IWENDI, C., IBEKE, E., EGGONI, H., VELAGALA, S. and SRIVASTAVA, G. 2022. Pointer-based item-to-item collaborative filtering recommendation system using a machine learning model. *International journal of information technology and decision making* [online], 21(1), pages 463-484. Available from: https://doi.org/10.1142/S0219622021500619

# Pointer-based item-to-item collaborative filtering recommendation system using a machine learning model.

IWENDI, C., IBEKE, E., EGGONI, H., VELAGALA, S. and SRIVASTAVA, G.

2022

*Electronic version of an article published as International journal of information technology and decision making, 21(1), pages 463-484, https://doi.org/10.1142/S0219622021500619. © copyright World Scientific Publishing Company. https://www.worldscientific.com/doi/10.1142/S0219622021500619*

2

This document was downloaded from https://openair.rgu.ac.uk

# POINTER-BASED ITEM-TO-ITEM COLLABORATIVE FILTERING RECOMMENDATION SYSTEM USING A MACHINE LEARNING MODEL

CELESTINE IWENDI

*School of Creative Technologies, University of Bolton, United Kingdom*
*celestine.iwendi@ieee.org*

EBUKA IBEKE

*School of Creative and Cultural Business, Robert Gordon University, Aberdeen, Scotland, UK*
*e.ibeke@rgu.ac.uk*

HARSHINI EGGONI

*Department of Computer Science and Engineering. Nagarjuna College of Engineering and Technology, Bangalore 560300. India*
*harshinieggoni@gmail.com*

SREERAJAVENKATAREDDY VELAGALA

*Department of Computer Science and Engineering. Nagarjuna College of Engineering and Technology, Bangalore 560300. India*
*vsrvenkatareddy@gmail.com*

GAUTAM SRIVASTAVA

*Department of Math and Computer Science, Brandon University, MB, Canada.*
*Research Centre for Interneural Computing, China Medical University, Taiching, Taiwan*
*Corresponding Author: srivastavag@brandonu.ca*

The advent of digital marketing has enabled companies to adopt personalized item recommendations for their customers. This process keeps them ahead of the competition. One of the techniques used in item recommendation is known as an item-based recommendation system or item-item collaborative filtering. Presently, item recommendation is based completely on ratings like $1-5$, which is not included in the comment section. In this context, users or customers express their feelings and thoughts about products or services. This paper proposes a machine learning model system where 0, 2, 4 are used to rate products. 0 is negative, 2 is neutral, 4 is positive. This will be in addition to the existing review system that takes care of the users' reviews and comments, without disrupting it. We have implemented this model by using Keras, Pandas and Sci-kit Learning libraries to run the internal work. The proposed approach improved prediction with 79% Accuracy for Yelp datasets of businesses across 11 metropolitan areas in four countries, along with a Mean Absolute Error (MAE) of 21%, Precision at 79%, Recall at 80% and F1-Score at 79%. Our model shows scalability advantage and how organizations can revolutionize their recommender systems to attract possible customers and increase patronage. Also, the proposed similarity algorithm was compared to conventional algorithms to estimate its performance and accuracy in terms of its Root Mean Square Error (RSME), Precision and Recall. Results of this experiment indicate that the similarity recommendation algorithm performs better than the conventional algorithm and enhances recommendation accuracy.

*Keywords*: recommender systems; Keyword-Item recommendation, machine learning, collaborative filtering, rating

## 1. Introduction

Recommendation systems have contributed immensely to meeting the needs of consumers and promoting sales through advertisement on social platforms such as YouTube, Amazon, Facebook and many others. The classification technique of recommendation systems has not fully considered all relevant reviews and sub-reviews. In other words, they failed to consider the interaction between matching reviews and comments. The authors in [1] previously proposed this word-level interaction between the users and the producers to reduce the problems faced by the users. This differs from the traditional models where questions and relationships are represented by a sequence of vectors [2–4]. The authors performed a comparison on lower-level interactions that relieves the information not recorded when the sequence is merged into a direct dimensional vector. The fact is that the recommendation has to be based on user and item reviews, and a machine learning model instead of deep learning model predictions. The objective of this research is to introduce an attention mechanism, which retrieves essential reviews by placing dynamic attention on the item reviews, hence improving the scalability of additional reviews for a better recommendation. Presently, item recommendation is fully based on ratings like $1-5$, which is not included in the comment section where the users or customers express their feelings and thoughts about products and services [5,6]. Therefore, the classification technique of this approach, compared to other models, is projected as the only one which fully considers relevant reviews as well as relevant sub-reviews and comments of the users which play a major role in a recommendation. This paper's contributions include:

- A proposal for a modern machine learning attention model for recommendation with reviews including comments by the users. The proposed model makes use of a new learning scheme that is pointer-based. It then generates temporary table mapping from the purchased item to the customer after it retrieves all the ratings for each item sorted via a priority list. After passing all the items through the filters and truncating the other items, a recommendation of the relevant item is made. This recommendation is based on the user ratings and comments.
- Training of the dataset, using the Word2Vec algorithm, where it divides the comments into three categories known as positive, neutral and negative into ratings of 4, 2, and 0. It then adds to the existing system item recommendations based on ratings from 1-5 on the Yelp datasets. The baselines


Fig. 1: Recommendation Scenario

used in this study are very fierce, incorporating review-based models as well as modern interaction-only models.

- This research improved prediction with 79% Accuracy on the Yelp dataset of businesses across 11 metropolitan areas in four countries, along with a Mean Absolute Error of 21%, Precision at 79%, Recall at 80% and F1-Score of 79%. The model shows scalability advantage and how organizations can revolutionize their recommender systems to attract possible customers and increase patronage.

The paper is organized as such: Section 2 describes the recommender related works, Section 3 explains the methodology used, in Section 4, we present and discuss the results and Section 5 gives the conclusion drawn from the work.

## 2. Related Work

With the advent and rise of the use of web services, Recommender systems (RS) continue to play a significant role in daily online activities [7]. From a vast number of items and/or services, RS predicts personalized items or services to users [8]. The study of RS is an interesting and trendy area of study and has been researched in many fields. Some of these fields include human-computer interaction [9], machine learning [10, 11], statistics [12], artificial networks [13], calculative trust [14–18], argumentation [19], among others. Recommender systems also have many application areas such as medicine. [20] developed a machine learning model for recommending patient diet; [21] proposed a recommendation system that assists students' course selection, [22] on the other hand developed a method that recommends an e-learning system based on personal learning style. Another application area is aviation. [23] introduced a model for recommending airline itinerary, as well as and movie rec-

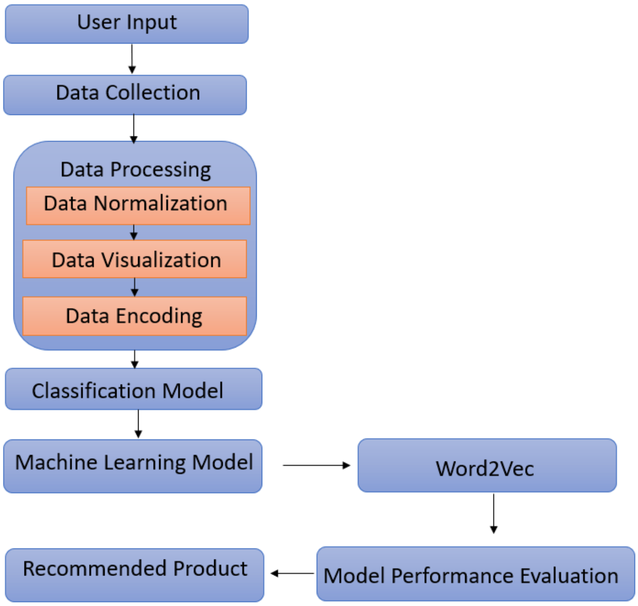


Fig. 2: Recommendation Methodology Flow

ommendation [24–26]. Early RS are based on a filtering algorithm with the first filtering system known as Tapestry [27]. The tapestry was used to filter email messages relating to certain queries or comments. Over the years, filtering systems have seen many improvements and can perform RS effectively and efficiently [28].

There are three major paradigms of RS namely content-based method, collaborative filtering method, and hybrid method; among which collaborative filtering method has been noted to be more efficient [28]. The collaborative filtering (CF) method is additionally categorized into model-based and memory-based methods. The Model-based CF method utilizes user ratings to make recommendations [29]. The model-based approach employs clustering and classification techniques. It compresses a huge database into a single model from which the recommendation task is performed [30], whereas the memory-based method generates predictions by utilizing the entire database. This method is also referred to as nearest-neighbour CF as it uses statistical methods to cluster users with related transaction history to an active user [29]. The memory-based method is further classified into item-based and user-based approaches. [29] noted that the item-based approach outperforms the user-based method where the quantity of items is high and the number of ratings is low.

The item-based method has been researched in various forms. It has been noted that they are more scalable than the user-based method and less affected by the cold-start issue. An item-based top$N$ recommendation system was proposed by [31] using a pre-computed model. This model at the first stage calculates the similarities between each item and then identifies a set or sets of an item for the recommendation. [32] presented a dynamic item-based topN RS model which provides a recommendation while considering time decay. The recommendation results in an item-based topN recommendation system are computed based on the correlation of items among all users. However, these systems suffer from frequent malicious user attacks since they are mostly based on aggregation information. As a result, [33] proposed a supervised randomization technique that shields item-based top$N$ recommendation systems from such attacks. More recently, [34] while taking into account users' privacy, proposed a privacy-preserving item-based CF model. An item-based social recommender system was proposed in [35] using item asymmetric correlation. In [36] the inter- and intra-transaction dependencies were jointly modelled for next-item recommendations. The item-based method has been researched in various forms. It has been noted that they are more scalable than the user-based method and less affected by the cold-start issue. An item-based topN recommendation system was proposed by [31] using a pre-computed model. This model at the first stage calculates the similarities between each item and then identifies a set or sets of an item for the recommendation. [32] presented a dynamic item-based topN RS model which provides a recommendation while considering time decay. The recommendation results in an item-based topN recommendation system are computed based on the correlation of items among all users. However, these systems suffer from frequent malicious user attacks since they are mostly based on aggregation information. As a result, [33] proposed a supervised randomization technique that shields item-based topN recommendation systems from such attacks. More recently, [34] while taking into account users' privacy, proposed a privacy-preserving item-based CF model. An item-based social recommender system was proposed in [35] using item asymmetric correlation. In [36] the inter- and intra-transaction dependencies were jointly modelled for the next-item recommendation.

The main factors affecting RS are the issues of cold-start and data sparsity. Many researchers have considered the relationship between users on a social network as a solution to these factors. But actual relationships on a social network are themselves sparse and this affects the performance of RS. The trust propagation approach is then introduced to compute both direct and indirect relationships. Trust is classified into implicit and explicit trust. Some studies have incorporated both implicit and explicit trusts into their RS model [37]. The implicit trust focuses on the degree of similarity between users or items and are computed either by integrating a social network trust [16, 38], applying weighted similarity measures [39, 40], incorporating fuzzy logic [41, 42] or using probabilistic methods [43] in the RS. Whereas in an explicit trust, trust values are expressed by the users in binary format. The two users either trust themselves or don't. Explicit trust has been implemented in several studies such as [44, 45].

In the proposed model, we place dynamic attention on item reviews while also considering relevant sub-reviews and comments [46]. This is against most conventional models based solely on user rating which do not account for users' opinions and feelings.

## 3. Methodology

The objective of this paper is to introduce an attention mechanism, that retrieves essential reviews by placing dynamic attention on the item reviews, hence improving the scalability of additional reviews for a better recommendation. Presently, Amazon is one of the companies that use item-to-item collaborative filtering for its operation. It is a filtering system that links purchased and rated items of the user to alike items, and intermixes these alike items into a recommendation list, instead of conventionally matching a user to similar customers. Therefore, to calculate the most alike match for a specific item, our algorithm creates an alike-items table by ascertaining items that customers are likely to buy together. A product-to-product matrix could be built by going through all pairs of items and calculating a metric to determine similarity for each pair. Howbeit, a lot of product pairs do not have routine customers. Thus, this approach is in-efficacious when it comes to the usage of memory and time processing. The algorithm below gives a more effective approach by computing the affinity between all related products and a single one as shown in Algorithm 3.1.

**Algorithm 3.1** Similarity Algorithm

**for** *each item in product catalog* $I_1$ **do**<br>
    **for** *each customer* $C$ *who purchased item* $I_1$ **do**<br>
        **for** *each item* $I_2$ *purchased by customer* $C$ **do**<br>
            Record that a customer purchased $I_1$ and $I_2$<br>
        **end for**<br>
    **end for**<br>
**end for**<br>
**for** *each item* $I_2$ **do**<br>
    Calculate the similarity between $I_1$ and $I_2$<br>
**end for**

Calculating the similarities between two items is possible in various ways, but a customary method used is the cosine measure. There, each vector corresponds to an item; then the customers who have bought that specific item corresponds to the vector's $M$ dimensions. This ciphering of the similar-items table that occurs offline is very liable to absorb a great deal of time, with $\mathcal{O}(N2M)$ as the most defective case. In application, though, it is closer to $\mathcal{O}(NM)$, because a lot of customers make little purchases. Selecting customers who buy best-selling titles decreases runtime additionally, with a small decrease in quality. Given a table of similar items, our algorithm detects items that are alike to the purchases and ratings of the user, sums those items, and then the most common or associated items are recommended. This calculation is very swift, relying only on the users' purchased or rated amount of items. The steps taken for this research are illustrated in Fig. 1.

| UserID | ProductID | Rating | Timestamp |
|---|---|---|---|
| AKM1MP6POOYPR | 132793040 | 5 | 1365811200 |
| A2CX7LUOHB2NDG | 321732944 | 6 | 1341100800 |
| A2NWSAGRHCP8N5 | 439886341 | 1 | 1367193600 |
| A2WNBOD3WNDNKT | 439886341 | 3 | 1374451200 |
| A1GIOU4ZRJA8WN | 439886341 | 1 | 1334707200 |
| A1QGNMC6O1VW39 | 511189877 | 5 | 1397433600 |

Table 1: Dataset Sample.

### 3.1. *Dataset*

In this study, we made use of the Yelp dataset from which 1000 product reviews were retrieved. It contains user comments and ratings of the products as shown in Table 1. Fig. 2 shows the steps applied to the datasets and the sample data collected.

Because the dataset comprises sparse data, we utilized a cross-validation method to realize training data at 60% and 40% as the test data which were randomly selected in an orderly form. To estimate the accuracy of the proposed similarity recommendation algorithm, we utilized the Root Mean Square Error (RSME), while precision and recall were employed to estimate the performance of its recommendation. We formulated and established the RSME equation as in Eq. 1

$$RMSE = \sqrt{\frac{1}{|W|} \sum\nolimits_{(v,k)\in W} \left(g_{vk} - g'_{vk}\right)^2}, \quad (1)$$

where $W$ as the set of test, $g_{vk}$ denotes exact ratings of respective, while $g'_{vk}$ signifies the ratings of prediction of the recommendation technique. Using different number of neighbors $r$ we tested the accuracy of the proposed algorithm. The number of neighbors $r$ is set as $r = 10, 15, 20, 25, 30, 35, 40, 45, 50$, respectively.

### 3.2. *Algorithm*

We used the Word2Vec algorithm which is a more modern model that roots words in a vector space that is lower-dimensional using a shallow neural network, as shown in Fig. 3. The application of Word2Vec is listed below:

(1) Take a 3-layer neural network (1 input layer + 1 hidden layer + 1 output layer).

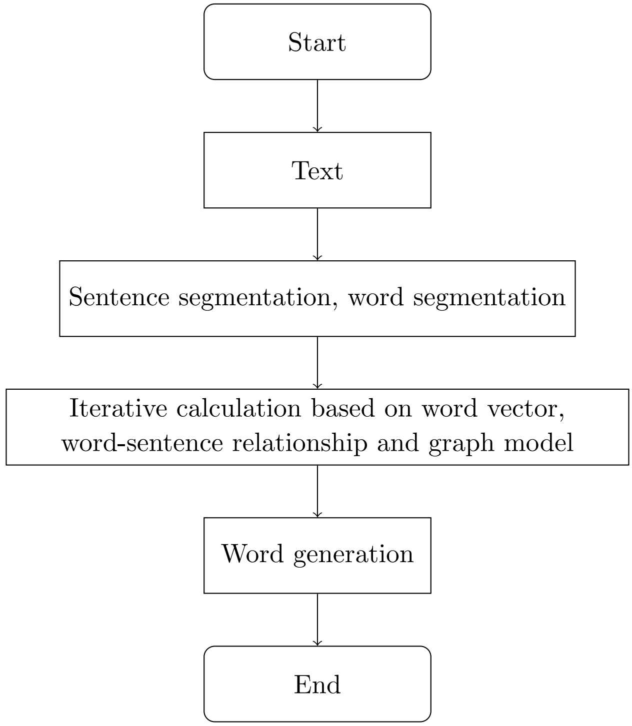


Fig. 3: Word2Vec Algorithmic Flow Chart

(2) Feed it a word and train it to predict its neighbouring word.
(3) Remove the last (output layer) and keep the input and hidden layer.
(4) Now, input a word from within the vocabulary. The output given at the hidden layer is the word embedding of the input word.
(5) That is it! Just doing this simple task enables our network to learn interesting representations of words.

For NLTK libraries, we used both BOW and TF-IDF approaches, semantic information is not stored. TF-IDF gives importance to uncommon words. So, there is a chance of overfitting. To overcome overfitting, the Word2Vec algorithm was used. In this particular model, each word is represented not as a single number, but as a vector that has 32 or more dimensions. This means each word is converted into a vector of 32 or more dimensions. Here, the semantic information and relation between different words are also preserved.

Fig. 4 shows the conversion of words into vectors in 2 dimensions, as well as where there is a huge vector difference between very good and bad. Instead of making one representation, it is better to use feature representation which is word embedding. Word2Vec is a lower dimension and dense matrix, whereas Bow is a higher dimension and sparse matrix.

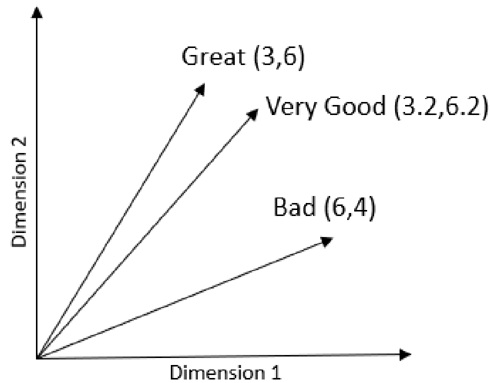


Fig. 4: Visual Representation

The steps for Word2Vec are given as follows:

(1) Tokenization of sentences
(2) Create histograms
(3) Take the most frequent word
(4) Create a grid with all the distinctive words. It characterizes the event relation linking the words too.

- **Data flow**: The flow of Data in our proposed system is displayed in Fig. 5. The output is 4, 2, and 0 rating which is derived from positive, neutral and negative comments, respectively. These ratings will add to existing system to recommend the item.
- **Word2Vec Objective Function**: For each position $t = 1, \ldots, T$, predict context words within a window of fixed size $m$, given center word $w_j$.

$$L(\theta) = \prod_{t-1}^{T} \prod_{\substack{-m \le j \le m \\ j \ne 0}} P(w_{t+j}|w_t;\theta) \tag{2}$$

The objective function $J(\theta)$ is the average negative log likelihood:

$$J(\theta) = -\frac{1}{T} logL(\theta) = -\frac{1}{T} \sum_{t-1}^{T} \sum_{\substack{-m \le j \le m \\ j \ne 0}} P(w_{t+j}|w_t;\theta) \tag{3}$$

Minimizing the objective function maximizes the predictive accuracy. To calculate $P(w_{t+j}|w_t;\theta)$, two vectors per word $w$ is used. $v_w$ when $w$ is

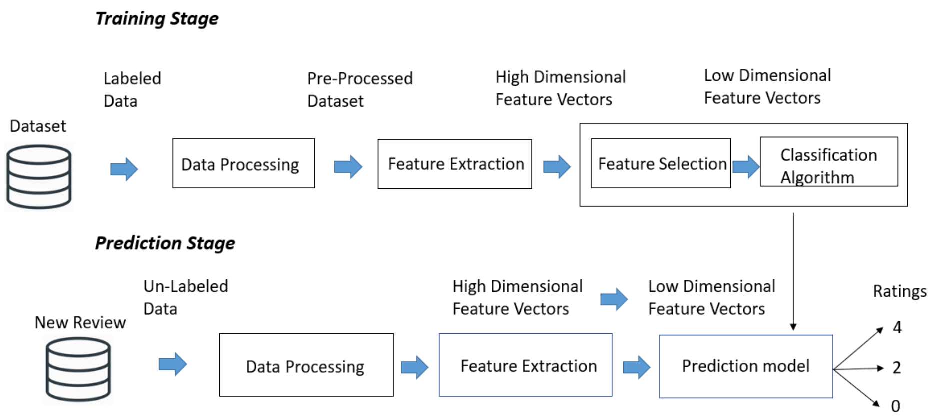


Fig. 5: Data Flow Diagram

a center word and $u_w$ when $w$ is a outside word. Then, for a center word $c$ and a context word $o$,

$$P(o|c) = \frac{exp(u_o^T v_c)}{\sum_{w \in v} exp(u_w^T v_c)} \tag{4}$$

The dot product $u_w^T v_c)$ compares the similarity of $o$ and $c$. $u_w^T = u.v = \sum_{i=1}^{n} u_i v_i$. $\sum_{e \in v} exp(u_w^T v_c)$ normalizes over the entire vocabulary to give a probability distribution.

This is an example of the softmax function $R^n \to R^n$.

$$softmax(x_i) = \frac{exp(x_i)}{\sum_{j=1}^{n} exp(x_j)} = P_i \tag{5}$$

The softmax function maps arbitrary values $x_i$ to a probability distribution $P_i$.

Word vectors provide an efficient feature representation of words, which helps in word similarity and other advanced tasks.

- **Conventional systems**: In comparison with existing systems, item-item based collaborative filtering (IBCF) is one of the recommendation methods that look for similar items that are based on the likes of the users or the items that are positively rated by the users. By looking at these items, IBCF recommends accordingly.
- **Task**: Finding similarity between Items$(i, j)$ where $i$ and $j$ are two items. This is illustrated in Fig. 6.

However, the equation in Fig. 6 does not consider the optimistic behaviours of users. So to handle this, the researchers use Eq. 6.

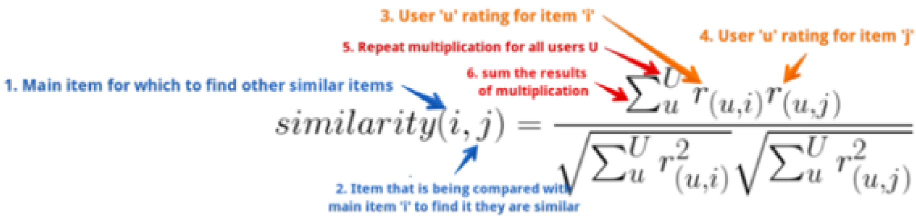


Fig. 6: Finding Similarity Between Items

$$similarity(i,j) = \frac{\sum_u^U (r_{(u,i)} - \bar{r}_u)(r_{(u,j)} - \bar{r}_u)}{\sqrt{\sum_u^U r_{(u,i)}^2}\sqrt{\sum_u^U r_{(u,j)}^2}} \quad (6)$$

$$score(u,i) = \frac{\sum_j^I similarity(i,j)(r_{(u,j)} - \bar{r}_j)}{\sum_j^I similarity(i,j)} + \bar{r}_i, \quad (7)$$

where $u$ is the user for whom recommendation is being generated, $i$ is the item considered for recommendation, $score(u,i)$ generates the recommendation score of item $i$ to user $u$, and $j$ indicates items which are similar to item $i$.

### 3.3. *Procedure*

Fig. 7 shows the recommendation flow. To train the dataset, we used a Word2Vec algorithm where it divides the comments into three categories- positive, neutral and negative into ratings, i.e. 4, 2, and 0. This adds to the existing system where item recommendations are based on ratings from $1-5$. It generates temporary table mapping from the purchased item to the customer after it retrieves all the ratings for each item (high rated item, other items) pair-count, where the common customer is sorted via a priority list. After passing all the items through the filters and truncating the other items, it recommends the relevant items based on their ratings and comments.

### 3.4. *Evaluation Metrics*

A confusion matrix is a figure that is frequently used to explain the execution of a classification model (or classifier) on a set of test data for which the actual estimates are recognized [47,48]. The confusion matrix is fairly straightforward to comprehend, but the affiliated expressions can be puzzling. There are several measures offered for calculating and equating the outcomes of our experiments. The most commonly used measures include precision, accuracy etc. It is important to note that each of

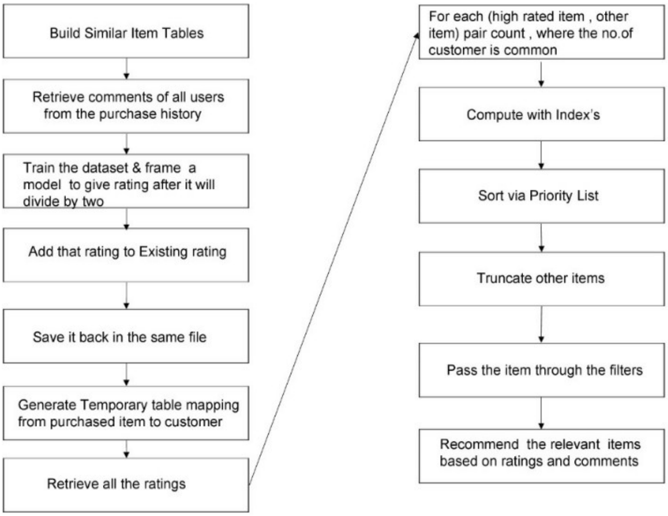


Fig. 7: The Recommender Flowchart

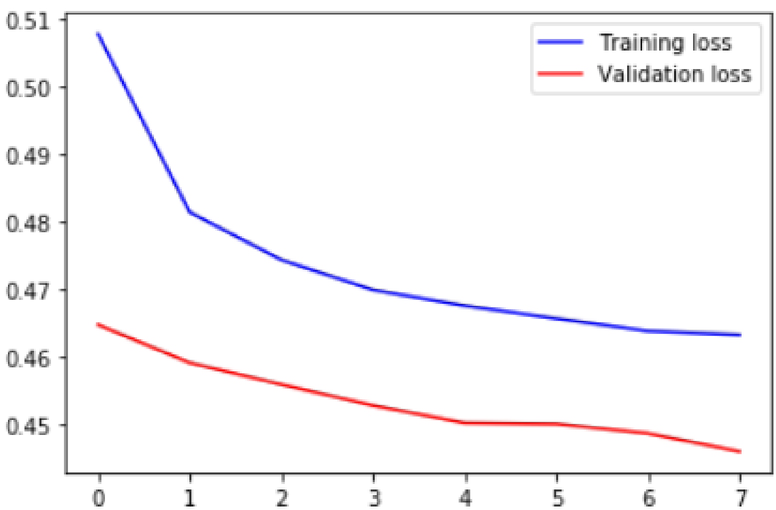


Fig. 8: Training and Validation Loss

these measures is computed separately for each class, then the mean for the general classifier performance is calculated.

$$\text{Accuracy} = \frac{T_p + T_n}{T_p + F_p + F_n + T_n} \tag{8}$$

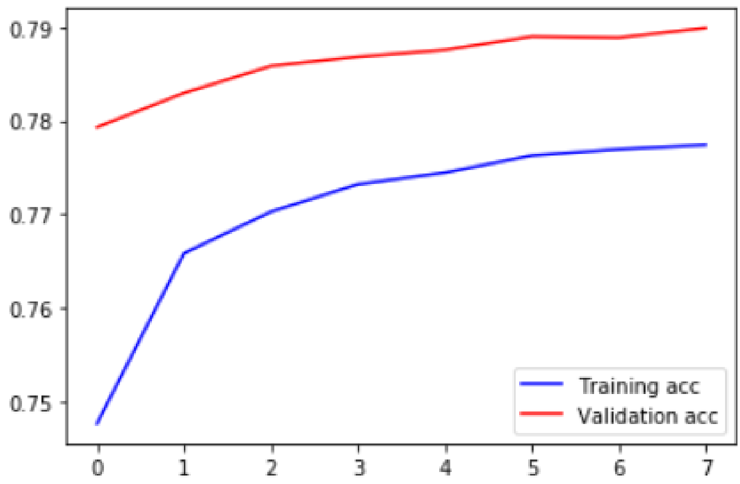


Fig. 9: Training and Validation Accuracy

$$\text{Precision} = \frac{T_p}{T_p + F_p} \tag{9}$$

$$\text{Recall} = \frac{T_p}{T_p + F_n} \tag{10}$$

$$\text{F1-Measure} = 2 * \frac{\text{Precision*Recall}}{\text{Precision + Recall}} \tag{11}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |x_i - x| \tag{12}$$

## 4. Results and Discussion

We have used the above libraries to simplify our approach and we know that most ML model developers use this for developing their models. There are three classifications of a dataset.

- **Training:** The model is trained using a training dataset.
- **Validation:** This helps to prevent Overfitting. The dataset is validated for every epoch, every time the weights are adjusted and the training loss is calculated.
- **Test:** The model is tested with the test dataset. The accuracy obtained at this stage is the accuracy of our machine learning algorithm.

| | ITEM_1 | | | ITEM_2 | | | ITEM_3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | RATING | COMMENT | TOTAL RATING | RATING | COMMENT | TOTAL RATING | RATING | COMMENT | TOTAL RATING |
| USER_1 | 4 | I Love this product | 4.5 | 1 | I hate this product | 0.5 | - | - | - |
| USER_2 | 5 | - | 5 | 2 | Satisfactory | 2.5 | 5 | This product is best | 5 |
| USER_3 | 3 | The product is worthful for money | 3 | 1 | Not good product | 1 | - | - | - |

Fig. 10: The Recommender technique

Note : Total rating $= \frac{\text{rating+comment rating}}{2}$

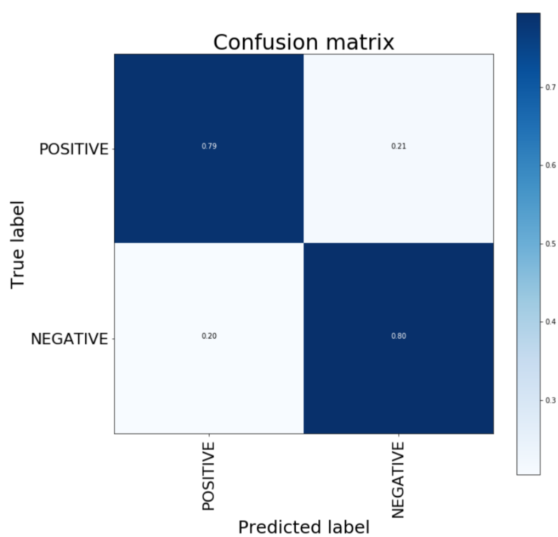


Fig. 11: Confusion Matrix

Meanwhile, if the validation loss is much higher than the training loss, it could mean the network may be overfitted. In Fig. 8, the training loss is more than validation loss. That means this network is exactly fitted. Fig. 9 illustrates the training and validation accuracy when the dataset fragmented is not arbitrary. In order words, spatial or temporal configurations may occur. Moreover, the validation set could be essentially unfitted, which means there may be less bluster or less friction from the training. Thus, it will be easier to forecast, resulting in higher precision on the validation set than on training.

If the training loss is far less than validation loss then this means the network might have been overfitted. But in the above case training loss is more than validation loss, which means the network is exactly fitted. Fig. 9 illustrates the training and validation accuracy, that is when the dataset is not randomly divided especially when temporal or spatial patterns tend to exist. There will be less noise or less variance because the validation set is different from the training set. Therefore, it is easier to predict a higher accuracy of precision on the validation set used during the training.

predict('Satisfactory')
'label': 'NEUTRAL', 'score': 0.2848373985215409, 'elapsed_time': 0.25728774070739742

predict('I love this product')
'label': 'POSITIVE', 'score': 0.9656286239624023, 'elapsed_time': 0.44394254684448241
predict('I hate this item')
'label': 'NEGATIVE', 'score': 0.010753681883215904, 'elapsed_time': 0.266440868377685551

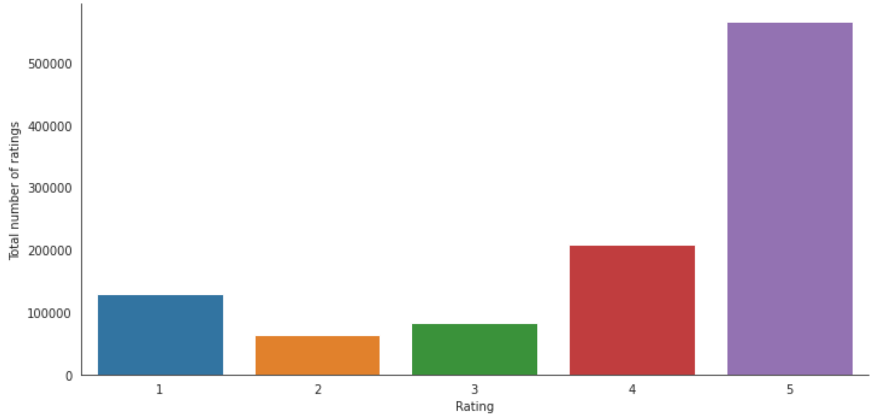


Fig. 12: Statistics of Ratings by Product

After predicting the ratings for user comments based on the Word2Vec algorithm, it will add those ratings to the existing rating. Fig. 10 shows the Recommender Technique as the sample output of rating that illustrates the process. For item_1 , user_1, given a 4-star rating and comment 'I love this product, our model was trained to convert comments into ratings. There, ratings and comment ratings will be divided by 2 and that will be the final rating. Based on this rating, the item will be recommended to the user.

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Negative | 0.79 | 0.79 | 0.79 | 159494 |
| Positive | 0.79 | 0.80 | 0.79 | 160506 |
| Micro Avg | 0.79 | 0.79 | 0.79 | 320000 |
| Macro Avg | 0.79 | 0.79 | 0.79 | 320000 |
| Weighted Avg | 0.79 | 0.79 | 0.79 | 320000 |

Table 2: Classifier Report

| UserID | ProductID | Rating | Timestamp |
|---|---|---|---|
| AKM1MP6POOYPR | 132793040 | 5 | 1365811200 |
| A2CX7LUOHB2NDG | 321732944 | 6 | 1341100800 |
| A2NWSAGRHCP8N5 | 439886341 | 1 | 1367193600 |
| A2WNBOD3WNDNKT | 439886341 | 3 | 1374451200 |
| A1GIOU4ZRJA8WN | 439886341 | 1 | 1334707200 |
| A1QGNMC6O1VW39 | 511189877 | 5 | 1397433600 |
| A3J3BRHTDRFJ2G | 511189877 | 2 | 1397433600 |
| A2TYOBTJOTENPG | 511189877 | 5 | 1395878400 |
| A34ATBPOK6HCHY | 511189877 | 9 | 1395532800 |
| A89D069P0XZ27 | 511189877 | 5 | 1395446400 |
| AZYNQZ94U6VDB | 511189877 | 5 | 1401321600 |
| A1DA3W4GTFXP6O | 528881469 | 5 | 1405641600 |
| A29LPQQDG7LD5J | 528881469 | 1 | 1352073600 |
| AO94DHGC771SJ | 528881469 | 5 | 1370131200 |
| AMO214LNFCEI4 | 528881469 | 1 | 1290643200 |

Table 3: Product Rating

Table 2: the classifier report shows Precision, Recall and F1-score for negative comments is 0.79 and for positive comments Recall is different i.e., 0.80.

Fig. 11 describes the confusion matrix which explains the classification model's performance in the form of two labels, which are the true label (actual) and predicted label. It is easy to find accuracy through the confusion matrix Accuracy = $(0.80 + 0.79)/(0.80 + 0.21 + 0.20 + 0.79) = 0.79$ This calculation clearly shows as model Accuracy is 0.79.

Table 3 shows the ratings each user id given for each product plus the ratings from converting the comments. Ratings = user ratings given by user + ratings based

| ProductID | Rating | Rating-count |
| --- | --- | --- |
| 1400501466 | 3.560000 | 250 |
| 1400501520 | 4.243902 | 82 |
| 1400501776 | 3.884892 | 139 |
| 1400532620 | 3.684211 | 171 |
| 1400532655 | 3.727273 | 484 |

Table 4: Number of Ratings per Product

| ProductID | Ave. Rating |
| --- | --- |
| 130000DYV9H | 4.947368 |
| 8000053HC5 | 4.945783 |
| 1300009R96C | 4.885714 |
| 1300005LE76 | 4.879310 |
| 13000I1X3W8 | 4.869565 |

Table 5: Average Ratings per Product

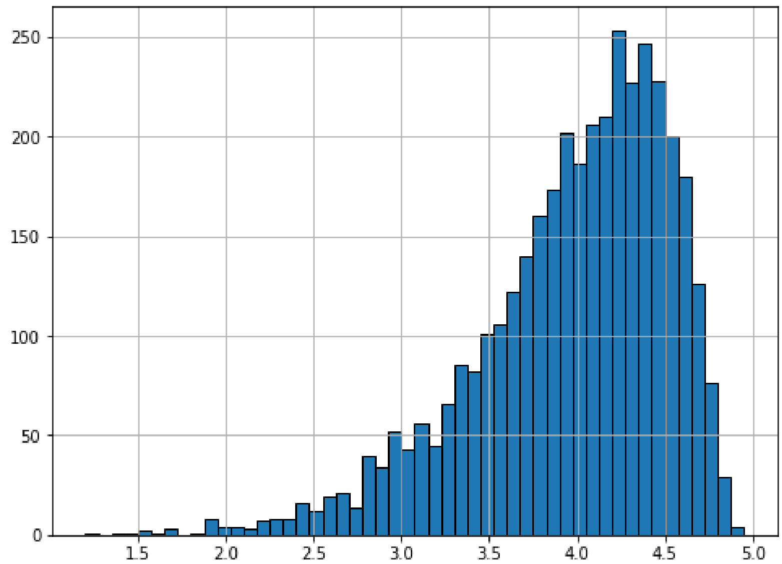


Fig. 13: Number of Ratings per Scale

on user comment. While Fig. 12 shows the statistics of the rating concerning the product. Table 4 shows the rating count of each product. While Table 5 shows a sample of average ratings for each product.

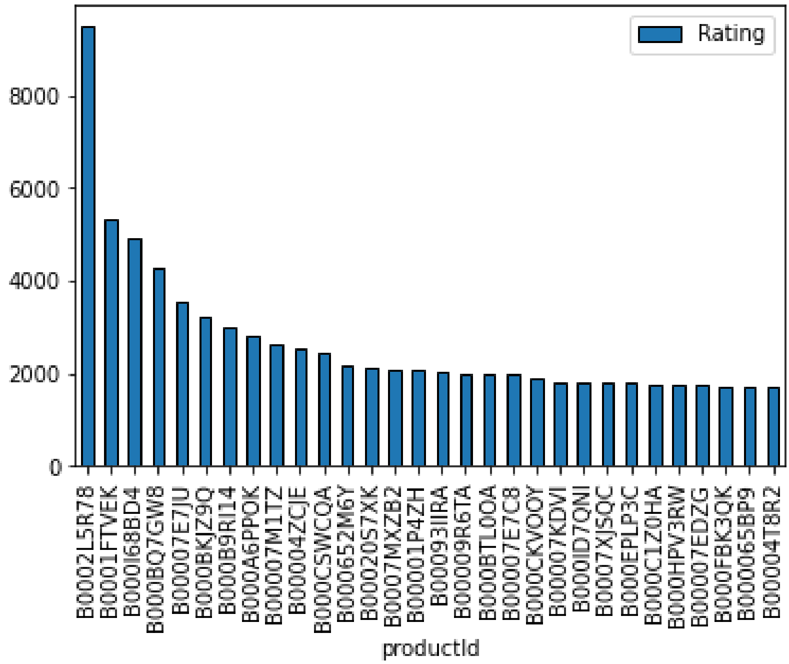


Fig. 14: Number of Ratings by User ID and Product ID

Fig. 13 represents the number of ratings according to the given scale i.e., $1.5, 2.0, 2.5, \ldots, 5.0$. Whereas, Fig. 14 shows the number of ratings for products.

### 4.1. *Performance Comparison*

In this section, we experimentally compare the performance of our proposed algorithm against conventional algorithm with respect to their RSME value, Precision, Recall, and Accuracy. Fig. 15 illustrates the value of RSME for the two compared algorithms with the different changes in the number of nearest neighbor.

Concerning the results obtained by utilizing a different number of neighbours between 10 and 55, the Root Mean Square Error (RSME) of the proposed similarity algorithm is made up of an insignificant prediction error when compared with the conventional algorithm. Thus, this result denotes that the proposed similarity recommendation algorithm performs better concerning prediction accuracy. The research also utilized Precision and Recall to estimate the performance of the proposed similarity algorithm in making relevant/irrelevant recommendations. Using Precision, the experiment estimates the number of items in the list of Top $M$ for clusters that fit in individual cluster associates. On the other hand, the Recall is utilized to estimate the number of the Top $M$ items in the list that are properly predicted by the similarity recommendation algorithm.

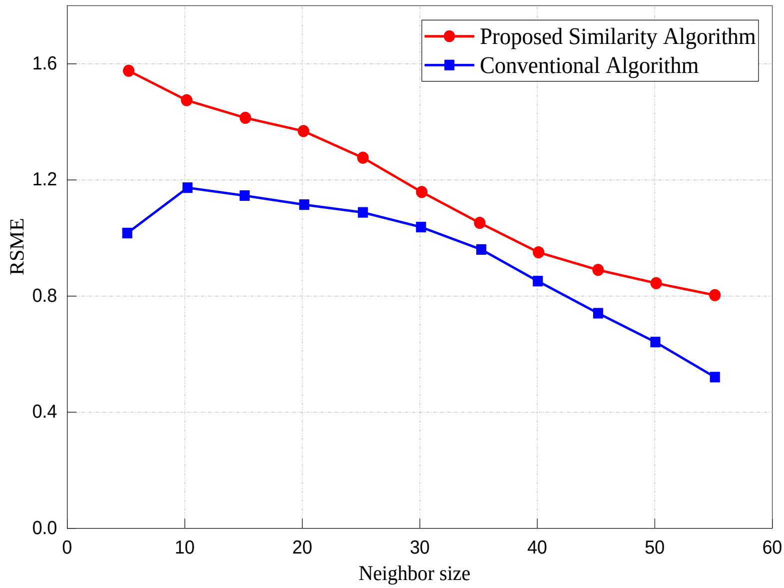


Fig. 15: Comparison of RSME values

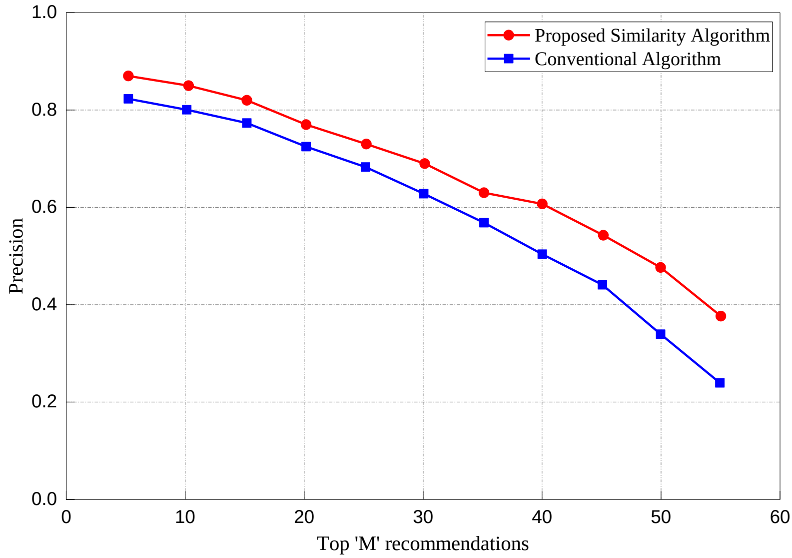


Fig. 16: Precision Performance Comparison

The recommendation performance of the algorithm is verified according to different amounts of Top $M$ recommendations ranging from 5 to 55. The experiment

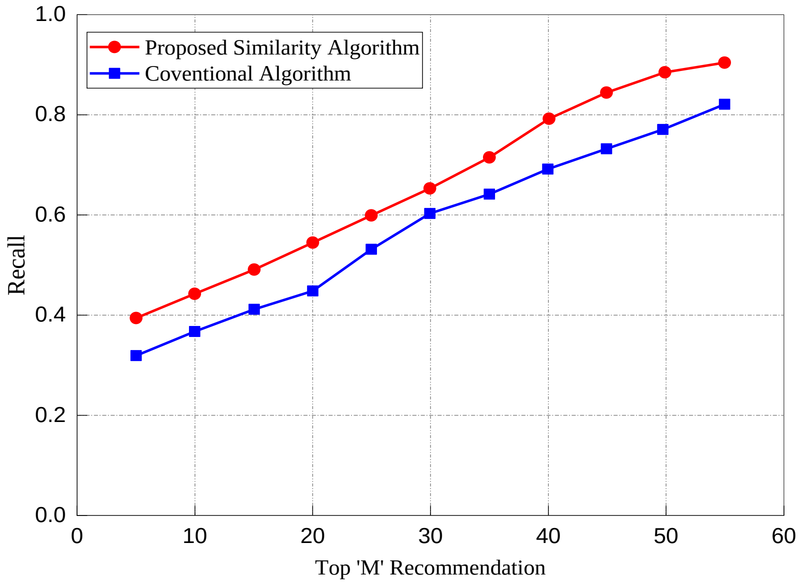


Fig. 17: Recall Performance Comparison

in Fig. 16 illustrates the comparative performance of both the proposed and conventional recommendation algorithms concerning precision. The result of the experiment indicates that the proposed similarity recommendation algorithm outperforms the compared conventional algorithm in terms of precision and recall, respectively. It denotes that the similarity algorithm achieves better performance in recommending significant items only, consequently disregarding insignificant items to be recommended. Likewise, the experiment in Fig. 17 demonstrates the evaluation of two algorithms concerning the recall.

## 5. Conclusion

Item recommendation has been fully based on ratings like $1-5$ which is not included in the comment section where users and/or customers express their thoughts and feelings about a given product. In this work, a classification technique is applied with a different approach compared to other models that considers not only relevant reviews but also relevant sub-reviews and users' comments which can play a major role in a recommendation. In other words, a novel attention mechanism is introduced in this work that retrieves essential reviews by placing dynamic attention on the item reviews improving the scalability of additional reviews for a better recommendation. This paper shows improvements in the Recommender System used by YouTube, Facebook, and Amazon where users' comments and reviews are not considered. We have proposed a Machine Learning model system with 0, 2, 4 scored ratings where 0 is a negative score, 2 is neutral, and 4 is positive, respectively. Additionally, we propose a review system that takes care of all the users' comments and reviews and will not disturb present system implementations. We have implemented this model using Keras, Pandas and Sci-kit Learning libraries to run the internal work faster. The proposed approach improved prediction with 79% Accuracy for Yelp datasets of businesses across 11 metropolitan areas in four countries with a Mean Absolute Error (MAE) of 21%, Precision at 79%, Recall at 80%, and F1-Score of 79%, respectively. The proposed similarity algorithm was compared with conventional algorithms to estimate its performance and accuracy in terms of its Root Mean Square Error (RSME), Precision, and Recall. Results of this experimental analysis indicate that the similarity recommendation algorithm performs better than the conventional algorithm and enhances recommendation accuracy. Our model shows scalability advantages with how organizations can revolutionize their Recommender Systems to attract potential customers and increase patronage. In the future, we shall introduce an attention mechanism, which retrieves essential reviews by placing dynamic attention on three quadrants of every set of item reviews, hence improving the time latency of extracting essential reviews for a better recommendation. The model will use pointers in each quadrant to extract relevant reviews, not leaving any sub-reviews as compared to our model. The pointer mechanism extracts the named reviews for direct review-to-review matching. Moreover, exploring other newer diverse datasets would improve the applicability of the framework and provide glimpses into more future directions.

## References


1. H. Zhang, G. Xu, X. Liang, T. Huang and K. Fu, An attention-based word-level interaction model: Relation detection for knowledge base question answering, *CoRR* **abs/1801.09893** (2018).
2. Y. Li, G. Kou, G. Li and H. Wang, Multi-attribute group decision making with opinion dynamics based on social trust network, *Information Fusion* **75** (2021) 102–115.
3. H. Wang, Y. Peng and G. Kou, A two-stage ranking method to minimize ordinal violation for pairwise comparisons, *Applied Soft Computing* **106** (2021) p. 107287.
4. T. Li, G. Kou, Y. Peng and Y. Shi, Classifying with adaptive hyper-spheres: An incremental classifier based on competitive learning, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **50**(4) (2020) 1218–1229.
5. G. Kou, Y. Peng, X. Chao, E. Herrera-Viedma and F. E. Alsaadi, A geometrical method for consensus building in gdm with incomplete heterogeneous preference information, *Applied Soft Computing* **105** (2021) p. 107224.
6. G. Li, G. Kou and Y. Peng, Heterogeneous large-scale group decision making using fuzzy cluster analysis and its application to emergency response plan selection, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2021).
7. C. Lin and G. Kou, A heuristic method to rank the alternatives in the ahp synthesis, *Applied Soft Computing* **100** (2021) p. 106916.
8. P. Melville and V. Sindhwani, Recommender systems., *Encyclopedia of machine learning* **1** (2010) 829–838.
9. P. Pu and L. Chen, Trust building with explanation interfaces, in *Proceedings of the 11th international conference on Intelligent user interfaces*, 2006 pp. 93–100.
10. I. Portugal, P. Alencar and D. Cowan, The use of machine learning algorithms in rec-

ommender systems: A systematic review, *Expert Systems with Applications* **97** (2018) 205–227.

11. A. V. Singh, M. Sharma *et al.*, Building an effective recommender system using machine learning based framework, in *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS)*, 2017 pp. 215–219.
12. F. Isinkaye, Y. Folajimi and B. Ojokoh, Recommendation systems: Principles, methods and evaluation, *Egyptian Informatics Journal* **16**(3) (2015) 261–273.
13. M.-I. Dascalu, C.-N. Bodea, A. Moldoveanu, A. Mohora, M. Lytras and P. O. de Pablos, A recommender agent based on learning styles for better virtual collaborative learning experiences, *Computers in Human Behavior* **45** (2015) 243–253.
14. T. Duricic, E. Lacic, D. Kowald and E. Lex, Trust-based collaborative filtering: Tackling the cold start problem using regular equivalence, in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018 pp. 446–450.
15. R. Logesh and V. Subramaniyaswamy, A reliable point of interest recommendation based on trust relevancy between users, *Wireless Personal Communications* **97**(2) (2017) 2751–2780.
16. X. Ma, J. Ma, H. Li, Q. Jiang and S. Gao, Armor: A trust-based privacy-preserving framework for decentralized friend recommendation in online social networks, *Future Generation Computer Systems* **79** (2018) 82–94.
17. P. Berkhim, Z. Xu, J. Mao, D. E. Rose, A. Taha and F. Maghoul, Trust propagation through both explicit and implicit social networks(May 14 2013), US Patent 8,442,978.
18. Y. Li, J. Liu and J. Ren, A novel learning model based on trust diffusion and global item for recommender systems, *IEEE Access* **7** (2019) 170270–170281.
19. M. Leiva, M. C. Budan and G. I. Simari, Guidelines for the analysis and design of argumentation-based recommendation systems, *IEEE Intelligent Systems* (2020).
20. C. Iwendi, S. Khan, J. H. Anajemba, A. K. Bashir and F. Noor, Realizing an efficient iomt-assisted patient diet recommendation system through machine learning model, *IEEE Access* **8** (2020) 28462–28474.
21. J. Naren, M. Z. Banu and S. Lohavani, Recommendation system for students' course selection, in *Smart Systems and IoT: Innovations in Computing*, (Springer, 2020) pp. 825–834.
22. N. N. Qomariyah and A. N. Fajar, Recommender system for e-learning based on personal learning style, in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2019 pp. 563–567.
23. A. Mottini and R. Acuna-Agost, Deep choice model using pointer networks for airline itinerary prediction, *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2017).
24. R. E. Nakhli, H. Moradi and M. A. Sadeghi, Movie recommender system based on percentage of view, in *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, 2019 pp. 656–660.
25. L. T. Ponnam, S. Deepak Punyasamudram, S. N. Nallagulla and S. Yellamati, Movie recommender system using item based collaborative filtering technique, in *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, 2016 pp. 1–5.
26. J. M. Musa and X. Zhihong, Item based collaborative filtering approach in movie recommendation system using different similarity measures, in *Proceedings of the 2020 6th International Conference on Computer and Technology Applications*, 2020 pp. 31–34.
27. D. Goldberg, D. Nichols, B. M. Oki and D. Terry, Using collaborative filtering to

weave an information tapestry, *Communications of the ACM* **35**(12) (1992) 61–70.
28. M. Hasan and F. Roy, An item–item collaborative filtering recommender system using trust and genre to address the cold-start problem, *Big Data and Cognitive Computing* **3**(3) (2019) p. 39.
29. B. Sarwar, G. Karypis, J. Konstan and J. Riedl, Item-based collaborative filtering recommendation algorithms, in *Proceedings of the 10th international conference on World Wide Web*, 2001 pp. 285–295.
30. M.-P. T. Do, D. Nguyen and L. Nguyen, Model-based approach for collaborative filtering, in *6th International Conference on Information Technology for Education*, 2010 pp. 217–228.
31. M. Deshpande and G. Karypis, Item-based top-n recommendation algorithms, *ACM Transactions on Information Systems (TOIS)* **22**(1) (2004) 143–177.
32. C. Xia, X. Jiang, S. Liu, Z. Luo and Z. Yu, Dynamic item-based recommendation algorithm with time decay, in *2010 Sixth International Conference on Natural Computation*, **1** 2010 pp. 242–247.
33. D. Li, Q. Lv, L. Shang and N. Gu, Item-based top-n recommendation resilient to aggregated information revelation, *Knowledge-Based Systems* **67** (2014) 290–304.
34. D. Li, C. Chen, Q. Lv, L. Shang, Y. Zhao, T. Lu and N. Gu, An algorithm for efficient privacy-preserving item-based collaborative filtering, *Future Generation Computer Systems* **55** (2016) 311–320.
35. A. M. Dakhel, H. T. Malazi and M. Mahdavi, A social recommender system using item asymmetric correlation, *Applied Intelligence* **48**(3) (2018) 527–540.
36. S. Wang, L. Cao, L. Hu, S. Berkovsky, X. Huang, L. Xiao and W. Lu, Jointly modeling intra-and inter-transaction dependencies with hierarchical attentive transaction embeddings for next-item recommendation, *IEEE Intelligent Systems* (2020).
37. Z. Li, F. Xiong, X. Wang, Z. Guan and H. Chen, Mining heterogeneous influence and indirect trust for recommendation, *IEEE Access* **8** (2020) 21282–21290.
38. Y. Jiang, H. Chen and B. Yang, Deep social collaborative filtering by trust, in *Proceedings of 2018 International Conference on Big Data Technologies*, 2018 pp. 52–56.
39. Z. Zhang, Y. Liu, Z. Jin and R. Zhang, A dynamic trust based two-layer neighbor selection scheme towards online recommender systems, *Neurocomputing* **285** (2018) 94–103.
40. V. Nobahari, M. Jalali and S. J. S. Mahdavi, Isotrustseq: a social recommender system based on implicit interest, trust and sequential behaviors of users using matrix factorization, *Journal of Intelligent Information Systems* **52**(2) (2019) 239–268.
41. S. Linda and K. K. Bharadwaj, A fuzzy trust enhanced collaborative filtering for effective context-aware recommender systems, in *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 2*, 2016 pp. 227–237.
42. R. Yera and L. Martinez, Fuzzy tools in recommender systems: A survey, *International Journal of Computational Intelligence Systems* **10**(1) (2017) 776–803.
43. H. Parvin, P. Moradi and S. Esmaeili, Tcfaco: Trust-aware collaborative filtering method based on ant colony optimization, *Expert Systems with Applications* **118** (2019) 152–168.
44. Y.-D. Seo, E. Lee and Y.-G. Kim, Video on demand recommender system for internet protocol television service based on explicit information fusion, *Expert Systems with Applications* **143** (2020) p. 113045.
45. X. Ma, H. Lu, Z. Gan and J. Zeng, An explicit trust and distrust clustering based collaborative filtering recommendation approach, *Electronic Commerce Research and Applications* **25** (2017) 29–39.

46. Q. Zha, G. Kou, H. Zhang, H. Liang, X. Chen, C.-C. Li and Y. Dong, Opinion dynamics in finance and business: a literature review and research opportunities, *Financial Innovation* **6**(1) (2020) 1–22.
47. J. H. Anajemba, Y. Tang, J. A. Ansere and S. H. Sackey, Efficient switched digital beamforming radar system based on SIMO/MIMO receiver, in *Computing, Communications and IoT Applications, ComComAp 2019, Shenzhen, China, October 26-28, 2019*, (IEEE, 2019), pp. 411–416.
48. S. H. Sackey, J. Chen, J. H. Anajemba and X. Zhang, A clustering approach based on genetic algorithm for wireless sensor network localization, in *15th International Conference on Computational Intelligence and Security, CIS 2019, Macao, SAR, China, December 13-16, 2019*, (IEEE, 2019), pp. 54–58.