




# Customer centric hybrid recommendation system for E-Commerce applications by integrating hybrid sentiment analysis

Arodh Lal Karn<sup>1</sup> · Rakshha Kumari Karna<sup>2</sup> · Bhavana Raj Kondamudi<sup>3</sup> · Girish Bagale<sup>4</sup> · Denis A. Pustokhin<sup>5</sup> · Irina V. Pustokhina<sup>6</sup> · Sudhakar Sengan<sup>7</sup> 

Accepted: 13 October 2022 / Published online: 29 October 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

Internet applications such as Online Social Networking and Electronic commerce are becoming incredibly common, resulting in more content being available. Recommender systems (RS) assist users in identifying appropriate information out of a large pool of options. In today's internet applications, RS are extremely important. To increase user satisfaction, this type of system supports personalized recommendations based on a massive volume of data. These suggestions assist clients in selecting products, while concerns can increase product utilization. We discovered that much research work is going on in the field of recommendation and that there are some effective systems out there. In the context of social information, sentimental analysis (SA) can aid in increasing the knowledge of a user's behaviour, views, and reactions, which is helpful for incorporating into RS to improve recommendation accuracy. RS has been found to resolve information overload challenges in information retrieval, but they still have issues with cold-start and data sparsity. SA, on the other hand, is well-known for interpreting text and conveying user choices. It's frequently used to assist E-Commerce in tracking customer feedback on their products and attempting to comprehend customer needs and preferences. To improve the accuracy and correctness of any RS, this paper proposes a recommendation model based on a Hybrid Recommendation Model (HRM) and hybrid SA. In the proposed method, we first generate a preliminary recommendation list using a HRM. To generate the final recommendation list, the HRM with SA is used. In terms of various evaluation criteria, the HRM with SA outperforms traditional models.

**Keywords** Hybrid recommender system · E-Commerce · Sentiment analysis · Neural network · Accuracy · Filtering · Deep learning · Bert

---

✉ Sudhakar Sengan  
sudhasengan@gmail.com

Extended author information available on the last page of the article

## 1 Introduction

Electronic commerce (E-Comm) has been rising to prominence in our day-to-day life. The way people adopt to do business across the world is transformed by E-Comm. Over the years, it has changed drastically. The purchasing and selling of any goods and services by people, including books, electronics, clothes, software, and furniture, using the internet platform is performed by E-Comm. E-Comm enhances the users to buy and make money by selling what they like, whenever they want. The main mode of commerce is the Brick-And-Mortar (BAM) model before the emergence of E-Comm. Retail stores owning a minimum of fixed premises are called BAM businesses. Buying and selling items necessitate the buyer and seller who are involved in the communication of active and physical means. The BAM business is quickly succeeded by E-Comm in spite of its existence. On the other hand, E-Comm stores are the place where a majority of BAM retailers are entering. Consequently, there is a requirement for developing a web profile and transferring the crucial business functions online.

The research that displays the facts and trends for portraying the global E-Comm development [1] to help businesses launch outside borders well depicts the expansion of E-Comm. The research includes facts and trends on E-Comm from 2014 to 2023, as well as projections for the future of E-Comm. Global E-Comm sales are expected to expand by 276.9% over the most recent recorded period, according to cumulative data. More people will be able to gain from most online services, especially E-Comm, as Internet usage and accessibility improve. E-Comm has become commonplace in people's daily lives, with numerous advantages. A relatively recent transformation, E-Comm ("online commerce"), renders it feasible to buy and sell products and services digitally. This paper outlines the various types of E-Commerce businesses, each of which provides services to a distinct business model. An offline, "brick-and-mortar" establishment is one that has a physical location, such as a storefront or office building, from which it sells its wares and provides its services directly to customers. The feature makes it possible for customers to investigate and try out products before deciding to purchase them. Customers receive immediate gratification. Permit customers to inquire about the products or services with members of staff. Permit customers to inquire about the products or services with members of staff. Several distinct categories of E-Commerce solutions exist, including Business-to-Business (B2B), Business-to-Consumer (B2C), Consumer-to-Consumer (C2C), Consumer-to-Business (C2B), Business-to-Administration (B2A), Consumer-to-Administration (C2A). The location where the two types of businesses sell their products is the first and most absolutely essential feature that distinguishes them.

Over time, the internet has evolved into the main thoroughfare for E-Comm transactions. All industries and consumer types of businesses have jumped on to the expressway for reaching millions of clients over the internet, from banks delivering internet banking to ticket booking for airlines, hotels, and travel through insurance sectors. These aren't the only companies that use E-Comm to run business. Most giant firms have adopted the E-Comm model for online procurement, sales

order management, and customer management. ERP solutions have successfully integrated and brought all business operations modules online in web mode. Business organizations are investing millions of dollars in renovating their Information Technology infrastructure and application platforms and re-engineering their company processes to enable the usage of the E-Comm interface for respective business functions. Online stores have a huge advantage over BAM stores because they can sell to more customers. Internet-based stores host their operations. Customers can shop from anywhere, but they can't try on or touch the products. Many of them serve entire states, nations, and the world. The ease of using online businesses is another benefit; this is especially helpful for customers who want to avoid the hassle and extra time of in-store purchases. E-Comm sales are growing faster than BAM sales and may surpass in-person sales. Online retailers' main drawback is administrative costs, which most conventional shops don't have. Because there are no digital shelves or warehouses to store physical products, online retailers must pay for delivery (if they don't ask customers to pay), packaging, and warehouse management. Online shoppers can't try out products, which is another issue. Customers may lose interest in a product or service they can't physically inspect.

Consumers were first restricted by the small display as well as slow bandwidth. However, due to the widespread usage of smartphones and tablets, this situation is rapidly changing. Furthermore, customers are becoming more receptive to the portable culture. Furthermore, the development of 4G, as well as 5G networks, is hastening the acceptance of Mobile-Commerce (M-Comm) (and soon 6G). Finally, numerous locations offer free Wi-Fi Internet connection. The Internet of Things (IoT) is likely to have a significant impact on all aspects of our life. Apart from making our lives easier and wiser, the linked ecosystems of devices, as well as gadgets, is gradually changing our old behaviours. Retail and shopping experiences have greatly improved as a result of IoT-driven innovation. BAM stores have physical locations where customers can view goods. Customers must travel there to shop. Online sellers have opposite pros and cons to physical stores. A BAM shop has many advantages over an online store, including not having to pay for shipping, having warehouses to store its goods, and allowing customers to examine them closely before buying. Physical business and trade offer instant rewards. Buyers aren't subjected to weeks-long delays. They leave with it. In-person sales have disadvantages. Because customers must visit the store, BAM stores can likely only sell to customers within a certain radius. Single-location businesses serve only the neighbourhood. More extensive chain operations can oppose E-Commerce here. Online shopping is often more convenient than going to a store for quick purchases.

While mobile commerce apps have mostly replaced traditional E-Comm stores as most customers prefer to browse and buy on their mobile devices, linked devices with location-aware sensor technologies such as beacons and geofencing have helped to close the gap between physical and online retailers. The buyer can quickly move from online and offline purchasing experiences thanks to IoT. In terms of organizational quality and customer service, linked devices are increasingly leading the way for enhanced quality through automation and increased productivity with fewer people. Mobile application development features and needs for E-Comm companies have shifted to the latest technology and automation as a result of IoT.

IoT is revolutionizing E-Comm and M-Comm businesses and mobile shopping experiences in a variety of ways, from faster supply chain operations to automated sorting centres and from smart inventory systems to a personalized experience for customers.

Online retailers will be able to personalise customer satisfaction in two different ways using the smartphone application. First, generate deals based on customers' preferences; second, provide deals and delivery at times convenient for customers. Retailers can also set up system alerts and email warranty extension offers when information from a user's mobile app is updated, including the manufacturer's warranty and the date of purchase. In the case of mobile devices, users may not even be aware of a problem until it is reported by the device's developer. This implies that potential issues can be identified and addressed by store owners on time. The current in-store shopping experience can be customised by monitoring store traffic and customer demand in real-time. That opens the door for extensive in-store digital marketing and event announcements via mobile devices.

Cognitive computing and Machine Learning (ML) have recently become more efficient and reliable in the linked ecosystem of gadgets and sensors. Apart from IoT's potential to streamline processes through real-time data exchange and processing, IoT-led automation for mobile commerce businesses is benefiting from ML, AI, and cognitive computing to provide accurate demand predictions. In the coming years, IoT will be observed in association with these technologies in the area of M-Comm. The majority of E-Comm, as well as M-Comm stores, are now focusing on using data analytics and a variety of prediction tools to provide highly tailored shopping experiences to customers. All of these techniques and technology are likely to aid inventory services and customer forecasting. In the coming years, automation is anticipated to be a crucial feature for effective M-Comm stores. Because competition is increasing and there is limited room for profit margin expansion until you lower your bottom line costs, E-Comm stores will place a greater emphasis on automation to minimize personnel for repetitive jobs and thus reduce costs. The system under study is assumed to be a Markov process (or "hidden" states) in a Hidden Markov Model (HMM), a type of statistical Markov model. The existence of an observable process whose results are "influenced" by results in a known manner is a prerequisite for the definition of HMM.

The recommendation system (RS) [2] is one technique to make shopping easier for customers by providing a list that offers customer-specific goods depending on the customer's preferences. Much research has been published in this sector that indicates various strategies to develop RS, thus enhancing the accuracy of commercial sites. A Recommender System (RS) is a type of data filtering system that tries to predict a customer's "rating" or "preference" for a product [3]. Examples of RS in use include playlist generators in music and video services, product recommenders in online merchants, contents RS or social media sites, and open online content recommenders [4]. Inside and across the platforms, such systems could focus on a single input, like music, or multiple inputs, like books, news and search queries. [5]. There are also popular RS for niche themes like restaurants as well as online dating. RS has been used to investigate research publications, collaborators, financial services as well as experts [6].

One of the several RS objectives is to assist consumers in managing a massive amount of data [7]. RS increases EC sales by changing visitors to purchasers, presenting new products, nurturing consumer loyalty, improving customer pleasure, and raising the likelihood that a satisfied client would return. According to studies, tailored RS improve sales by up to 35% through recommended items, making them an essential part of the electronic industry. [8]. As a result, in the E-Comm era, these systems are becoming increasingly important as a strategy for executing person-to-person marketing. The Collaboration Filter (CF) is a popular and widely implemented RS [9], and this is considered one of the most significant elements of good E-Comm systems [10]. The absence of cold-start capabilities, scalability, invariance of the user-item matrix [11], and the shift in consumer preferences over time [12] are all downsides of CF. One of the most severe issues with CF is a cold-start, which occurs when the system lacks sufficient data to make reliable recommendations. The characteristics of the items are redundant for CFM. A feature vector or embedding is used to describe each user and item. It automatically makes embedding for both users and items. It does so by placing users and objects into the same embedding space. While recommending a specific user, it takes into account the reactions of other users. It uses a user's likes and the likes of similar users to recommend items. Recommendations are based on the consolidated opinions of the system's users. When newly added items in a catalogue don't have any or very few interactions, this is known as the cold-start problem. Collaborative Filtering Methods (CFM) are highly susceptible to this issue because they rely on the item's relationships to take decisions.

With the growth of forums, blogs, and OSN, millions of individuals can instantly find opposing viewpoints on a given topic. Users can, for example, share their recent experiences, express their opinions on a particular fact, or praise or criticize a product they have recently purchased. Sentiment analysis (SA), a method for categorizing text-based viewpoints to determine a user's attitude, can help to gain a better understanding of the public's attitudes, opinions, and emotions in a variety of domains, including business, government, and biomedicine, thanks to the vast amount of data available online. Analysis of customer sentiment is a potential solution to this problem that should be investigated by online retailers. In order to understand the attitudes, opinions, and emotions expressed in an online mention, SA is the process of identifying the underlying emotional message behind a string of words. Retailers and brands alike can better understand the opinions of customers by using SA to decipher user-generated content and customer feedback. Does it have a positive or negative connotation? What is their perspective's context? Are they referring to the entire product or just a specific feature? Recent advances in text and SA have facilitated the ability not only to realize the brief history and tone of a wide range of opinions but also to critically analyse these opinions for granular details. Due to the overwhelming volume of customer feedback, more and more businesses are turning to SA.

Several research on the benefits of SA in gathering feedback and determining client interests and opinions are reviewed and addressed in [13]. SA has the potential to help RS. Preethi et al. [14] provide an example of this, wherein a cloud-based RS analyses sentiments of reviews to enhance and authenticate restaurant and movie

suggestions using recursive neural networks. SA is an effective technique for commodity markets, in addition to behavioural analysis [15]. The subject of obtaining ideas from online user-generated writings SA—has recently been a popular research area. User information has been used in a variety of ways to solve problems, most notably in CFM [16]. The feature vectors of the item and the user's preferred feature vectors from previous records are used to determine the similarity metrics that are used during the recommendation. Then, only a select few are recommended.

The data of other users are not required in order to make recommendations to a single user when using content-based filtering.

*CFM* Sharing the items' characteristics is optional in a collaborative effort. There is an embedding or feature vector that describes every user and object. The system generates embed codes for users and products automatically. In the same embedding space, it embeds both users and items. While recommending to specific user, it takes into account the reactions of other users. In order to make recommendations, it keeps track of what a user likes and what other users with similar habits and tastes like. Recommendations are based on the aggregated opinions of the system's users.

The Hadamard Product (HP) (element-wise product) produces a new vector with the same dimension as the original operand vectors by multiplying their corresponding components without aggregating by summation. The corresponding parts are multiplied, and the outcome products are added to the dot product. Under Hadamard multiplication of two  $P \times Q$  matrices, the identity matrix is a  $P \times Q$  matrix with all elements equal to 1. While under conventional matrix multiplication, only the main diagonal elements of the identity matrix are equal to 1, this is not the case when inverting the matrix. The HP transform two matrices of the same dimension into another matrix of the same dimension where each element  $[a, b]$  is the product of the elements  $[a, b]$  of the two original matrices. Linear is the HP.

We suggest an RS which integrates SA and CF in this study. The method is applied in an adaptable RS architecture that includes Deep Learning (DL)-based SA. The results of an empirical investigation using two standard datasets suggest that integrating DL-based SA and CF approaches increases the performance of the RS significantly.

The paper is organized as follows: Sect. 2 presents the literature survey, which is followed by the proposed model in Sect. 3; Sect. 4 presents the results and discussion, followed by the conclusion of the paper in Sect. 5.

## 2 Literature review

By utilizing textual reviews, the SA strategy appears to be among the worth-exploring alternatives. According to SA research, corresponding ratings have the ability to improve sentiment accuracy [17]. As per [18], a SA method can be used to infer users' preferences from textual reviews and then transfer those favourites presented as arithmetical ratings that could be interpreted as well as combined into the general CF procedure. Ricci et al. [19] similarly mentioned the overview and problems of RS, which leads to the solution of using SA in RS. The following are among the work that uses SA in RS. A business can learn how customers feel about the

brand and its offerings through SA. Companies can use this information to tailor their sales tactics to customers' opinions. SA on ONS sites is undertaken for one primary reason: to gain valuable business insights that can be used to increase the company's bottom line. Many areas of the business can receive support from SA's impact on learning outcomes. Online retailers benefit significantly from SA. Automatically gather many online opinions. Feature User-Generated Content (UGC) on your E-Commerce site that offers insightful advice to customers. More than just star ratings, provide a deeper level of understanding.

The polarization of adverbs is used by [20] to determine the polarity of phrases retrieved from social networks. The method is implemented in an RS that delivers positive messages to people using phrases taken from a social network that are polarity-evaluated as unfavourable. In an attempt to improve the rating predictions of RS, [21] proposed combining or fusing three sentiment components, namely user sentiment similarity, interpersonal sentiment influence, and item's reputation similarity. Peleja et al. [22] is another effort worth reviewing? By combining updated comments and movie ratings, they proposed an RS for TV on the internet. In this scenario, unratable movie ratings are used to infer ratings from the reviews, which are afterwards merged along the existing user-reviews matrix. In order to improve suggestion outcomes, [23] incorporate the user's biological status as a context parameter. The physiological context of such a user is determined by the user's physical signals, which determine their emotional state. As a result, the ECG signal, which measures Heart Rate Variation (HRV), helps to identify a user's specific biological circumstances. They're a step up from the sparse representations used in the rudimentary bag of words models. Text data can be used to learn and reuse word embeddings. They can also be learned by fitting a neural network to text data. This segment discusses how to use word embeddings in Python with Keras. The Embedding layer in Keras supports word embeddings, the process for learning a word embedding while neural network fitting and in a neural network, how to use pre-trained word embedding. In the field of analytics called SA, massive text databases from various industries are used to glean helpful information. This provides a thorough overview of SA and proper techniques for E-Commerce, which seeks to understand what customers think of its goods and services. SA is the fundamental concept to be explained because it has become a widely known approach to comprehending the emotional states of customers. Methods of SA, such as those based on a lexicon and monitored ML, are also presented, along with the typical steps performed in each. On the other hand, the retail business frequently uses SA, especially in E-Commerce, to help its operators serve their customers more effectively.

An innovative strategy was provided by [24]. By measuring the novelty of a song by using the "Forgetting Curve" and evaluating "preferences" using the client log, the system aims to offer songs that are favoured by the users, fresh to the user's ear, and suit the user's listening habit. In order to establish the user's level of interest in the following song, the user's listening pattern is examined. So, when the cold-start problem for a new customer is not solved, a random song is chosen, according to an analysis of this paper based on the RS challenges. Scalability of the number of songs and diversity of client's tastes (diversity) resulted in increased time complexity owing to the requirement for numerous calculations. This study [25] provided an



effective recommendation based on user behaviour. A desire matrix was created to collect customer preferences related to product attributes since consumers communicate their ideas implicitly depending on product attributes. Balanced association rules are also used to find these trends for improving the value of recommendations. An introduction of a customer behaviour-based RS that determines users imply that product attribute ratings are the most significant addition.

Moreover, to increase the quality of recommendations, this system adopts a sequential buying pattern. Without addressing the rest of the RSs' difficulties, this work focuses on fixing sparsity and uses product attributes to quantify similarity among them. Because not all users have similar patterns, relying on product similarity is insufficient to ensure that a good recommendation is made based on the customer's preferences. A new RS of comprehensive online shopping websites has been presented [26]. The desired degree for each item is determined using the client's context data, like access, click, read and buy data; goods with greater preference degrees are then offered to the customer. The remaining operations, like rating, browsing and searching, were not relevant enough to be examined in this study; therefore, only the buy and click behaviours were considered. It was only able to solve the sparsity problem, leaving the other issues unsolved. A personalized RA is based on a user's implicit feedback [27]. BUIF considers not just the user's purchase habits but as well as item comparisons and orders by them. The consumer's purchase behaviour, comparison behaviour, as well as item orders are all recovered from the consumer's behaviour record; the consumer's comparison is calculated by means of buying and comparison behaviour, and then the item's comparison are calculated by extended word-enrooting to item-enrooting. The issue of diversity, as well as scalability, can be solved using similarity measurements, but the other issues remain unsolved.

Several academic researchers [28], Kumar et al. [29] presented the approaches for using SA in RS [30]. SVM, CNN, RNN, as well as DNN, are some of the approaches that can be used to perform SA. RS depend on detailed user ratings; however, in a growing number of domains, this is not possible [31]. Furthermore, when detailed ratings are given, the ratings' trust and reliability might restrict the RS [32]. While having huge opinions as well as feedback on such goods, assessing feelings inside the word to gather relevant responses in association with conventional item reviews is beneficial, which aids in improving user recommendations. Although the research mentioned above incorporates SA in their recommendation approaches, the majority of them use traditional sentiment methodologies or a single DL model [33].

In order to determine how to suggest an item, CF recommendation uses consumer response in terms of reviews to things in a certain field besides exploits similar in reviewing behaviour between different users. The item and user-based techniques are the two most used approaches in CF [34]. However, despite CF's reputation, CF was unable to recognize user preferences in cold-start conditions to a certain extent. By utilizing social tagging in their works, [35, 36] have offered a strategy for addressing the cold-starting suggestion issue. They employed a CF technique based on user tag expansion. Tagging, for example, has the potential to improve discovering comparable users, but assuming that people are interested in the tagged items is naive. Content-Based Filtering (CBF) approaches rely heavily on the field of information



retrieval, in which metadata and content of documents are used to choose documents that are relevant to a user's query. Recommendations are made in the context of RS by matching the representation of information explaining a product to a representation of content that a user is interested in [37]. By learning more about a consumer or a product, like the director, actor and genre of a movie, CB recommendation can provide a more personalized recommendation.

Hybrid techniques are being explored to enhance the abilities of CB and CF. The basic example of a hybrid is to build unique ranked lists of suggestions for both methodologies and thereafter combine its results to provide a Recommendation List (RL) [38]. However, while confronting the vast measurement of the consumer product matrix, ratings have long been plagued by a phenomenon known as data sparsity. As a response, few research studies have been conducted to address the aforementioned issue [39]. One way by employing consumer favorites indicated in documented ratings is an issue called SA, later putting those favorites into a reviewing level, which CF algorithms could understand, and then it is processed [40]. Bidirectional Encoder Representations from Transformers (BERT)'s advancement of technology is the application of bidirectional training of Transformer, a well-liked attention model, to language modelling. Previously, researchers looked at a text sequence from left to right or a combination of left-to-right and right-to-left training. The paper's findings show that bidirectionally trained language models have a better sense of language context and flow than single-direction language models [41]. We desire to use the latest feature extraction methods and hybrid DL algorithms to SA in this project, leveraging the benefits of BERT in order to include sentiments within RS for added responses and therefore increase the action and consistency of RS [42].

### 3 Proposed methodology

#### 3.1 A framework for recommendation of sentiment-enhancement

This paper adopts CF as well as content-based hybrid RS, as previously stated. The CF and CB techniques can compensate for each other's flaws, guaranteeing that the RS is accurate and stable. On the one side, CF can prevent the loss of personalization of the CB technique; but on the other side, CB can compensate for the defect in the CF method's scalability. Generally, the hybrid RS is used to generate a preliminary RL using user and product data. The preliminary list is then optimized using SA to obtain the final RL. Moreover, this research extensively addresses the effectiveness of the RS based on the hybrid RS. This study focuses on user reviews as part of the item recommendation process. Users are influenced by the herding effect to choose the services or goods that the majority of people like. As a result, rather than items with a lot of negative ratings, goods with a lot of positive ratings will be suggested to users first [43]. The final RL is formed after optimization, as seen in Fig. 1.

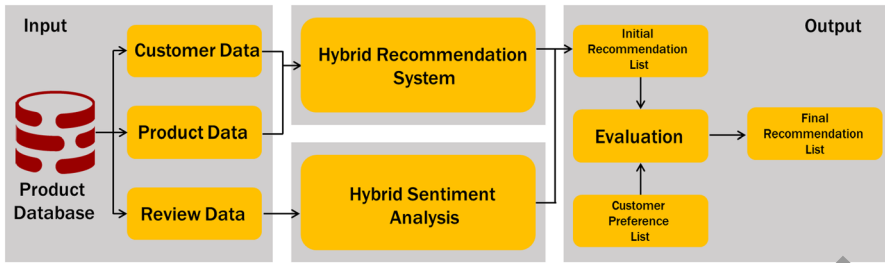
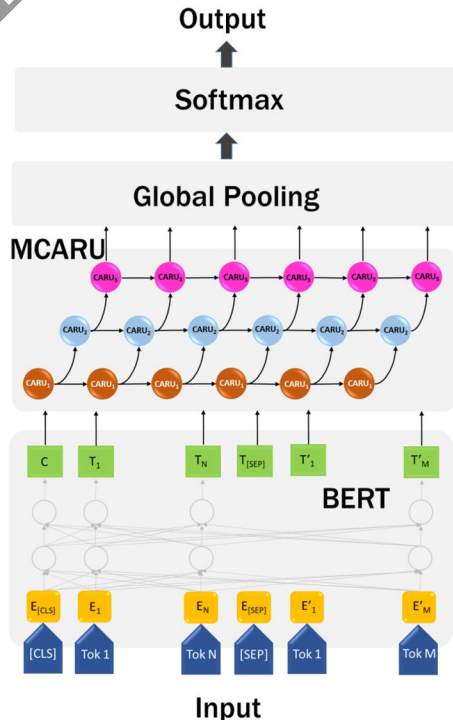


Fig. 1 Proposed HSA-based HRS model

### 3.2 Hybrid sentiment analysis (HSA)

The Bert-MCARU-GP-SoftMax model of SA on E-Comm item value ratings is investigated in this study. During the pre-processing step, the sentiment Bert model extracts the feature as an input layer. MCARU's hidden layer accomplishes dimension-oriented SA by using Global pooling (GP) for storing the long-standing needs that is present in the text irrespective of distances and occurrences. As per the attention mechanism, SoftMax's output layer estimates sentiment polarity through aggregating smaller weighted dimensions. The Bert-MCARU-GP-SoftMax model's structural chart is shown in Fig. 2. There has been much effort put into finding the

Fig. 2 Proposed HSA model



most effective lexical information because the polarity of the text is determined by the score given to each lexicon. When evaluating the subjectiveness of a sentence, Hatzivassiloglou and Wiebe<sup>1</sup> reported an accuracy of over 80% using lexicons consisting solely of adjectives that had been hand tagged. Using the same methodology, Kennedy and Inkpen<sup>2</sup> reported a significantly lower accuracy rate of 62% on a dataset of film reviews. By using an online search engine, Turney<sup>3</sup> was able to determine the polarity of different words [44]. In two separate searches on the AltaVista search engine, Turney<sup>3</sup> used the same target word in conjunction with the words “good” and “bad”. The search result with the highest number of results determined the polarity of the target word. A accuracy of 65% was achieved through the use of this method.

### 3.2.1 BERT model

The transmission learning-pre-training Neural Network (NN) model is the Bilateral Encoder Representations from Transformers (BERT). To acquire contextual relationships between words as well as tokens ( $tok_1, \dots, tok_N$ ) in the text, the BERT uses bidirectional transformer training. The transformer has two mechanisms: an encoder ( $E_1, \dots, E_N$ ) that interprets the text input, as well as a decoder that produces task predictions. The transformer encoder ( $T_1, \dots, T_N$ ) reads the complete bidirectional sequence of words at once, unlike single-direction models like Recurrent Neural Network (RNN) and LSTM, which read the text input sequentially (L-to-R/R-to-L). It has led to a debate in the ML community by delivering front-line results in an extensive range of Natural Language Processing (NLP) and other applications [45]. The multi-level structural chart of the BERT model is shown in Fig. 3.

BERT uses Transformer, an exciting technique that learns context semantic relationships (or sub-words) in such a text. Basically, there are two distinct methods in Transformer: encoding for interpreting the input text and decoding to generate job anticipation. Since the main objective of BERT is to build a language model, there is a requirement for the encoder technique alone. Almost a complete word array is read by the Transformer encoder, which is not similar to directional models that perform sequential interpretation of the input text (left-to-right/right-to-left). Consequently, the classification is bidirectional; however, the apt word is non-directional. The model is, however, enabled to deduce the context of a word from its surroundings (a word on left and right). Prior to the analysis done by the NN, the input comprises an array of tokens that are embedded into vectors. An array of vectors, in which each of them is correlated with an input text of identical index, is the consequence of an H-dimensional series of vectors. Two training methods are used by BERT in order to fulfil this challenge:

**3.2.1.1 Masked LM (MLM)** Prior to the feeding of with such a token [MASK], it is substituted by 50% of the words for each word sequence. In the context of non-masked words in the sequence, the next attempt of the model is to anticipate the actual value of the disguised words. In the technical context, the prediction of output word prompts:

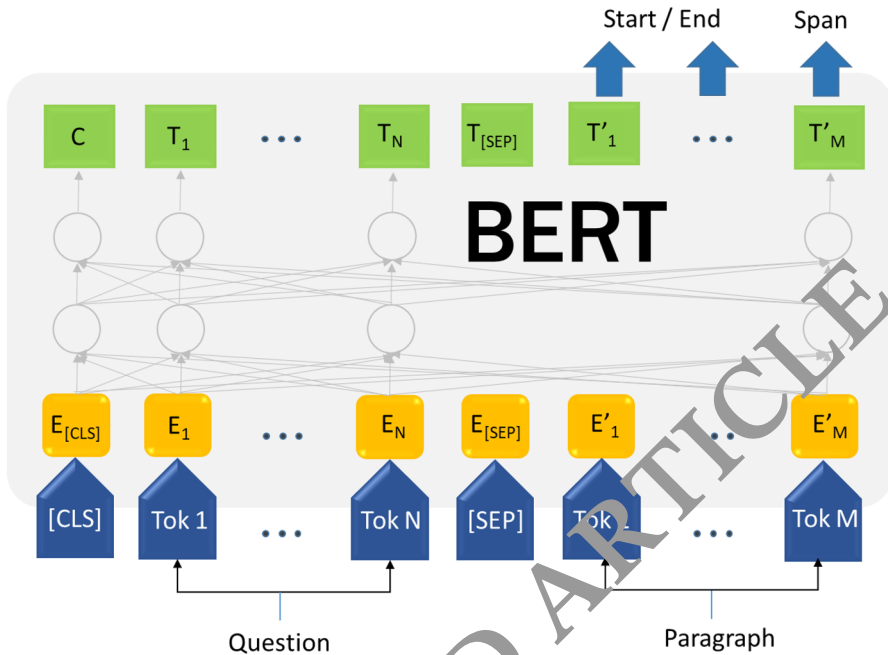


Fig. 3 BERT structure

1. There is an addition of a classification layer on top of an encoder output.
2. The embedding matrix is used to multiply the transformation of the output vectors further into the vocabulary dimension.
3. The similarity of every individual word in the dictionary is calculated using SoftMax.

The BERT loss function, which discards the non-masked words' prediction, considers only the masked values' prediction. Consequently, when compared to directional models the convergence of the model is very slow; however, its higher context awareness encounters this.

**3.2.1.2 Next sentence prediction (NSP)** The training approach of BERT includes feeding the sentences' model pairs and learning the art of predicting whether the pair that contains the second sentence is the following sentence in the actual document. A pair is 50% of inputs during training, whereas the following sentence in the original text is the second sentence, and the remaining 50% from the corpus is a random sentence. Indeed, it is assumed that the random sentence is detached from the first sentence.

Prior to getting exposed to the model, in the following way, the input is handled to assist the model in order to get distinguished between both the two sentences during training:

1. The insertion of A [CLS] Cumulative Layout Shift token occurs at the start of the first sentence, but a [SEP] Significant Economic Presence token is inserted at the end of each sentence.
2. The discovery of whether each token containing a sentence embedding is Sentence A or B is done. As per the concept, sentence embeddings and token embeddings are identical to each other and also in those with two vocabularies.
3. The place in the sequence can be denoted by each token since it is provided with a positional embedding. The positional embedding's concept and execution is explained in the Transformer paper.

In order to find out whether the second and the first sentence is actually linked with each other, the following steps are carried out:

1. For processing the absolute input sequence, the use of the Transformer model is crucial.
2. For utilizing a superficial classification layer, the transformation of the [CLS] token's output into a 21-shaped vector (biases and weights' learned matrices) is done.
3. For evaluating the probability of IsNextSequence, SoftMax is used significantly.

The merging of Masked LM and Next Sentence Prediction is done to train the BERT model, correspondingly reducing the two techniques' combined loss function. The collected features are then sent into the next layer, the Multi Content-Adaptive Recurrent Unit (M-CARU), which uses multiple clause-rules recognition to handle sentiment word disambiguation problems. The purpose of CARU is outlined below:

### 3.2.2 Content-adaptive recurrent unit (CARU)

Chan et al. [46] introduced CARU, a version of GRU, in 2020. The CARU has the same update gate as the GRU, but instead of the reset gate, it has a Content-Adaptive Gate (CAG), the adaptable interface where the product catalogue's customization options are displayed. During the design phase, the adaptable interface was implemented to ensure that the predefined contents were locked down and secure.

Users can change the content presentation to their liking. Content-adaptive input modulation is element-wise rather than input-wise. Without accepting the user's preferences into account, the products are displayed in this interface. This motivates the user to spend time customising the product catalogue in order to search for desired products more efficiently. Customizing the product catalogue includes changing the list's order, moving less-preferred products to the end, and repositioning a product.

CARU was created to solve the problem of RNN models' long-term reliance. It was revealed that it accomplishes somewhat improved NLP responsibilities besides fewer parameters than with the GRU. In the following equations, the lowercase variables represent vectors and  $[W;B]$  represent the training parameters, which are linear layers consisting of biases and weights, respectively. Primarily, for  $b = 0$ , CARU

directly returns  $h^{(1)} \leftarrow W_{vn}v^{(0)} + B_{vn}$ ; Next, for  $t > 0$ , a complete architecture is given by Eqs. (1), (2), (3), (4), and (5):

$$x^{(t)} = W_{vn}v^{(t)} + B_{vn} \quad (1)$$

$$n^{(t)} = \phi((W_{hn}h^{(t)} + B_{hn}) + x^{(t)}) \quad (2)$$

$$z^{(t)} = \sigma(W_{hz}h^{(t)} + B_{hz} + W_{vz}v^{(t)} + B_{vz}) \quad (3)$$

$$l^{(t)} = \sigma(x^{(t)}) \odot z^{(t)} \quad (4)$$

$$h^{(t+1)} = (1 - l^{(t)}) \odot h^{(t)} + l^{(t)} \odot n^{(t)} \quad (5)$$

It's important to note that this  $t \leftarrow t + 1$  exists at the end of every recurrent loop. The operator  $\odot$  specifies the Hadamard product,  $\sigma$  and  $\phi$ , the sigmoid and hyperbolic tangent activation functions, respectively.  $\tanh$  (also “*tanh*” and “TanH”) is an abbreviation for the hyperbolic tangent activation function, which is also referred to as simply the *Tanh* function. Its S-shape is reminiscent of the sigmoid activation function, with which it holds many similarities. Inputs of any actual number are used to generate results between -1 and 1. When the input is more significant (has a higher positive value), the value of the output will be closer to ‘1.0’. When the input is more minor (has a higher negative value), the output will be closer to -1.0.

- $x^{(t)}$  It commences by projecting the current word  $v^{(t)}$  into the feature input  $x^{(t)}$ . That outcome will be forwarded to a suggested CAG in the following concealed state.
- $n^{(t)}$  In contrast to GRU, the resetting gate has been removed. It simply combines the  $h^{(t)}$  as well as  $x^{(t)}$  parameters for creating a current unseen layer  $n^{(t)}$ .
- $z^{(t)}$  This is similar to the update gate of GRU, and it's used to switch between hidden and visible states.
- $l^{(t)}$  The update gate and the weight of the current feature are combined using a Hadamard operator. Instead of diluting the current concealed state, we call this gate the CAG because it will alter the quantity of gradual transition.
- $h^{(t+1)}$  The following hidden state is developed by integrating  $h^{(t)}$  and  $n^{(t)}$ .

The whole diagram of CARU is depicted in Fig. 4.

**3.2.2.1 CARU procedure** As shown in Algorithm 1, a CARU unit could be made up of many Linear layers. The Linear layer, which concentrates on state and weight calculations, processes the hidden state as well as the current input feature first, and the results are then sent to the CAG for the hidden state transition. The gate has been used to integrate all of the weights collected and directly construct the next concealed state. It should be noted that we implemented two Sigmoid functions (line 12 of

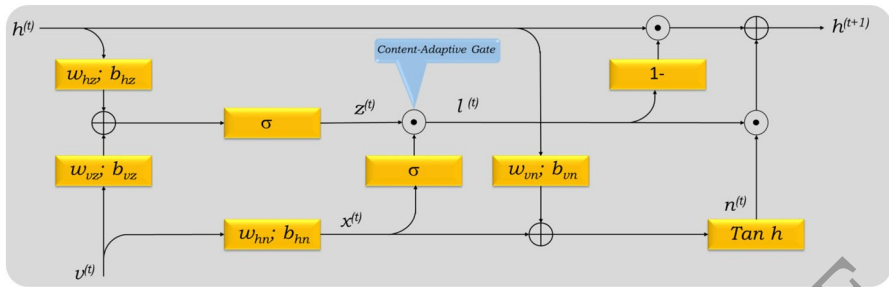


Fig. 4 CARU Architecture

Algorithm 1) to assure that the output state can be confined in the range of  $(-1, +1)$ , giving a more stable characteristic for later processing.

Algorithm 1: CARU Pseudocode	
Step 1:	Constructor (inputsize, hiddensize)
Step 2:	{
Step 3:	Int I := inputsize, O := hiddensize;
Step 4:	$\mathcal{L}w \leftarrow \text{Linear}(O, I)$ ;
Step 5:	$\mathcal{L}l \leftarrow \text{Linear}(I, O)$ ;
Step 6:	Weight $\leftarrow \text{Linear}(O, O)$ ;
Step 7:	}
Step 8:	Forward (word, hidden):
Step 9:	{
Step 10:	feature $\leftarrow \text{Linear}(\text{word})$ ;
Step 11:	$n \leftarrow \text{Tanh}(\text{Weight}(\text{hidden}) + \text{feature})$ ;
Step 12:	$l \leftarrow \text{Sigmoid}(\text{feature}) \odot \text{Sigmoid}(\mathcal{L}w(\text{hidden}) + \mathcal{L}l(\text{word}))$ ;
Step 13:	Return $((1 - l) \odot \text{hidden}) \oplus (l \odot n)$ ;
Step 14:	}

### 3.2.3 Global pooling (GP)

Global pooling (GP) is a pooling operation designed to replace fully connected layers in classical CNNs. In the final MLP CONV layer, one feature map will be generated for each corresponding category of the classification task. We do not use fully connected layers on top of the feature maps; instead, we average the feature



maps and feed the resulting vector into the SoftMax layer. When compared to fully connected layers, the GP has the advantage of being more native to the convolution structure because it enforces correlations between feature maps and categories. This is one of the obvious benefits that the GP has over fully connected layers. This means that the feature maps can be acknowledged as a form of confidence mapping for different categories. Since the GP does not contain any dimensions that need to be optimised, there is no risk of overfitting at this layer. GP is also more susceptible to spatial language translation of the input because it encapsulates the spatial information.

The GP mechanism combines non-linear activation as well as pooling computations into a single, straightforward pooling action. Quadratic calculations are used to activate each of the characteristics. As an outcome, the value of essential qualities rises, whereas the value of unimportant features falls drastically. The relationship between every user and OS component is the key consideration in the collaborative approach. Here, user consensus is used to determine what constitutes a “close enough” comparison between two items. The name suggests that they work together and exchange data regarding the products. The fundamental idea behind CFM is that users who have previously liked the same item will probably like a related item in the future. When there is little information available about a specific buyer, this filtering is used. In order to combine this data with that of other users who share similar preferences, the algorithms gather information such as evaluations and past purchases. This leads them to infer which products he might actually be interested in. The pooling technique gathers all of the candidates for overall context features and determines their moral salience. Thus, it can compute the salient score among each feature candidate to associate items at different points from a single sequence and then adaptively build task-friendly while using entire context feature candidates for practical features. Furthermore, our innovative GP mechanism combines non-linear activation as well as pooling computations into a single straightforward operation that can drastically enhance the value of prominent characteristics while dramatically lowering the value of unlabeled data, alleviating the zero-gradient problem, Eq. (6).

$$\hat{o}_h = \frac{1}{|m-h+1|} \sum_{i=1}^{m-h+1} (o_h^i)^2 \quad (6)$$

The prominence of an  $h$ -gram feature in such a sentence is determined  $\hat{o}_h$  as follows: wherein  $\hat{o}_h \in \mathbf{c}_h$  seems to be the  $i$ th feature vector twisted by a filter of size ‘ $h$ ’. Because the  $h$ -gram channel has one filter, the GP operator is used in the feature map to  $\mathbf{c}_h$  generating a feature prominence vector for the channel, Eq. (7)

$$\mathbf{z}_h = [\hat{o}_h^1, \hat{o}_h^2, \dots, \hat{o}_h^l]^T \quad (7)$$

The feature salience across all channels is then concatenated as the final feature vector that is indicated as  $\mathbf{z} = \mathbf{z}_1 \oplus \mathbf{z}_2 \oplus \dots \oplus \mathbf{z}_h$ . The operation of concatenating two vectors is denoted by  $\oplus$ . A fully linked SoftMax layer receives the representations

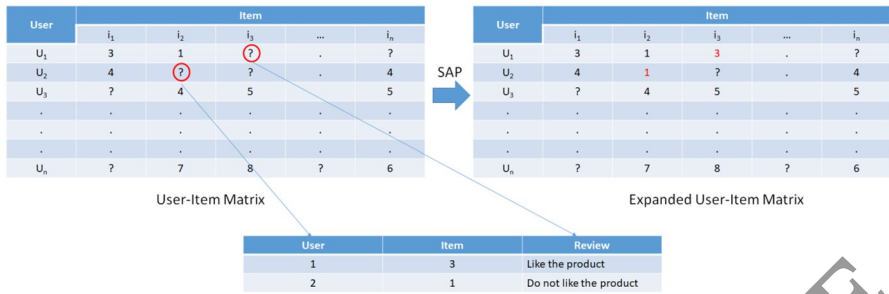


Fig. 5 The extended consumer-product matrix

from the pooling layer. The network has been trained to minimize the cross-entropy between both the true and forecast distributions, Eq. (8)

$$\mathcal{L}_c = - \sum_{i=1}^G y_i \log(\text{softmax}(z_i)) \tag{8}$$

where \$y\_i\$ is the ground truth, expressed by a one-hot vector, and \$L\_c\$ is the cross-entropy loss function. For each class, \$p\_i\$ is the expected probability. We finished the feature extraction using the specified pooling by combining the aforementioned layers.

### 3.2.4 SoftMax with attention mechanism

The consideration mechanism calculates the linear weight amount of entire positive, negative, as well as neutral polarizations for sentence orders from the previous unseen state level, and a SoftMax layer calculates the attention probability as well as sentiment polarity weights of reviews by combining the various semantic features of the Bert-MC/RU-CP models.

The following are SoftMax's detailed approaches to the attention mechanism:

By obtaining the feature vector from the attention layer, the weight matrix is calculated by the attention function, Eqs. (9), (10) and (11)

$$u_t = \tanh(w_a h_t + b_a) \tag{9}$$

$$\alpha_t = \frac{\exp(u_t^T u_a)}{\sum_t \exp(u_t^T u_a)} \tag{10}$$

$$s_{it} = \sum_{i=1}^n a_{it} h_{it} \tag{11}$$

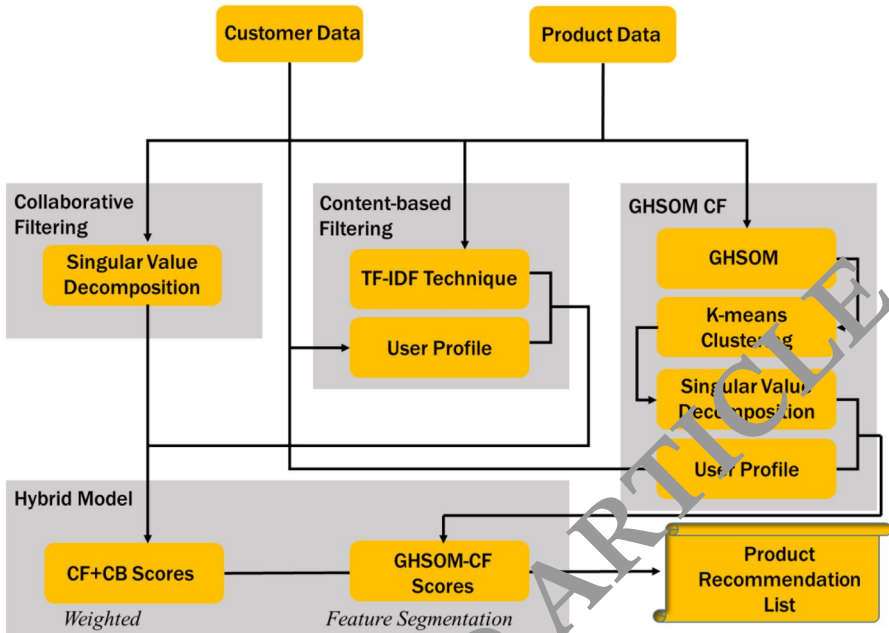


Fig. 6 Proposed HRS model

wherein  $\tanh$  is said to be a sigmoid function, the hidden state ‘ $h$ ’ is supplied to the learning function for forming a probability vector, which is then calculated as mean weight. Use Eq. (12) to do SoftMax’s normalizing operation:

$$y = \text{softmax}(w_a s_{it} + b_a) \tag{12}$$

wherein  $y$  seems to be the SA result and is the output layer’s matching offset.

To calculate the sentiment score for each review, we used the unsupervised ML technique described above. This procedure produces a sequence of reviewing scale data. Together, the set of input information is hence combined to generate an extended consumer-product rating (Fig. 5), which is then filled by sentiment ratings that reflect users’ preferences. The rating matrix, on the other hand, is extremely sparse.

Ratings alone are insufficient for making accurate forecasts and recommending trustworthy things to users. As a result, the SA procedure is used to analyze context data in user evaluations. User  $u_1$ , for example, did not rate item  $i_3$  and then offered a remark, to be seen in Fig. 5. SA helps to decode a textual remark which turns it to scales. The larger user-item matrix is then filled with an arithmetical rating generated by SA, known as a sentimental rating. Such a user-item matrix is supplied into the hybrid RS’ NN layer, which uses it to find items with such a robust predictive score, which can be suggested and presented to the user.

### 3.3 Hybrid recommendation system

An advanced hybrid RS (Fig. 6) that includes four main factors such as SOM-CF, CF, Hybrid Filtering, and CB filtering, are offered in this section.

#### 3.3.1 Collaborative filtering model (CFM)

Here the user-product connections are used as a base for making recommendations. To put it in the other way, the perceptions of the remaining customers are the basis for filtering products and also selecting a group of individuals' who share the likes of a user. The preferred items are examined by the strategy, and a list of recommendations is categorized. On the basis of the comments of a user, the matching items are aimed at by the CF system. There can be explicit and implicit feedback

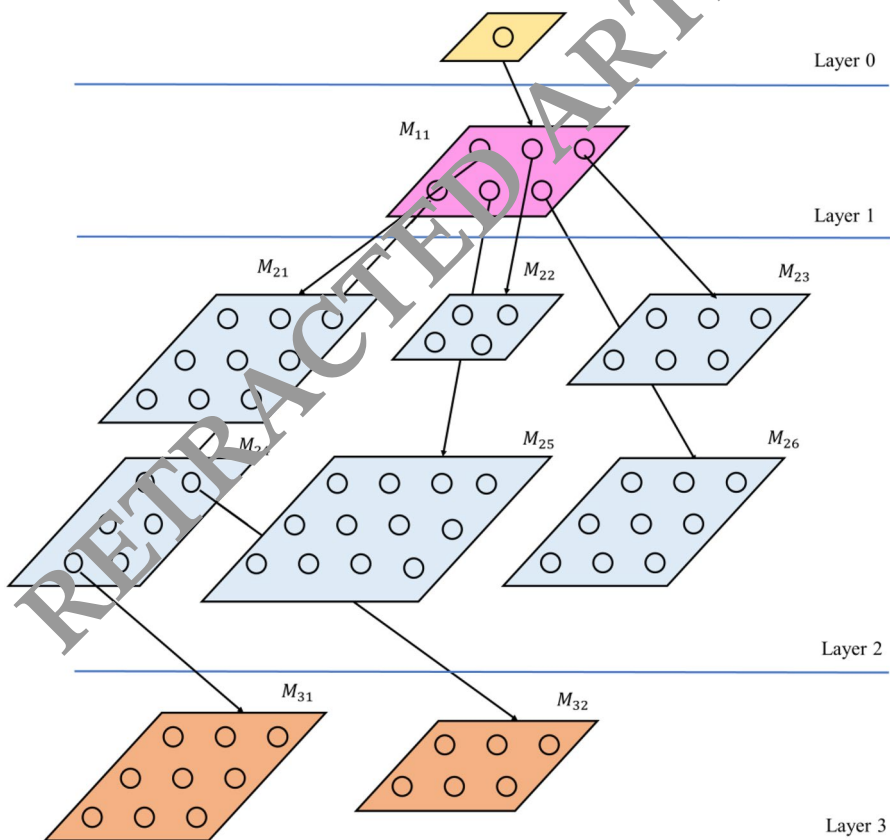


Fig. 7 GHSOM representation

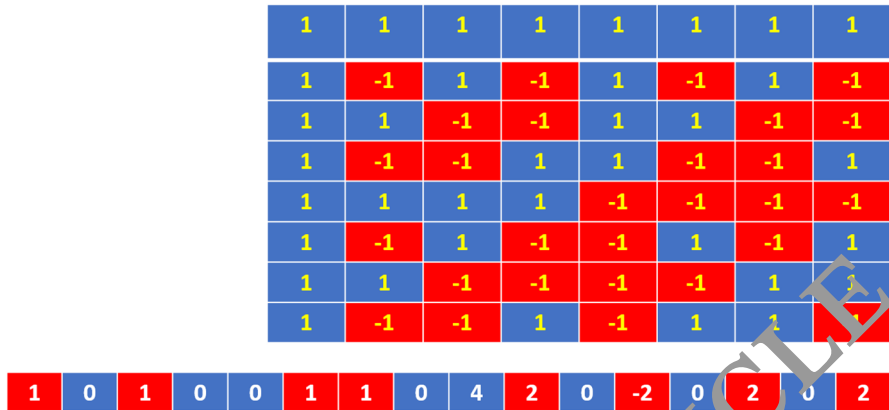


Fig. 8 HM formation

from users, such as a numerical rating for indicating the satisfaction level of consumers on a particular item, such as 1 for detest and 5 for extremely like, or implicit, such as a website’s browsing history or a particular product’s reading time. The two types of CF algorithms are Memory and Model-Based algorithms. In memory, the first save products as well as user data, then applies mathematical techniques to create data-based estimations. The model process is built using a variety of ML algorithms, including the rule-based, Bayesian network, as well as clustering methods. In our approach, the second Model-Based approach is adopted since it is capable of responding to user requests quickly. The method we used to generate a CFM was Singular Value Decomposition (SVD). Having said that, Matrix computations that use all of the values in a matrix are sometimes redundant or take much time to do. To facilitate further calculations, the authors need a description of the matrix that makes it easy to extract its most crucial portion. This is where the SVD becomes relevant. The singular value decomposition, or SVD, is a matrix factorization technique that decomposes any matrix into three standard and well-known matrices. It has some exciting applications in ML and Image Processing. Knowledge of eigenvalues and eigenvectors is essential for understanding the concept of SVD. Scroll down a little to experience the SVD if you’re familiar with eigenvalues and eigenvectors. By reducing the space dimensions from A to B, the SVD method minimizes the features’ count in a dataset in which B is more significant than A. Deconstructing a matrix into three other matrices is SVD’s fundamental purpose, Eq. (13), because the matrix factorization part of RS the maintains identical dimensionality is our main concern:

$$X = U \times S \times V^T \tag{13}$$

In our scenario,  $M \times N$  utility matrix is  $X$ , and an  $M \times R$  orthogonal’s left singular matrix is  $U$ , which depicts the link between consumers and latent components,

**Table 1** Dataset statistics

Parameters	Amazon fine foods reviews	Amazon movie reviews
Number of reviews	568,454	7,911,684
Number of USERS	256,059	889,176
Number of Products	74,258	253,059
Users with > 50 reviews	260	16,341
Average No. of words per review	56	101
Timespan	Oct 1999–Oct 2012	Aug 1997–Oct 2012

with the latent factors being item features. The similarity between elements and latent components are represented by  $V$ , which is an  $R \times N$  diagonal right singular matrix. The intensity of every latent factor is defined by  $S$ , which is an  $R \times R$  diagonal matrix. To build our CFM, we employ a matrix structure, with each row and column representing a consumer and an item, respectively, and the components of this matrix representing the ranking data is an item contributed by the consumer. By using a dot product of  $U$ ,  $S$ , and  $V$ , the SVD methodology is then used to obtain all of the expected user ratings, Eq. (14). In order to test our model, a user is taken, and his entire projected ratings are rated for the purpose of recommending the top- $N$  products.

$$\hat{r} \approx U \cdot S \cdot V^T \quad (14)$$

### 3.3.2 Content-based model

“Content” refers to the characteristics and content of the goods sought after. CBF’s goal is to group items with similar features, taking into account the user’s preferences, and then search the dataset for those phrases. Finally, we recommend a variety of things that share comparable characteristics. In order for each word to be allocated the value with a description of the item in the dataset, a weight per each word is desired to be added with a description of an item to our CBF model. The keyword is weighted in either document (title of the product) but also assigns value to it on the basis of its number of appearances in the text using the TF-IDF algorithm; being a rarer and more relevant phrase is an implication of a more excellent TF-IDF score (weight), and vice versa. Consequently, in the final step, on the basis of title features, the TF-IDF vector reads each item. One way to account for the fact that some words appear more frequently than others is to use the Term Frequency-Inverse Document Frequency (TF-IDF) value, which intensifies in direct proportion to the number of times a word occurs in the document and is offset by the number of documents in the corpus that contain the word. When it comes to term-weighting schemes, TF-IDF is currently among the most widely used. 83% of text-based RS in digital libraries use TF-IDF, according to a 2015 survey. Search engines frequently employ variants of the TF-IDF weighting scheme as a central metric for scoring and ranking the

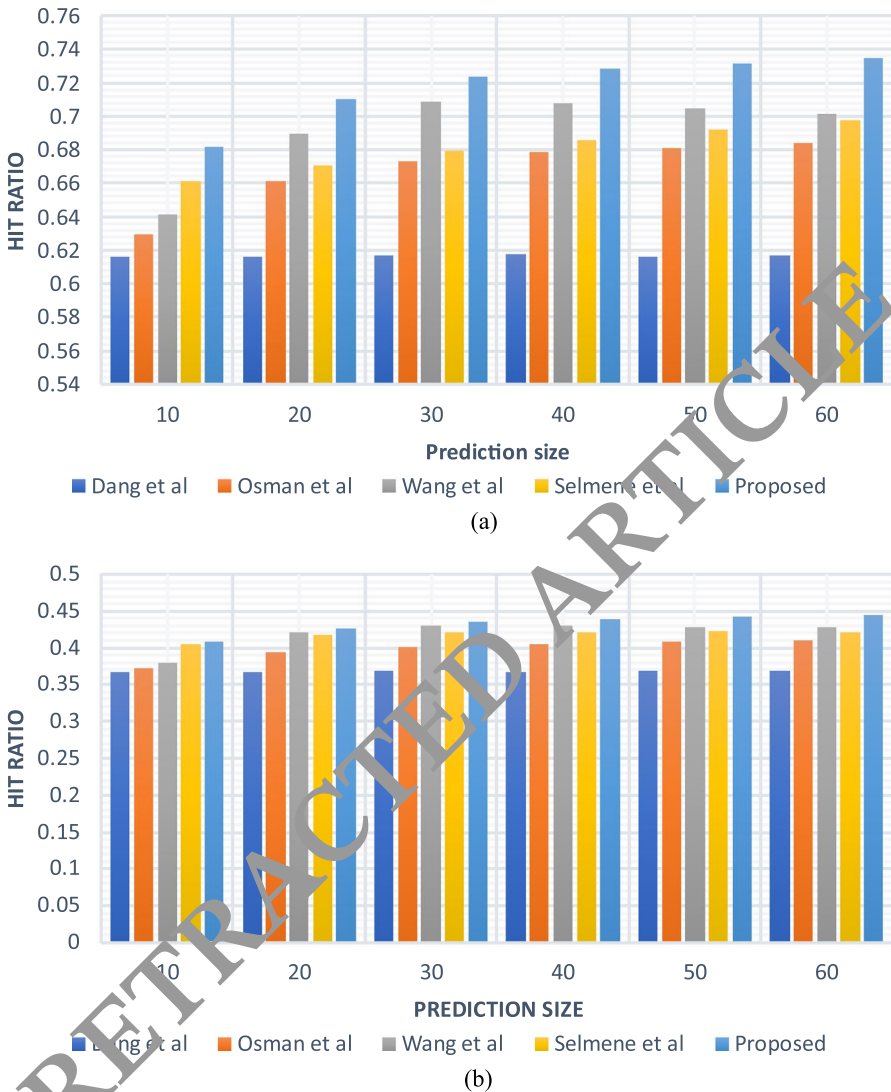


Fig. 9 Performance of a HitRatio@10 and b NDCG@10 on Amazon fine foods review dataset

relevance of a document in response to a user query. For example, in text summarization and classification, TF-IDF can be used to filter out ideal words effectively.

The total number of times a term occurs in the current document is called Term Frequency (TF). This refers to the frequency with which a word appears in a text; as a result, we should normalize the result by dividing it by the duration of the document, as stated in Eq. (15), where  $(T_x, D_y)$  is the appearance of term  $T_x$ 's overall



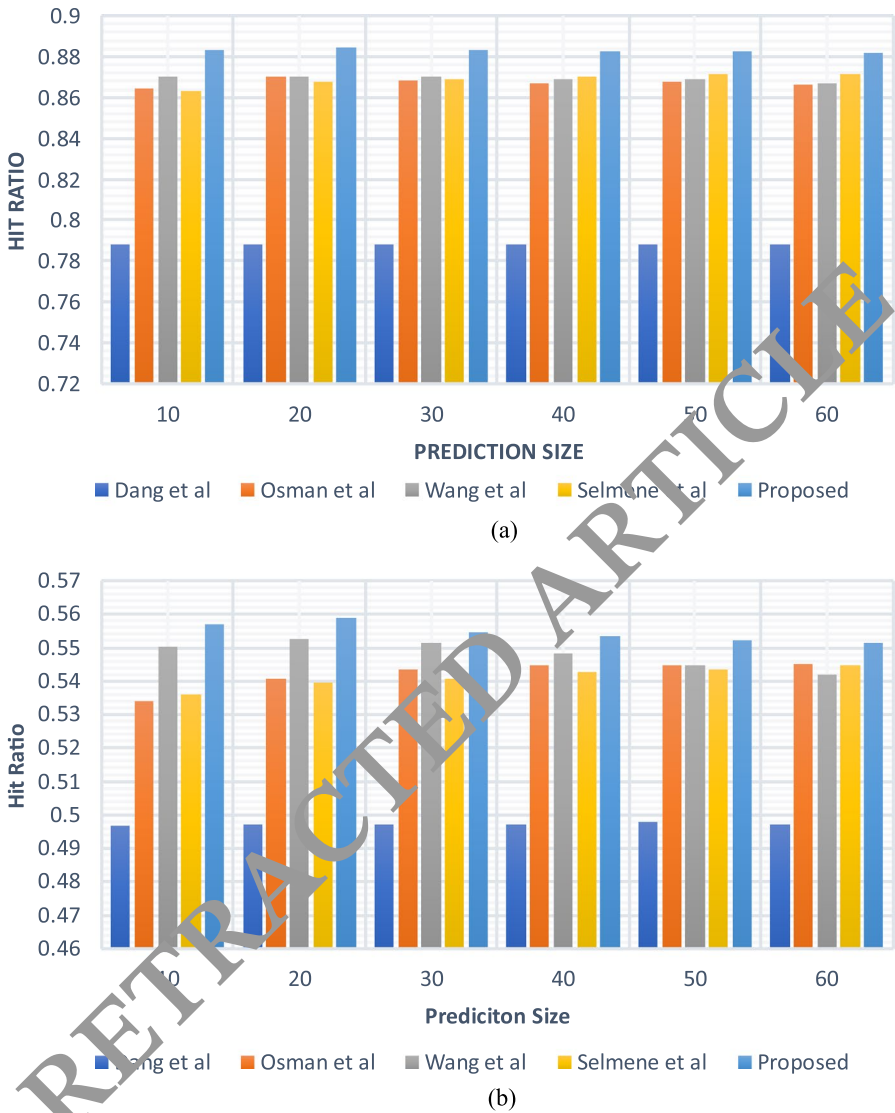
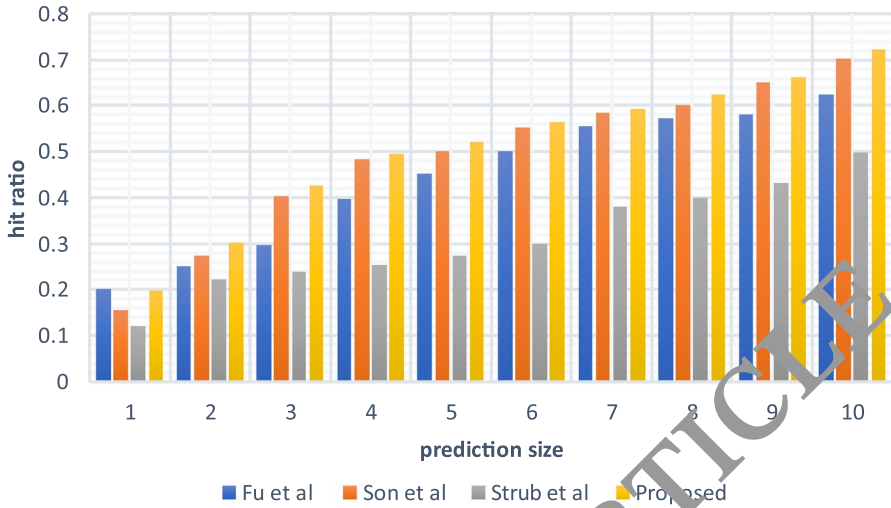


Fig. 10 Performance of a HitRatio@10 and b NDCG@10 on Amazon movie review dataset

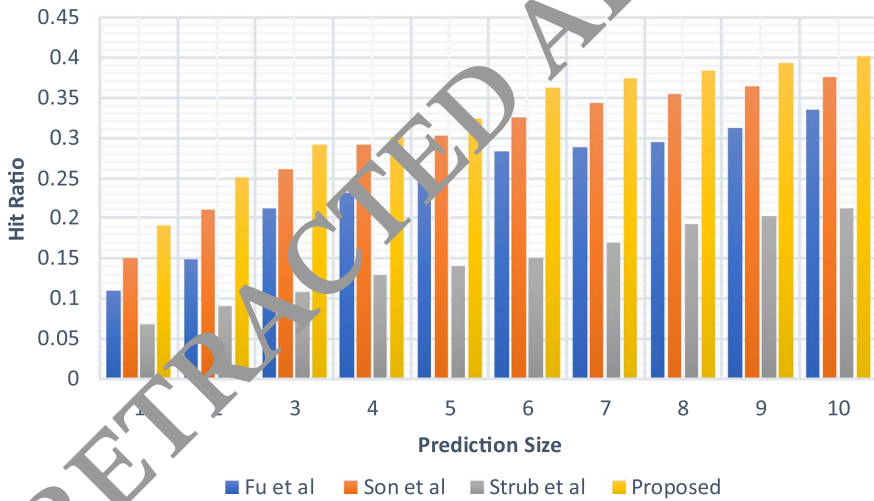
multitude of times in a document. The overall number of words in a document is  $D_y$  and ( $P_y$ ).

$$T(T_{x,y}) = N(T_x, D_y) / N(P_y) \tag{15}$$

The total number of records that includes the word 'x' is Inverse Document Frequency (IDF); the IDF reduces as the word that appears in the text, indicating its



(a)



(b)

Fig. 11 Top-K item recommendation on Amazon fine foods review dataset

rarity. It helps to determine the meaning of a word throughout the entire corpus. For a better understanding of the IDF, consider the following scenario: if you search for “the hybrid system” on Google, the *TF* of the word ‘the’ will automatically be more remarkable than “*hybrid*” and “*system*”. In this case, the IDF’s role was to reduce the word “*The*” and its overall weight so as to give the relevant words further weight. The Eq. (16) is used to compute IDF, where the total number of documents is ( $D$ ), and the number of documents that include the term ‘x’ is ( $D_x$ ).

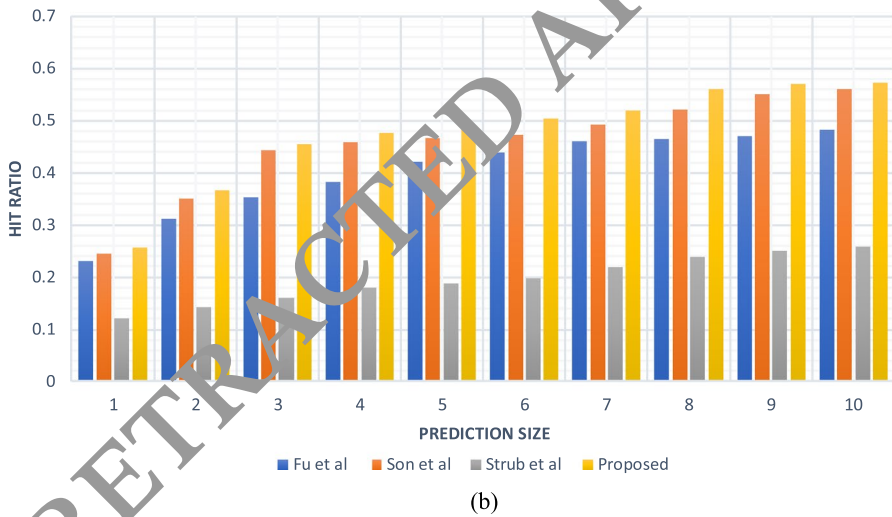
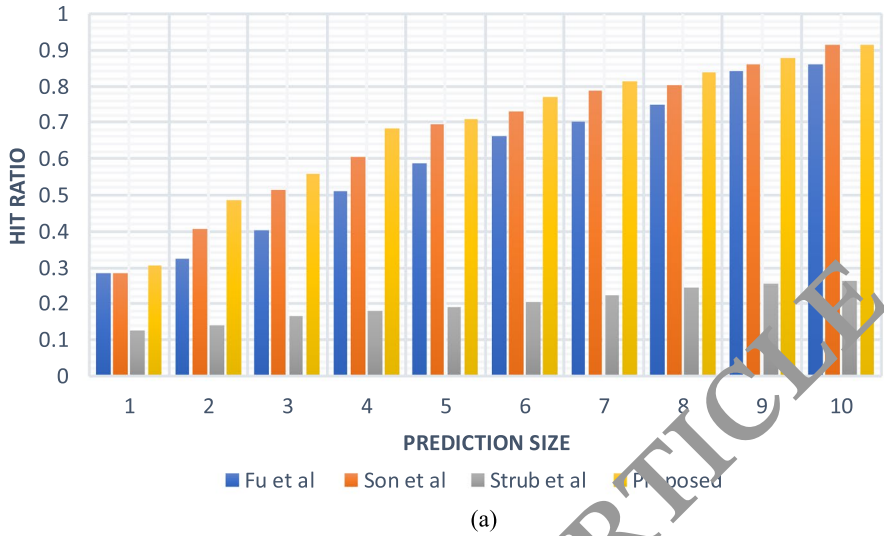


Fig. 12 Top-K item recommendation on Amazon movie review dataset

$$ID(T_x) = \text{Log}(N(D)/N(D, T_x)) \tag{16}$$

Eventually, the TF-IDF is a normalizing method for determining the relevance of a word in a collection of documents. The TF-IDF is calculated using Eq. (17).

$$TF - IDF(T_{x,y}) = TF(T_{x,y}) \times \log_{10}(N(D)/N(D, T_i)) \tag{17}$$

In this case, after the TF-IDF score is calculated word by word for such a product name, a matrix comprising each word is produced along with its TF-IDF score

inside our framework, preventing stop phrases. For each consumer, a profile is generated in the dataset by means of picking the whole goods liked by the user, using TF-IDF ratings for categorizing the words, and then with respective scores that reflect the profile of the user for generating a list of terms.

Currently, the items defined in a Vector Space Model (VSM) comprising every word with both the score of TF-IDF and the same with the profiles of the user are possessed in this study. Here, the cosine similarity approach is used as stated in Eq. (18) between the profile of the user and the products' VSM to locate things related to the user profile as well as give him the suggestion after classification.

$$\cos(\theta) = A \cdot B / \|A\| \cdot \|B\| \quad (18)$$

### 3.3.3 Neural network model

We have the past preferences of users and views on an item set as a base, the process of filtering objects as detailed in the CFM. For the purpose of developing our hybrid system, we will employ the Growing Hierarchical Self-Organizing Map (GHSOM) approach to enhance the classic CFM. A GHSOM is a type of Artificial Neural Network (ANN) taught by means of unsupervised learning for creating a low-dimensional, especially a 2D image. To overcome the problem of unsupervised clusters of the product dataset, we will employ the GHSOM in our model. Clustering technology breaks down the structure of a dataset and splits it into distinct clusters, making it easier for users to view and interpret the data. The fundamentals of GHSOM and their function are presented in the following section.

**3.3.3.1 GHSOM model** Certain restrictions must be solved, and the SOM has been shown to be a capable device for detecting structure for high-dimensional data and arranging it in a 2D output space. One of the constraints is the inability to capture the entire hierarchical data structure. Furthermore, when an accurate comprehension of a data distribution feature is not available, the map size must be determined first. None of these approaches, however, provide an architecture that adapts entirely to the properties of the input data.

The GHSOM has a hierarchical structure with numerous layers in which distinct separate developing SOMs are stored in each layer. As per the Growing Grid Model, beginning with a top-level map, the size of the map increases in order to depict data collection at a specific detail level. It is determined to see if the units accurately represent the data at the most minimal granularity level. Those units that signify various input data produce a new lowest developing SOM at the following layer, where the related data must be specified in greater detail. These new maps will continue to grow in size until a particular improvement in data representation accuracy is achieved. Alternatively, for units that currently signify a similar dataset, any further extension into consecutive layers will be unnecessary. As a result of its architecture,

the resulting GHSOM is entirely adaptable for constructing a hierarchical model integrated into the data while still leaving enough room for the depiction of diverse regions in the input space.

A graphical illustration of a GHSOM is shown in Fig. 7. Layer 1 has  $3 \times 2$  units in the map, whereas layer 2 contains an uneven layout of main clusters inside the input data. The 6 different maps, as in the second layer, provide a comprehensive view of the data. The input data to one map is the subset which has been mapped on to succeeding unit as in the top layer. Further, extending into third-layer maps with two units by one of the second-layer maps, as per the data, provides an acceptable granular input data representation, freeing us from the issue of predetermining the architectural structure. Layer 0 is the representation of the entire data set, and it is critical for managing the growth process.

- (i) *Unit generation* The environment for unit creation occurs when there is insufficiency in the potential of network representation. In case the hidden layer are provided with enough units, NN is capable of often generating any mapping with anticipated accuracy. As a result, a stabilized error is being used as an index to find out the network's requirement for any new unit generation. If the stabilized error is more than the target value in a certain duration, a new unit could be created. The following task involves deciding the location of a new unit inside the hidden layer. We can find the appropriate place by watching the activity of the units inside the concealed layer.
- (ii) *Unit elimination* The preceding observations indicate the succeeding procedures for the unit elimination (i.e.,) the following unit can be eliminated when: @@ In the network, that is not an operational component. @@ That is a redundant member in the network.

### 3.3.4 Algorithm for GHSOM

Dataset I, Map $M$ ,	
Input	:
Step 1:	Define Parameters
Step 2:	Calculate Mean Quantization Error (MQE), for the unit "i" in level 0 of the GHSOM
Step 3:	FOR $M_v < n$ , DO
Step 4:	Use SOM clustering algorithm for each map;
Step 5:	Cluster samples into I subcategories;
Step 6:	Select winner unit 'k' from each subcategory;
Step 6:	END FOR;
Step 7:	Calculate the MQE and Quantization Error (QE) for each winner unit k;
Step 8:	FOR $n \leq u_o$ :
Step 9:	for $m \leq x_o$ ,
Step 9:	IF $MQE(m_k)_m < MQE(C_k)_n$
Step 10:	$e \leftarrow MQE(C_k)_n$ ;
Step 11:	ELSE
Step 12:	$e \leftarrow MQE(m_k)_m$ ;
Step 13:	END IF;

Step 14:	END FOR;
Step 15:	END FOR;
Step 16:	Calculate the MQE using: $MQE_{M_{c,p}} = \frac{1}{n_u} \sum_{k \in u} qe_k$ , $n_u =  u $ , for each subset 'u' of the winner unit k;
Step 17:	IF $MQE_{M_{l,v}} == \tau_1 qe_w$
Step 18:	insert unit in either row/column;
Step 19:	ELSE
Step 20:	return to the above layer of the map;
Step 21:	END IF;
Step 22:	IF $qe_k \geq \tau_2 qe_0$
Step 23:	add a new layer to the map;
Step 24:	END IF;

We train the model after initializing our map dimensions as well as randomly initializing COSM weights. For the K-means clustering model, after the map has been learned, it returns weights, which is utilized as input data. We will utilize the Elbow approach to determine the clusters' optimal number. The Elbow technique states that a number of clusters are graphed against the variance; the number of clusters is the elbow point wherein there is no declining in variance. To make suggestions, the dataset is divided into a particular number of clusters, select all ratings of positively rated items for an active user (and only those rating value greater or equal to 2.5 is considered), and assign each item a cluster number. In order to do this, the distance among the item attributes and the k-means centroids applying the Euclidean distance method are computed as described in Eq. (19). The least distanced cluster is then chosen. By showing the positively scored items with their cluster class, we count the number of every item cluster in the list, and the top result class is utilized as the preferred class of the user. All selected users' matrix and all list items are selected, then the SVD-CFM is used to estimate item ratings, and in order to provide



recommendations, those from best to the worst are sorted out; once the favorite class of the user is acquired, one's age (demographic attribute) is obtained in case greater or equal to 35 years is the age, then the consumers with the same age interval and vice versa is obtained, then all target users' matrix and all selected items are constructed, and then for estimating, the SVD-CFM is used.

$$d(a, b) = d(b, a) = \sqrt{\sum_{i=1}^n (b_i - a_i)^2} \quad (19)$$

Items are sorted by their scores in the preliminary RL depending on hybrid RS. The scores are determined from the CFM as well as content-based RS calculations. The preliminary RL will be optimized by SA. The sentiment value will be incorporated into the item's overall score. The revised score is used to create the final RL. The desired list is a collection of items with no particular order. The final RL would be presented with no order to conform to this scenario. As a result, more products are chosen by hybrid RS in necessary to pick sufficient appropriate items, while some are eliminated by final scores. The RS shows users an optimum set of recommendations.

HRMs combine two or more RS in various way to take advantage of their supplementary benefits. We will use two methods in one in our architecture: feature augmentation and weighted methods. We combine the results of the CFM and Content-Based models by using a linear combination of their scores, Eq. (20) (weighted method). The list is then classified, and the first 100 recommended items are chosen. Finally, we take the selected results, and reclassify them from best to worst in terms of GHSOM-CF scores, with the first  $N$  items serving as recommendations.

$$Hybrid_{weighted} = U + V \quad (20)$$

### 3.3.5 Hadamard product matrix formation

Assume  $X$ ,  $Y$ , and  $Z$  are  $P \times Q$  matrices (Fig. 8). If this is the case, then  $Z \circ (X + Y) = Z \circ X + Y \circ Y$ . In addition, the Eqs. (21) and (22)  $\alpha(X \circ Y) = (\alpha X) \circ Y = X \circ (\alpha Y)$  holds true.

$$\mathbf{Proof 1} : [Z \circ (X + Y)]_{ab} = [Z \circ X + Y \circ Y]_{ab} \quad (21)$$

$$\mathbf{Proof 2} : [Z \circ X + Z \circ Y]_{ab} = [\alpha X \circ Y]_{ab} \quad (22)$$

**Definition** The Hadamard transform HM is a  $2^p \times 2^p$  matrix called the Hadamard matrix that changes  $2^p$  real numbers  $P_n$  into  $2^p$  real numbers  $X_k$ . This matrix is scaled by a normalisation factor. Recursion and the binary (BASE-2) representation of 'm' and 'n' both work for defining the Hadamard transform.

## 4 Result and discussion

The experiments applied to assess the effectiveness of the proposed method to RS are presented in this section. In order to validate the concept, we utilized two well-known datasets: Amazon Fine Food ratings and Amazon Movie reviews.

### 4.1 Dataset

The datasets were selected based on their availability and accessibility. Furthermore, we considered how generally they are accepted by the scientific community. The following are the datasets:

- *Amazon fine foods ratings are Amazon's ratings of fine foods*. Each review comprises information about the product and the consumer, as well as a rating and the plaintext evaluation submitted by every user to each thing they evaluated. Up until October 2012, the data covered a period of much more than ten years, with 568,454 reviews from 256,059 consumers and 74,238 products.
- *Amazon movie ratings is a collection of Amazon movie ratings*. Each review also offers information about the product and its users, as well as ratings and textual reviews. It also spans a 10-year period, with 7,911,684 reviews from 889,176 individuals and 253,059 goods up until October 2012.

Table 1 lists the statistics of both datasets.

### 4.2 Metrics

To monitor our models' efficiency, the leave-one-out technique is employed. According to this methodology or technique, we were leaving out the interaction of the final user-item used for testing for each user. Training is done with the remaining user-item interactions. Additional implicit feedback RS commonly use this mechanism. Now we must assign a ranking to each item for each user. A popular technique [46], which randomly chooses the top-K items with which the user has not engaged and then ranks the leave-out item between these top-K item lists, is utilised. To evaluate the performance, we employed the Hit Ratio (HR) and the evaluation metrics of Normalized Discounted Cumulative Gain (NDCG) [47]. The Hit Ratio determines whether the remaining appears in the top-K ranked list, whereas the NDCG determines the position of the left-out product in the ranking by giving top scores if it appears in the top-K ranked list. Our trials were conducted on a top-10 product ranking for users [48].

### 4.3 Competing methods

We contrasted recommendation models with *state-of-the-art* approaches. Because our NN models evaluate user sentiment, we pitted them against SA-based RS proposed by (i) Dang et al. [49], (ii) Osman et al. [38], and (iii) Wang et al. [50]. On

both datasets, we measured the effectiveness of alternative models in Figs. 9 and 10. The purpose of RS on E-Comm sites is to make product suggestions to the users. Customers can be given product recommendations based on the site's best-selling products, the customer's demographic characteristics, or an analysis of their purchase history to accurately predict their future preferences. The performance metrics Hit Ratio, as well as Normalized Discounted Cumulative Gain, are used on the Y-axes, while the magnitude of predictive factors is used on the X-axes in these plots. Among every challenging technique, Figs. 11 and 12 demonstrate that the proposed hybrid model is the winner among the other advanced CF models that are outperformed by a significant margin. A wide margin outperforms the advanced one on both datasets. The Osman et al.'s model in association with 48 predictive factors is outperformed by the proposed with at least a least amount of significant predictors like 8, 16 on the Amazon Movie Review dataset. This illustration demonstrates the model's expressiveness that was created by combining HRS integrated with HRS. Other neural models, such as Wang et al. and Selmen et al. [41], also perform admirably. When compared to Wang et al.'s work, Selmen et al. slightly perform in a worse manner. Wang et al. outperformed Osman et al. and Dang et al. significantly.

The performance appraisal of Top-K item suggestions as depicted in Figs. 11 and 12, where the ranking position varies from 1 to 10, demonstrate the importance of including SA, we evaluate the proposed model to various non-sentimental based RS proposed by (i) Fu et al. [52], (ii) Son et al. [53], and (iii) Strub et al. [54]. We could see that the suggested model improves over time when compared to other techniques. Son et al.'s work performed better than the Fu et al. model. Finally, we could see that Strub et al.'s model had the weakest performance, emphasizing the value of customized RS.

## 5 Conclusion and future works

The accuracy and timeliness of an E-Comm RS are essential. To improve the accuracy of RS, this paper proposes a product RS based on hybrid recommendation and SA. The system architecture presented in this paper can integrate a variety of techniques such as CFM, content-based filtering, and CF with a GHSOM model that takes demographic attributes into account, as well as hybrid DL models with Bert-MCA-RU-GP-SoftMax for SA. Experiments with food and movie reviews were carried out. We show the functionality and suitability of our methods in creating personalized recommendations in E-Comm applications based on these experiments. The findings show that combining DL-based SA-CFM improves performance significantly. This is accomplished by utilizing additional information gleaned from user reviews. Its incorporation into conventional RS improves the reliability and ability to provide improved recommendations to users.

This paper shows that future work can be done to improve how MapReduce performs with cloud-based solutions during multithreaded. Optimizing MapReduce performance is a priority. Techniques for cloud vitalization must support elastic resource provisioning when user access varies dynamically. By posing specific

questions to buyers after they have made a purchase, a questionnaire can possibly learn good data for optimising a website and enhancing a RS.

**Funding** Not applicable.

**Availability of data and materials** Not applicable.

**Code availability** Not applicable.

## Declarations

**Conflict of interest** All authors declare that they have no conflict of interest.

## References


1. Aaron, O. "What is the future of Ecommerce? 10 insights on the evolution of an industry." *Shopify Plus*, 31 January 2019.
2. Shareef, S. M., & Hashim, S. H. (2020). Proposed hybrid classifier to improve network intrusion detection system using data mining techniques. *Engineering and Technology Journal*, 38, 6–14.
3. Zhang, H., Kong, X., & Zhang, Y. (2021). Selective knowledge transfer for cross-domain collaborative recommendation. *IEEE Access*, 9, 48039–48051.
4. Hanafizadeh, P., Firouzabadi, M. B., & Vu, K. M. (2021). Insight monetization intermediary platform using recommender systems. *Electronic Markets*, 31, 1–9.
5. Shafqat, W., & Byun, Y. C. (2020). A context aware location recommendation system for tourists using hierarchical LSTM model. *Sustainability*, 12, 4107.
6. Kang, L., Liu, S., Gong, D., & Tang, M. (2021). A personalized point-of interest recommendation system for O2O commerce. *Electronic Markets*, 31, 253–267.
7. Wang, Y.; Sharma, R. Design of front end for recommendation systems: Towards a hybrid architecture. In Proceedings of the 18th International Conference on Electronic Business, Guilin, China, 2–6 December 2018; pp. 211–220.
8. Li, G., Zhu, T., Hua, J., Yuan, T., Niu, Z., Li, T., & Zhang, H. (2019). Asking images: Hybrid recommendation system for tourist spots by hierarchical sampling statistics and multimodal visual bayesian personalized ranking. *IEEE Access*, 7, 126539–126560.
9. Fayyaz, Z., Ebrahimian, M., Nawara, D., Ibrahim, A., & Kashef, R. (2020). Recommendation systems: algorithms, challenges, metrics, and business opportunities. *Applied Sciences*, 10, 7748.
10. Zhuang, Y., & Kim, J. (2021). A BERT-based multi-criteria recommender system for hotel promotion management. *Sustainability*, 13, 8039.
11. Bangaru Samatchi, R. S. (2019). Improvement of crop production using recommender system by weather forecasts. *Procedia Computer Science*, 165, 724–732.
12. Jaiswal, B., Kharade, T., Kotambe, N., & Shinde, S. (2020). Collaborative recommendation system for agriculture sector. *ITM Web of Conferences*, 32, 03034.
13. Dang, N. C., Moreno-García, M. N., & De la Prieta, F. (2020). Sentiment analysis based on deep learning: A comparative study. *Electronics*, 9, 483.
14. Preethi, G.; Krishna, P.V.; Obaidat, M.S.; Saritha, V.; Yenduri, S. Application of Deep Learning to Sentiment Analysis for Recommender System on Cloud. In Proceedings of the 2017 International Conference on Computer, Information and Telecommunication Systems (CITS), Dalian, China, 21–23 July 2017; IEEE: New York, NY, USA, 2017; pp. 93–97.
15. Keenan, M. J. S. (2018). *Advanced positioning, flow, and sentiment analysis in commodity markets: Bridging fundamental and technical analysis* (2nd ed.). Wiley.
16. Sánchez-Moreno, D.; Moreno-García, M.N.; Sonboli, N.; Burke, R. Using Social Tag Embedding in a Collaborative Filtering Approach for Recommender Systems. In Proceedings of the 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, Melbourne, Australia, 14–17 December 2020; IEEE: New York, NY, USA, 2021.
17. Aggarwal, B., Mittal, N., Bansal, P., Garg, S. (2015). Sentiment analysis using common-sense and context information. *Computational Intelligence and Neuroscience* 2015(30).

18. García-Cumbreras, M. Á., Montejó-Ráez, A., & Díaz-Galiano, M. C. (2013). Pessimists and optimists: Improving collaborative filtering through sentiment analysis. *Expert Systems with Applications*, 40(17), 6758–6765.
19. Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender systems: Introduction and challenges. *Recommender systems handbook* (pp. 1–34). Springer.
20. Guimaraes R., Rodriguez D. Z., Gerais M., Rosa R. L., Bressan G., Paulo S., Recommendation system using sentiment analysis considering the polarity of the adverb. In Consumer Electronics (ISCE), IEEE International Symposium on IEEE, 2016, pp. 71–72.
21. Lei, X., Qian, X., & Zhao, G. (2016). Rating prediction based on social sentiment from textual reviews. *IEEE Transactions on Multimedia*, 18(9), 1910–1921.
22. Peleja, F., Dias, P., Martins, F., & Magalhães, J. (2013). A recommender system for the TV on the web: Integrating unrated reviews and movie ratings. *Multimedia Systems*, 19(6), 543–558.
23. Alhamid, M. F., Rawashdeh, M., Al, O. H., Hossain, M. S., & El Saddik, A. (2015). Towards context-sensitive collaborative media recommender system. *Multimedia Tools and Applications*, 74(24), 11399–11428.
24. Hu, Y.; Ogihara, M. Nextone Player: A Music Recommendation System Based on User Behavior. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), Miami, USA, 24–28 October 2011; pp. 103–108.
25. Salehi, M. (2013). An effective recommendation based on user behaviour: A hybrid of sequential pattern of user and attributes of product. *International Journal of Business Information Systems*, 14, 480–496.
26. Lin, D., Su, J. A Recommender System Based on Contextual Information of Click and Purchase Data to Items for E-Commerce. In Proceedings of the 3rd International Conference on Cyberspace Technology (Cct 2015), IEEE Xplore, Beijing, China, 17–18 October 2015; pp. 1–6.
27. Wang, B., Ye, F., & Xu, J. (2018). A personalized recommendation algorithm based on the user's implicit feedback in E-Commerce. *Future Internet*, 10, 117.
28. Wang, Y., Wang, M., & Xu, W. (2018). A sentiment-enhanced hybrid recommender system for movie recommendation: A big data analytics framework. *Wireless Communications and Mobile Computing*, 2018, 1–9.
29. Kumar, S., De, K., & Roy, P. P. (2020). Movie recommendation system using sentiment analysis from microblogging data. *IEEE Transactions on Computational Social Systems*, 7, 915–923.
30. Ziani, A., Azizi, N., Schwab, D., Aldwaini, M., Chekkai, N., Zenakhra, D., Cheriguene, S. Recommender system through sentiment analysis. In Proceedings of the 2nd International Conference on Automatic Control, Telecommunications and Signals, Annaba, Algeria, 11–12 December 2017
31. Nimirthi, P., Krishna, P., Sridat, M.S., Saritha, V. A framework for sentiment analysis based recommender system for agriculture using deep learning approach. In Social Network Forensics, Cyber Security, and Machine Learning; Springer, 2019; pp. 59–66.
32. Devipriya, K., Pragna, D., Pirya, V., & Sudhakar, S. (2020). Deep learning sentiment analysis for recommendations in social applications. *International Journal of Scientific & Technology Research*, 9, 3812–3815.
33. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies, (Vol. 1) (Long and Short Papers), Minneapolis, Minnesota. Association for Computational Linguistics, pp. 4171–4186
34. Su, X., & Khoshgoftaar, T. M. (2008). Collaborative filtering for multi-class data using Bayesian networks. *Collaborative filtering for multi-class data using Bayesian networks*, 17, 71–85. <https://doi.org/10.1142/S0218213008003789>
35. Pan R., Xu G. and Dolog P. (2012). Improving Recommendations. In: Tag-Based Systems with Spectral Clustering of Tag Neighbors. Park J.J., Chao H.C., Obaidat M.S. and Kim J., (Eds.) Springer Netherlands, ISBN-10: 978-94-007-2792- 2, pp. 355–364.
36. Ghabayen, A. S., & Mohd Noah, S. A. (2014). Exploiting social tags to overcome cold start recommendation problem. *Journal of Computer Science*, 10(7), 1166–1173.
37. Melville, P., & Sindhvani, V. (2011). Recommender systems. *Encyclopedia of machine learning* (pp. 829–838). Boston: Springer.
38. Osman, N. A., Mohd Noah, S. A., Darwich, M., & Mohd, M. (2021). Integrating contextual sentiment analysis in collaborative recommender systems. *PLoS ONE*, 16(3), e0248695. <https://doi.org/10.1371/journal.pone.0248695>

39. Ghabayen, A. S., & Noah, S. M. (2017). Using tags for measuring the semantic similarity of users to enhance collaborative filtering recommender systems. *International Journal on Advanced Science, Engineering and Information Technology*, 7(6), 2063–2070.
40. Hu, F., & Zhou, Z. (2022). Information services and omnichannel retailing strategy choices of E-Commerce platforms with supplier competition. *Electronic Commerce Research*. <https://doi.org/10.1007/s10660-022-09557-5>
41. Zhou, L. (2020). Product advertising recommendation in E-Commerce based on deep learning and distributed expression. *Electronic Commerce Research*, 20, 321–342. <https://doi.org/10.1007/s10660-020-09411-6>
42. Wu, X. Q., Zhang, L., Tian, S. L., & Wu, L. (2021). Scenario based E-Commerce recommendation algorithm based on customer interest in Internet of things environment. *Electronic Commerce Research*, 21(3), 689–705. <https://doi.org/10.1007/s10660-019-09339-6>
43. Kundu, S., & Chakraborti, S. (2022). A comparative study of online consumer reviews of Apple iPhone across Amazon, Twitter and MouthShut platforms. *Electronic Commerce Research*, 22, 925–950. <https://doi.org/10.1007/s10660-020-09429-w>
44. Kashyap, R., Kesharwani, A., & Ponnampalani, A. (2022). Measurement of online review helpfulness: A formative measure development and validation. *Electronic Commerce Research*. <https://doi.org/10.1007/s10660-022-09531-1>
45. He, P., Wang, T. Y., Shang, Q., et al. (2022). Knowledge mapping of E-Commerce supply chain management: A bibliometric analysis. *Electronic Commerce Research*. <https://doi.org/10.1007/s10660-022-09609-w>
46. Chan, K. H., Ke, W., & Im, S. K. (2020, November). CARU: A content-adaptive recurrent unit for the transition of hidden state in NLP. In *International Conference on Neural Information Processing* (pp. 693–703). Springer.
47. Cheng, T., Koc, L., Harmsen, J., & Shaked, T. (2016). Wide and deep learning for recommender systems. In *WWW*, pp. 2787–2795.
48. He, C. T., Kan, X., M., & Chen, X. (2001). TriRank: Review-aware explainable recommendation by modeling aspects. In *CIKM*, pp. 285–295.
49. Dang, C., García, M. M., & De La Prieta, F. (2021). An approach to integrating sentiment analysis into recommender systems. *Sensors*, 21, 5666. <https://doi.org/10.3390/s21165666>
50. Wang, Y., Mingming, W., & Xu, W. (2018). A sentiment-enhanced hybrid recommender system for movie recommendation: a big data analytics framework. *Wireless Communications and Mobile Computing*, 2018, 1–9. <https://doi.org/10.1155/2018/8263704>
51. Selmene, S., & Kodiah, Z. (2020, June). Recommender System Based on User's Tweets Sentiment Analysis. In *2020 The 14th International Conference on E-Commerce, E-Business and E-Government* (pp. 96–102). <https://doi.org/10.1145/3409929.3414744>
52. Fu, M., Qu, H., Qi, J., Lu, L., & Liu, Y. (2019). A novel deep learning-based collaborative filtering model for recommendation system. *IEEE Transactions on Cybernetics*, 49(3), 1084–1096. <https://doi.org/10.1109/TCYB.2018.2795041>
53. Son, J., & Kim, S. B. (2017). Content-based filtering for recommendation systems using multiattribute networks. *Expert Systems with Applications*, 89, 404–412.
54. Strub, F., Caudel, R., Mary, J. Hybrid recommender system based on Autoencoders. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, Boston, 15 Sept 2016; pp. 15–16.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## Authors and Affiliations

**Arodh Lal Karn<sup>1</sup> · Rakshha Kumari Karna<sup>2</sup> · Bhavana Raj Kondamudi<sup>3</sup> ·  
Girish Bagale<sup>4</sup> · Denis A. Pustokhin<sup>5</sup> · Irina V. Pustokhina<sup>6</sup> · Sudhakar Sengan<sup>7</sup> **

Arodh Lal Karn  
alkmaithili@foxmail.com

Rakshha Kumari Karna  
rakshha.karna@gmail.com

Bhavana Raj Kondamudi  
bhavana\_raj\_83@yahoo.com

Girish Bagale  
girishbagale08@gmail.com

Denis A. Pustokhin  
dpustokhin@yandex.ru

Irina V. Pustokhina  
ivpustokhina@yandex.ru

- <sup>1</sup> Department of Financial and Actuarial Mathematics, School of Mathematics and Physics, Xi'an Jiaotong-Liverpool University, Suzhou 215123, Jiangsu, China
- <sup>2</sup> School of Management, Harbin Institute of Technology, Harbin 150001, China
- <sup>3</sup> Department of Management Studies, Institute of Public Enterprise, Hyderabad 500 101, India
- <sup>4</sup> SBM's Pravin Dalal School of Entrepreneurship and Family Business Management, SVKM's NMIMS University, Mumbai 400 056, India
- <sup>5</sup> Department of Logistics, State University of Management, Moscow, Russia 109542
- <sup>6</sup> Department of Entrepreneurship and Logistics, Plekhanov Russian University of Economics, Moscow, Russia 117997
- <sup>7</sup> Department of Computer Science and Engineering, PSN College of Engineering and Technology, Tirunelveli, Tamil Nadu 627152, India