, brand name, category) and the metadata information of the purchases with the subfields such as action type (view,

add to cart, change quantity, purchase, remove), bytes, of purchase, pct_purchase, of view, Response number. The information such as Client IP, Server IP, Session ID, Rbytes, Rstat, Timestamp, and URL address are also referenced. The dataset information is collected from the Web Service of Amazon that joins with the administrative services.

A. EVALUATION METRICS

Performance measures are evaluated using metrics like MAE, RMSE, MSD, Precision F1-measure, and Recall.

MAE: This metric computes the average absolute error between actual ratings given by the user versus the predicted rating. This is one of the generally utilized accuracy metrics for assessing suggestions and the most powerful to identify anomalies. MAE ranges from 0 to infinity in which the lower the value defines the better accuracy and infinity is the maximum error on the prediction rating.

MSE: It is a quality measure that processes the average squared error between the true rating and the predicted rating. The predicted value closer to zero shows the better quality of the framework.

RMSE: It measures the root square of MSE, i.e., root mean square difference of the true and predicted ratings. The squared root around MSE translates the RMSE measure in units on the same scale. The lower RMSE value defines the better prediction accuracy produced by the RS.

Algorithm HAR-KNN Algorithm for Product Recommendation

```

Input: User,  $U \in \{u_1, u_2, \dots, u_i\}$  user behaviour matrix, product set.  $P = \{p_1, p_2, \dots, p_i\}$ 
Output: Recommendation list for  $U_{target}$ .
1.  $D_t \leftarrow$  Database is loaded.
2.  $U_k \leftarrow$  Acquire user behaviour data from the web access log database
3. User behaviour matrix ( $W_K$ )  $\leftarrow$  each user  $u_i$  // *Eq. (1)*//
4. For(Each User  $\leftarrow U$ ) {
5.   User behaviour form ( $U_k$ )  $\leftarrow U_i$  //  $U_i \rightarrow$  number of users //
6.   Sort similarities  $\leftarrow Usersim(P_i, U_{target_n})$  //  $U_{target_n} \rightarrow$  number of target user //
7. }
8.  $N_{non-zero} \leftarrow$  Number of non-zero rows in the ranking list. // *Eq. (2)* //
9. If ( $W_K \neq I_s$ ) {
10.   ReturnRanking List //  $I_s \rightarrow$  Input Sequence //
11. }
12. For (User U: Start the iteration for similarity) {
13.    $R_u \leftarrow$  Recommendation Engine has user behaviour features ( $P_n, V_n, C_A, P_n$ ). // *Eq. (3)* //
14.    $N_f \leftarrow$  Normalize the formation of the vector of features // *Eq. (4)* //
15.  $C(x) \leftarrow$  Classify class label features for  $P_n$  // *Eq. (6)* //
16.  $f(x, y) \leftarrow$  Divides the function for each user  $u_i$  // *Eq. (7)* //
17.  $\bar{C}(x) \leftarrow$  Weighted K-NN( $C(x)$ ) to employ the similarity feature
18. }
19. If( $sim(U_{target}, u_i) \leq P$ ) { // *Eq. (8)* //
20.   Return (recommended Product List)
21.   Sort List (recommended Product List) // *Eq. (9)* //
22.   Set of products recommended to the  $U_{target}$ 
23. }

```

MAE, MSE, and RMSE are estimated as follows:

$$\text{MAE} = \sum_{i=1}^n \frac{1}{n} |A_i - P_i| \quad (12)$$

$$\text{MSE} = \sum_{i=1}^n \frac{1}{n} (|A_i - P_i|)^2 \quad (13)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (|A_i - P_i|)^2} \quad (14)$$

where P , represents the user Predicted rating, A denotes the user actual rating and n denotes the total number of products in the recommendation list. A lower value of MAE, RMSE and MAE indicates a better accuracy of prediction.

Precision (P): It is defined as the fraction of the total number of recommended items from relevant recommended items for the target user. This measures to compute the values whether higher values specify better performance. The precision is defined as,

$$P = \frac{n_{rs}}{n_s} \quad (15)$$

where n_s denote the number of recommended items for the target user and n_{rs} denote the number of products that appeared in the recommended list for the target user. The higher precision value indicates better performance.

Recall (R): It states that the proportion of all relevant recommended items from a number of products for the target user. On the same way to precision, the recall measures to compute the values whether higher values specify better performance. The recall is defined as,

$$R = \frac{n_r}{n_s} \quad (16)$$

where n_r represents the number of products favoured by the target user.

F1 measure: It is a function of recall and precision, the formula can be defined as,

$$F1 = \frac{2 * P * R}{P + R} \quad (17)$$

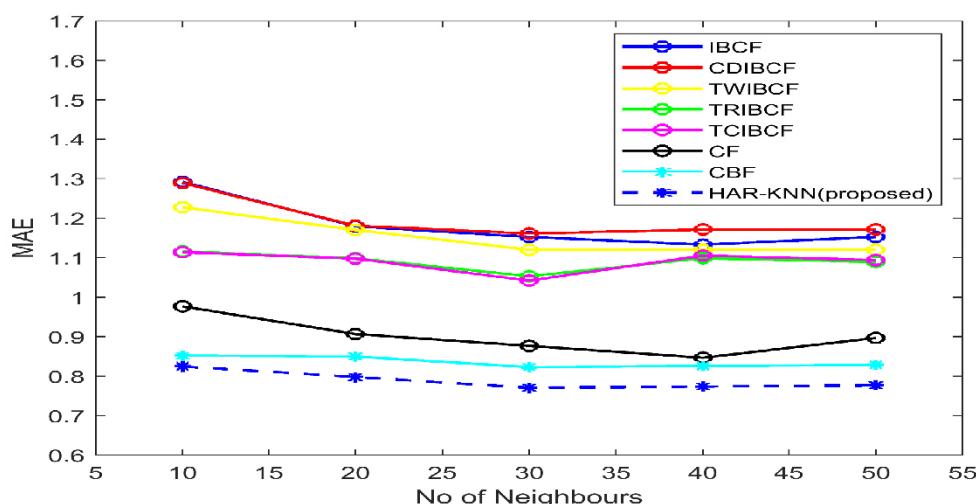


FIGURE 3. The comparison of MAE measures for various methods.

B. EXPERIMENTAL RESULT AND COMPARISON

The experimental performance is performed with an Amazon dataset [14]. The selected methods are compared with the HAR-KNN as follows:

1) COLLABORATIVE FILTERING (CF-BASED)

In this methodology, the proposal is resolved dependent on the collaborative method that uses the activities and evaluations of the user and network. This will in general gauge the user with comparable enthusiasm to foresee the comparative items. Additionally, the models such as the item-based CF (IBCF), IBCF using time-aware similarity computation (TRIBCF) and coverage degree based to represent the IBCF(CDIBCF) using coverage based rating prediction, Time weight IBCF (TWIBCF) and Time-related correlation degree (TCIBCF) are utilized to compare the proposed recommendation framework.

2) CONTENT-BASED (CB) FILTERING

This type of approach uses attributes such as user feedback and actions to estimate the recommendation of the item that is liked by the user before.

Fig 3. Represents the MAE result of the HAR-KNN compared to the existing techniques for the Amazon dataset. In this, for the increasing size of neighbours, the MAE value gets lower and then get increased prediction result. The performance of the TCIBCF, TWIBCF has a similar prediction, however, it achieves the improved results in comparison to CDIBCF, IBCF techniques. Lower the MAE value, better prediction measure. As seen, the CF and CBF enhance prediction accuracy whereas it achieves a lower error than the other methods. From all of the above results, we can assert that the proposed system improves prediction accuracy better than the other traditional and earlier approaches.

Fig 4. Shows the RMSE result of the HAR-KNN system with the existing techniques in terms of the Amazon dataset.

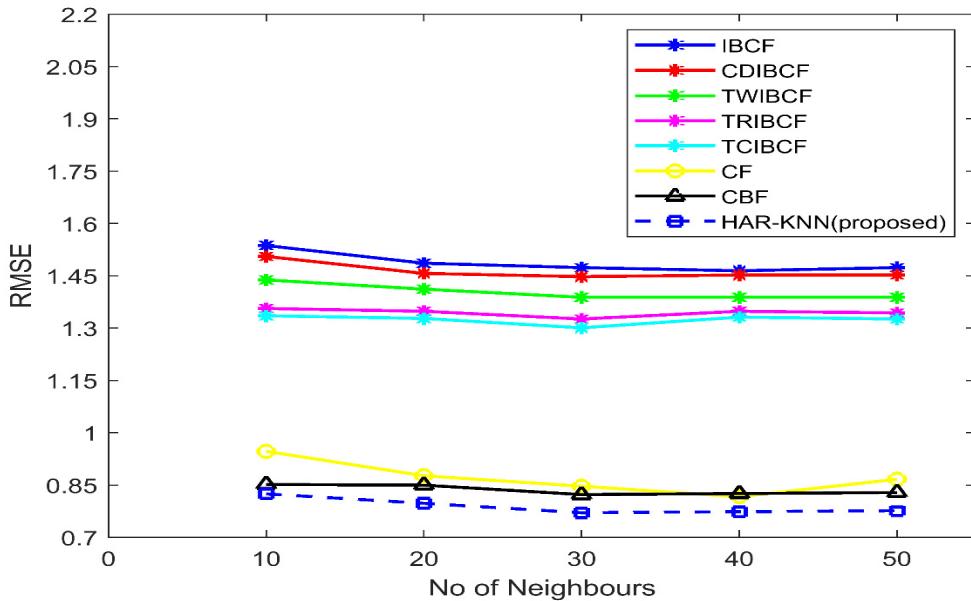


FIGURE 4. The comparison of RMSE measures for various methods.

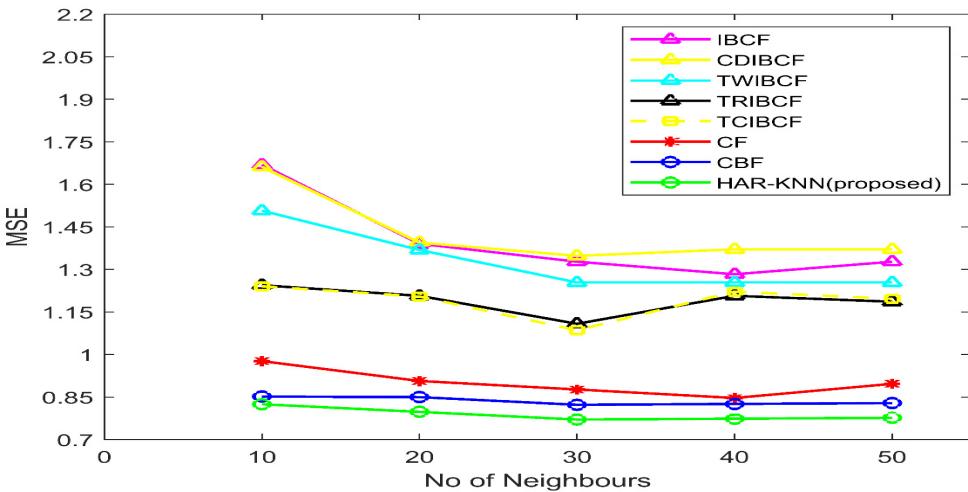


FIGURE 5. The comparison of MSE measures for various methods.

Initially, the error value gets decreased due to the higher value of neighbours and then increases. Lower the RMSE measure, the higher the prediction measure. Therefore, our proposed HAR-KNN approach achieves lower error value and better accuracy compared to the existing recommender system.

Fig 5. Illustrates the MSE value of the proposed RS in terms of the Amazon dataset. In this, the mean square error value for the HAR-KNN achieves better prediction with the

lower deviation value than the earlier models. The above results can improve prediction accuracy as the values in the computed measures are better than that of existing methods.

In Fig 6. Shows the precision value achieved with the Amazon dataset. In this, the precision values are computed for those existing methods drops in which the neighbourhood size increases. Moreover, the values of IBCF, TRIBCF

and TWIBCF methods perform the same compared to CF, CBF, and HAR-KNN. This is because the precision values of HAR-KNN have higher values than existing approaches.

In Fig 7. Shows the Recall value achieved with the Amazon dataset [14]. This shows that the TWIBCF, TRIBCF approaches are closely the same as the traditional IBCF methods. This is because the recall value of TCIBCF, CDIBCF, CF, and CBF increases faster than IBCF methods. Furthermore, the HAR-KNN method improves largely as the neighbourhood increases. Hence, HAR-KNN and TCIBCF improve the recall measures compared to traditional methods.

Fig 8 shows the F1 value achieved using the Amazon dataset. This depicts that the neighbourhood size increases as the F1 value increases. The list of recommendations with the neighbourhood size increases as the F1 value of HAR-KNN

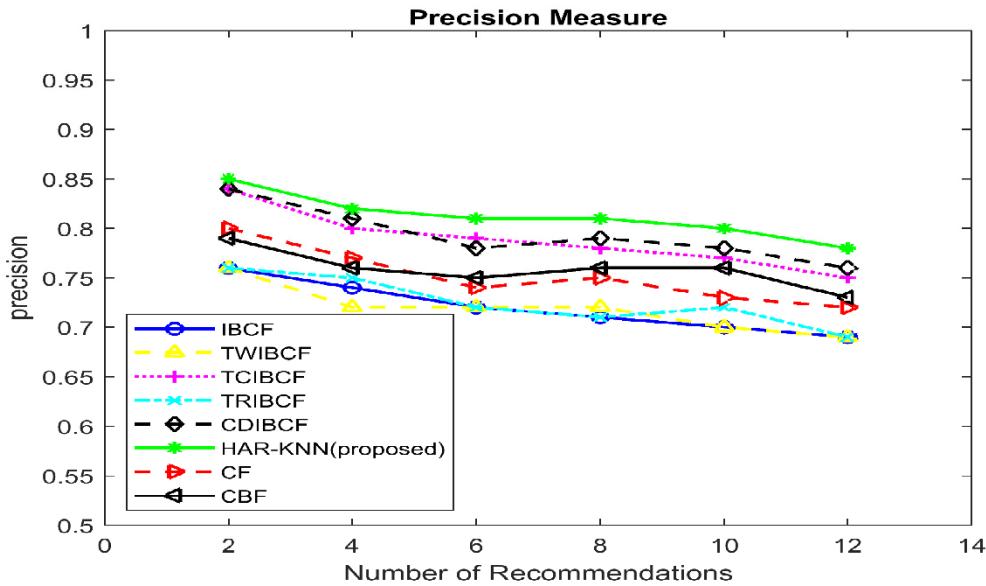


FIGURE 6. The comparison of precision measures for various methods.

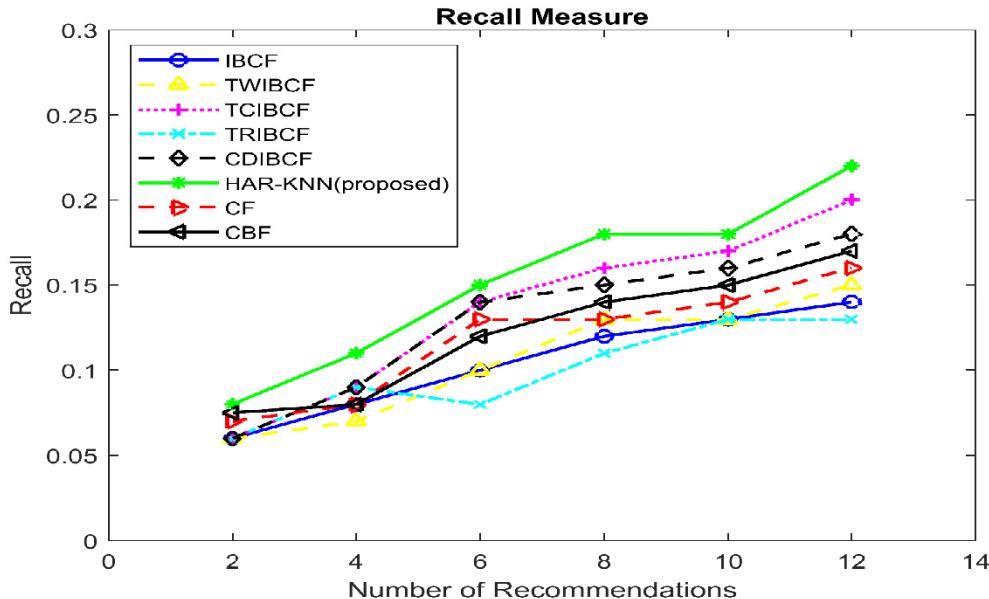


FIGURE 7. The comparison of Recall measures for various methods.

increases than that of existing TCIBCF, CDIBCF, TRIBCF, TWIBCF, CF, CBF, and IBCF methods.

In Table 2., from the simulation results, MAE, RMSE and MSE measures analyzed that the HAR-KNN is having less error rate as the number of the neighbourhood [10 to 50] increases compared to existing methods. The bold represents the low error value as the size of the neighbourhood increases.

Table 3. From the simulation result of Precision, Recall and F1 measures analyzed that the HAR-KNN is having higher values compared to existing methods. The bold represents higher values as the set of recommendation lists varies from [2]–[12].

In single attribute criteria, the collected data would not have the overall rating; this may lead to inaccurate visions on the true similarity among user preferences. But in the multi-attribute criteria, the user behaviour data captures better information which allows discovering connections between users and products. We use the user behaviour matrix table linked with HAR-KNN approach to evaluate the existing recommendation to test all approaches.

Our simulation uses different performance metrics to show how much added value an RS be which is referred to the average difference between the right value and the target user shown by the RMSE/MSE/MAE. Most real e-commerce data

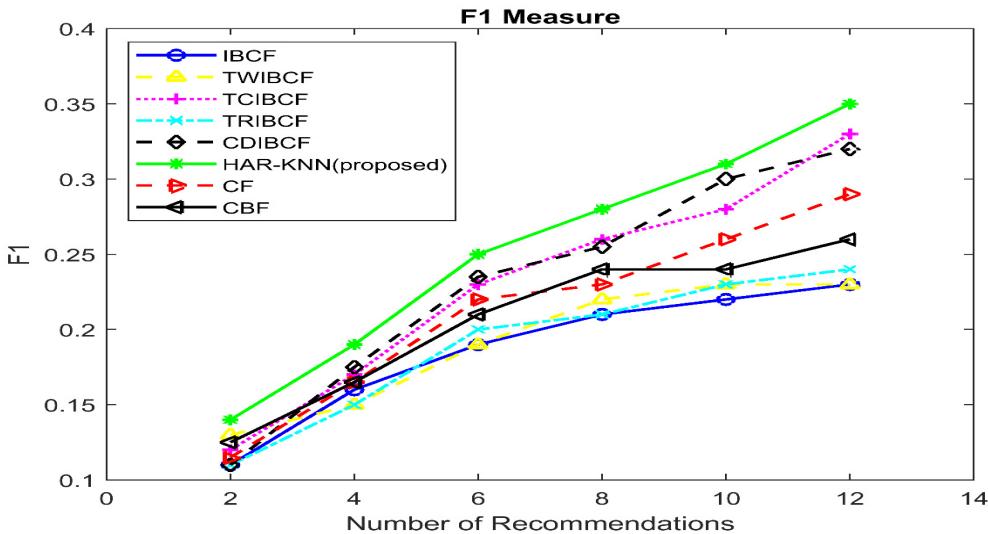


FIGURE 8. The comparison of F1- measures for various methods.

TABLE 2. Comparison of performance measures related to other existing methods.

S/ No	Parameter	Number of Neighbours	IBCF	CDIBCF	TWIBCF	TRIBCF	TCIBCF	CF	CBF	HAR-KNN
1.	MAE	10	1.2919	1.2890	1.2276	1.1154	1.1137	0.9765	0.7685	0.7165
		20	1.1791	1.1809	1.1698	1.0985	1.0974	0.9065	0.7955	0.7435
		30	1.1520	1.1609	1.1198	1.0527	1.0418	0.8765	0.8225	0.7705
		40	1.1326	1.1709	1.1197	1.0985	1.1052	0.8465	0.8495	0.7975
		50	1.1520	1.1708	1.1198	1.0890	1.0940	0.8965	0.8515	0.8245
2.	RMSE	10	1.5366	1.5053	1.4380	1.3561	1.3353	0.9465	0.7238	0.7578
		20	1.4859	1.4567	1.4116	1.3481	1.3276	0.8765	0.7365	0.7812
		30	1.4733	1.4474	1.3882	1.3260	1.3007	0.8465	0.8546	0.7322
		40	1.4642	1.4521	1.3881	1.3481	1.3313	0.8165	0.8765	0.7465
		50	1.4733	1.4561	1.3882	1.3436	1.3259	0.8665	0.8236	0.8201
3.	MSE	10	1.6690	1.6615	1.5071	1.2441	1.2403	0.9765	0.7543	0.7732
		20	1.3903	1.3944	1.3685	1.2068	1.2042	0.9065	0.7122	0.7664
		30	1.3271	1.3476	1.2540	1.1082	1.0854	0.8765	0.8987	0.7201
		40	1.2828	1.3709	1.2541	1.2068	1.2215	0.8465	0.8754	0.7388
		50	1.3271	1.3708	1.2510	1.1859	1.1967	0.8965	0.8193	0.8843

will use small dataset to run the RS to see the matched correct answers in the data. Then, construct a list of selected top-N recommendation which includes the product and user using precision/recall/f1. Among all these metrics, our experiment results show the importance of multi-attribute behaviour data which is very useful to predict a list of recommended product to a target user

V. CASE STUDY

To solve the issues of data sparsity and loss of neighbour activity, the HAR-KNN recommender system was developed to employ the user needs of suitable products. The recent study shows that RS is the main part of an E-Commerce site and is valuable for increasing buyers and for constructing user loyalty.

An Amazon dataset [14] was extracted from the system to relate the domain characteristics of E-Commerce RS utilized as a case study to further apply the feasibility of the proposed HAR-KNN approach to a real-time application. The dataset incorporates a large number of users and products.

In our experiment, we randomly select products to utilize the user behaviour matrix, rating orders, classification of feature ordering conditions to construct the training and testing phases. To check the performance of the RS, we divide the dataset into 80% training and the rest 20% as the testing set for verification.

We set the maximum length of the document to 300. To achieve better accuracy, few combinations are taken into account the nearest neighbours (K). The K values lie between 10 to 50. Generally, higher values of K reduce the effect

TABLE 3. Comparison of performance measures related to other existing methods.

S/No	Parameter	No of Recommendation	IBCF	CDIBCF	TWIBCF	TRIBCF	TCIBCF	CF	CBF	HAR-KNN
1.	Precision (P)	2	0.7611	0.8404	0.7601	0.7601	0.8403	0.8	0.7903	0.8502
		4	0.7421	0.8103	0.7206	0.7502	0.8002	0.7703	0.7602	0.8203
		6	0.7201	0.7802	0.7202	0.7206	0.7907	0.7404	0.7506	0.8102
		8	0.7105	0.7905	0.7206	0.7103	0.7809	0.7501	0.7609	0.8107
		10	0.7005	0.7808	0.7009	0.7212	0.7701	0.7308	0.7602	0.8004
		12	0.6907	0.7607	0.6902	0.6911	0.7502	0.7206	0.7301	0.7802
2.	Recall (R)	2	0.0602	0.0602	0.0602	0.0603	0.0604	0.0700	0.0750	0.0807
		4	0.0804	0.0906	0.0704	0.0910	0.0906	0.0800	0.0840	0.1104
		6	0.1005	0.1402	0.1005	0.0804	0.1407	0.1300	0.1203	0.1503
		8	0.1201	0.1505	0.1311	0.1101	0.1602	0.1300	0.1402	0.1810
		10	0.1303	0.1602	0.1320	0.1303	0.1705	0.1400	0.1501	0.1807
		12	0.1403	0.1806	0.1502	0.1306	0.2001	0.1600	0.1700	0.2201
3.	F1	2	0.1101	0.1106	0.1307	0.1108	0.1204	0.1150	0.1250	0.1409
		4	0.1602	0.1752	0.1504	0.1501	0.1702	0.1650	0.1650	0.1907
		6	0.1901	0.2350	0.1905	0.2001	0.2303	0.2203	0.2104	0.2502
		8	0.2107	0.2550	0.2203	0.2108	0.2608	0.2307	0.2403	0.2800
		10	0.2203	0.3005	0.2308	0.2304	0.2808	0.2600	0.2407	0.3101
		12	0.2301	0.3210	0.2302	0.2403	0.3301	0.2901	0.2604	0.3507

of noise in classification. With cross-validation, the nearest neighbour selects the value with less classification error. The new data instances with the training data can be measured by Euclidean distance. With various subsets of k, the process of K is repeated until it reaches a certain number of values. We can evaluate the error rate using MAE, MSE, and RMSE.

In simulation results, the evaluation metrics indicate the changes in the target item as the size of the neighbourhood increases. Compared to the relevant approaches, the obtained results are hard to perform in CF and CB filtering in a small size of nearest neighbours. Therefore, the predicted rating has been accomplished based on 10 to 50 nearest neighbours. At this point, the selected neighbourhood products of the target product are more consistent. Hence, the selected neighbourhood by HAR-KNN is similar to achieving the target products by the traditional methods. Moreover, the accuracy of the recall, F1 measure, and precision are computed to rate the relevant products and number of recommendations that were set to [2, 4, 6, 8, 10, 12]. Therefore, the HAR-KNN provides better RS with better predictive accuracy than existing methods for the Amazon dataset.

VI. CONCLUSION

In this paper, the proposed HAR-KNN model is based on purchase histories and user behaviour data like the number of views, clicks and purchased products. The proposed method constructs the user behaviour matrix and pre-processes the frequency of product purchased to predict the users using the vector of features for each neighbour and product.

The paper makes proper utilization after constructing the user behaviour matrix utilized to map out the other users with similar tastes and neighbours. It is distinguished by calculating the Euclidean distance and proper improvisation of all the algorithms used in the framework to achieve a similarity level more accurately. Moreover, the HAR-KNN model is validated on the Amazon dataset and effectively compared with IBCF, TRIBCF, TCIBCF, CDIBCF, TWIBCF, CB and CF in terms of MAE, RMSE, MSE, Recall, Precision, and F1-measure values. The experimental result depicts that the performance measures improve high predictive accuracy with low error than traditional methods.

Based on present data, the access of user behaviour matrix depends upon the preference of user behaviour. Many applications use two dimensions (i.e. users and products). This may not be sufficient as customer preferences may largely increase in different factors. In the future, there is a need to incorporate the concept of multi-dimensionality in RS.

REFERENCES

- [1] A. Davoudi and M. Chatterjee, "Social trust model for rating prediction in recommender systems: Effects of similarity, centrality, and social ties," *Online Social Netw. Media*, vol. 7, pp. 1–11, Sep. 2018.
- [2] B. B. Sinha and R. Dhanalakshmi, "Evolution of recommender system over the time," *Soft Comput.*, vol. 23, no. 23, pp. 12169–12188, Dec. 2019.
- [3] B. C. L. Christudas, E. Kirubakaran, and P. R. J. Thangaiah, "An evolutionary approach for personalization of content delivery in e-learning systems based on learner behavior forcing compatibility of learning materials," *Telematics Informat.*, vol. 35, no. 3, pp. 520–533, Jun. 2018.
- [4] A. Alazab, S. Bevinakoppa, and A. Khraisat, "Maximising competitive advantage on E-business websites: A data mining approach," in *Proc. IEEE Conf. Big Data Analytics (ICBDA)*, Nov. 2018, pp. 111–116.

- [5] J. K. Tarus, Z. Niu, and G. Mustafa, "Knowledge-based recommendation: A review of ontology-based recommender systems for e-learning," *Artif. Intell. Rev.*, vol. 50, no. 1, pp. 21–48, Jun. 2018.
- [6] S. Setiowati, T. B. Adjii, and I. Ardiyanto, "Context-based awareness in location recommendation system to enhance recommendation quality: A review," in *Proc. Int. Conf. Inf. Commun. Technol. (ICOIACT)*, Mar. 2018, pp. 90–95.
- [7] R. Chen, Q. Hua, Y.-S. Chang, B. Wang, L. Zhang, and X. Kong, "A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks," *IEEE Access*, vol. 6, pp. 64301–64320, 2018.
- [8] I. Schoinas and C. Tjortjis, "MuSIF: A product recommendation system based on multi-source implicit feedback," in *Proc. InIFIP Int. Conf. Artif. Intell. Appl. Innov.* Cham, Switzerland: Springer, May 2019, pp. 660–672.
- [9] G. George and A. M. Lal, "Review of ontology-based recommender systems in e-learning," *Comput. Educ.*, vol. 142, Dec. 2019, Art. no. 103642.
- [10] J. K. Tarus, Z. Niu, and D. Kalui, "A hybrid recommender system for e-learning based on context awareness and sequential pattern mining," *Soft Comput.*, vol. 22, no. 8, pp. 2449–2461, Apr. 2018.
- [11] M. Nilashi, O. Ibrahim, and K. Bagherifard, "A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques," *Expert Syst. Appl.*, vol. 92, pp. 507–520, Feb. 2018.
- [12] X. Cao and J. Sun, "Exploring the effect of overload on the discontinuous intention of social media users: An S-O-R perspective," *Comput. Hum. Behav.*, vol. 81, pp. 10–18, Apr. 2018.
- [13] S. Gupta, A. K. Kar, A. Baabdullah, and W. A. A. Al-Khowaiter, "Big data with cognitive computing: A review for the future," *Int. J. Inf. Manage.*, vol. 42, pp. 78–89, Oct. 2018.
- [14] J. Lv, B. Song, J. Guo, X. Du, and M. Guizani, "Interest-related item similarity model based on multimodal data for top-N recommendation," *IEEE Access*, vol. 7, pp. 12809–12821, 2019.
- [15] J. A. Pereira, P. Matuszyk, S. Krieter, M. Spiliopoulou, and G. Saake, "Personalized recommender systems for product-line configuration processes," *Comput. Lang., Syst. Struct.*, vol. 54, pp. 451–471, Dec. 2018.
- [16] H. Hwangbo, Y. S. Kim, and K. J. Cha, "Recommendation system development for fashion retail e-commerce," *Electron. Commerce Res. Appl.*, vol. 28, pp. 94–101, Mar. 2018.
- [17] S. Bandyopadhyay, S. S. Thakur, and J. K. Mandal, "A novel approach for product prediction using artificial neural networks," in *Proc. Annu. Conv. Comput. Soc. India*. Singapore: Springer, Jan. 2018, pp. 682–691.
- [18] J. Chen and H. Wang, "Guest editorial: Big data infrastructure II," *IEEE Trans. Big Data*, vol. 4, no. 3, pp. 299–300, Sep. 2018.
- [19] Z. Ji, H. Pi, W. Wei, B. Xiong, M. Wozniak, and R. Damasevicius, "Recommendation based on review texts and social communities: A hybrid model," *IEEE Access*, vol. 7, pp. 40416–40427, 2019.
- [20] Z. Yang, Q. Sun, and B. Zhang, "Evaluating prediction error for anomaly detection by exploiting matrix factorization in rating systems," *IEEE Access*, vol. 6, pp. 50014–50029, 2018.



SUNKURU GOPAL KRISHNA PATRO is currently a Ph.D. Research Scholar with GIET University, Gunupur, Odisha. He is having 740 six years of experience in both academic as well 741 as industry. He has published some international research journals articles. He is currently an Assistant Professor with the Department of CSE, GIET University. His area of research interest is the IoT and block chain, recommended system, and data mining.



BROJO KISHORE MISHRA received the Ph.D. degree in computer science from the Berhampur University, in 2012. He is currently working as a Professor with the Computer Science and Engineering Department, GIET University, Gunupur, India. He has published more than 30 research articles in national and international conference proceedings, 25 research articles in peer-reviewed journals, 22 book chapters, authored two books, and edited three books. His research interests include data mining and big data analysis, machine learning, soft computing, and evolutionary computation.



SANJAYA KUMAR PANDA received the Ph.D. degree from IIT (ISM) Dhanbad, Dhanbad, India, the B.Tech. degree in CSE from VSSUT, Burla, India, and the M.Tech. degree from NIT, Rourkela, India. He worked as an Assistant Professor and the Head of the Department, CSE, IIITDM Kurnool, Kurnool, India, and also as an Assistant Professor with the Department of IT, VSSUT, Burla. He has published more than 60 articles in reputed journals and conferences. His current research interests include recommender system, cloud computing, big data analytics, grid computing, fault tolerance, and load balancing. He has delivered several invited talks, chair sessions in many national and international conferences and workshops. He is currently working as an Assistant Professor with the Department of CSE, NIT Warangal, Warangal, India. He is an Associate Member of IEI, a Life Member of ISTE, a Life Member of CSI, IAENG, IACSIT, UACEE, ACEEE, and SDIWC. He received two Silver Medal Awards for Best Graduate, the Best Post-Graduate in CSE, the Institution Award, the IEEE Brand Ambassador 2019 and 2018, the SGSITS National Award for the Best Research Work by Young Teachers of Engineering College for the year 2017, the Faculty With Maximum Publishing in CSI Publications Award, the Young IT Professional Award in 2017 and 2016, the Young Scientist Award, the CSI Paper Presenter Award at International Conference in 2019 and 2015, and the CSI Distinguished Speaker Award. He acted as a reviewer in many reputed journals, including the IEEE TRANSACTION ON SYSTEMS, MAN AND CYBERNETICS, the IEEE/CAA Journal of Automatic Sinica, the IEEE Sensors Letters, Journal of Parallel and Distributed Computing, Elsevier, the Journal of Supercomputing, Springer, Arabian Journal for Science and Engineering, Springer, Science China Information Sciences, Mathematical Problems in Engineering, Hindawi, Applied Soft Computing, Elsevier, Ain Shams Engineering Journal etc. and conferences including GCC, INDICON, ICACCI, ICCVIA, PDGC, INDIACom, ISSPIT, MobiApps, MobiWIS, EmergiTech, PECON, WSCAR, ISGT, TCGC, WCI, SPICES and ISBEIA. He also acted as a Guest Editor in many international journals including IJITWE, IJRSDA, and IJKDB.



RAGHVENDRA KUMAR received the B.Tech., M.Tech., and Ph.D. degrees in computer science and engineering, from the Institute of Information Technology, Virtual Reality and Multimedia, Vietnam. He was a Postdoc Fellow with Institute of Information Technology. He is currently working as an Associate Professor with the Computer Science and Engineering Department, GIET University, India. He serves as a Series Editor of *Internet of Everything (IOE): Security and Privacy Paradigm, Green Engineering and Technology: Concepts and Applications*, published by CRC press, Taylor & Francis Group, USA, and *Bio-Medical Engineering: Techniques and Applications*, Published by Apple Academic Press, CRC Press, Taylor & Francis Group, USA. He also serves as an Acquisition Editor for *Computer Science* by Apple Academic Press, CRC Press, Taylor & Francis Group, USA. He has published number of research articles in international journals (SCI/SCIE/ESCI/Scopus) and conferences including the IEEE and Springer as well as serve as the Organizing Chair (RICE-2019, 2020), a Volume Editor (RICE-2018), a Keynote Speaker, the Session Chair, the Co-Chair, the Publicity Chair, the Publication Chair, an Advisory Board, the Technical Program Committee Members in many international and national conferences and serve as a Guest Editor in many special issues from reputed journals (Indexed By: Scopus, ESCI, SCI). He also published 13 chapters in edited book published by IGI Global, Springer, and Elsevier. His research areas are computer networks, data mining, cloud computing and secure multiparty computations, theory of computer science, and design of algorithms. He has authored and edited 23 computer science books in field of the Internet of Things, data mining, biomedical engineering, big data, robotics, and igi global publication, USA, IOS Press Netherland, Springer, Elsevier, CRC Press, USA.



HOANG VIET LONG received the Ph.D. degree in computer science from the Hanoi University of Science and Technology, in 2011, where he defended his thesis in the fuzzy and soft computing field. He has been promoted to an Associate Professor of information technology since 2017. Recently, he has been concerning in cybersecurity, machine learning, bitcoin, and blockchain and published more than 40 articles in ISI-covered journal.



ISHAANI PRIYADARSHINI received the bachelor's degree in computer science engineering and the master's degree in information security from the Kalinga Institute of Industrial Technology, India, and the master's degree in cybersecurity from the University of Delaware. She is currently pursuing the Ph.D. degree with the University of Delaware, USA. She has authored several book chapters for reputed publishers. She is an Author to several publications for SCIE indexed journals.

As a certified Reviewer, she conducts peer review of research articles for prestigious IEEE, Elsevier, and Springer journals and is a part of the Editorial Board for the *International Journal of Information Security and Privacy (IJISP)*. Her areas of research include cybersecurity, artificial intelligence, and HCI.



DAVID TANIAK received the bachelor's, master's, and Ph.D. degrees all in computer science, specializing in databases. He is currently an Associate Professor with Monash University, Australia. He has also published over 200 journal articles. His research areas are in big data processing, data warehousing, and mobile and spatial query processing. He has published a book on the *High-Performance Parallel Database Processing* (Wiley, 2008). He is a Regular Keynote Speaker at an international conference, delivering lectures and speeches on big data. He is a founding Editor-in-Chief of the International Journal of Web and Grid Services and the International Journal of Data Warehousing and Mining.

• • •