Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 5, pp. 3408–3426, DOI 10.12694/scpe.v25i5.3007

IOT-DRIVEN HYBRID DEEP COLLABORATIVE TRANSFORMER WITH FEDERATED LEARNING FOR PERSONALIZED E-COMMERCE RECOMMENDATIONS: AN OPTIMIZED APPROACH

ABDULMAJEED ALQHATANI [*]AND SURBHI BHATIA KHAN [†]

Abstract. Recommender systems are already being used by several biggest e-commerce websites to assist users in finding things to buy. A recommender system gains knowledge from a consumer and suggests goods from the available goods that will find most value. In this deep learning technique, the Hybrid Deep Collaborative Transformer (HDCT) method has emerged as a promising approach. However, it is crucial to thoroughly examine and rectify any potential errors or limitations in the optimization process to ensure the optimal performance of the HDCT model. This study aims to address this concern by thoroughly evaluating the HDCT method uncovering any underlying errors or shortcomings. By comparing its performance against other existing models, the proposed HDCT with Federated Learning method demonstrates superior recommendation accuracy and effectiveness. Through a comprehensive analysis, this research identifies and rectifies the errors in the HDCT model, thereby enhancing its overall performance. The findings of this study provide valuable insights for researchers and practitioners in the field of e-commerce recommendation systems. Data for the RS is collected from the Myntra fashion product dataset. By understanding and addressing the limitations of the HDCT method, businesses can leverage its advantages to improve customer satisfaction and boost their revenue. Ultimately, this research contributes to the ongoing advancements in e-commerce recommendation systems and paves the way for future improvements in this rapidly evolving domain. The suggested model's efficacy is assessed using metrics for MSE, MSRE, NMSE, RMSE, and MAPE. The suggested values in metrics are 0.2971, 0.2763, 0.4013, 0.3222, 0.2911 at a 70% learn rate and 0.2403, 0.2234, 0.3506, 0.2025, 0.2597 at an 80% learn rate, and the proposed model outperformed with the least amount of error.

Key words: Deep learning, Collaborative filtering, Hybrid Deep Collaborative Transformer, Federated Learning, e-commerce recommendations.

1. Introduction. In today's rapidly evolving world of e-commerce, personalized recommendations have emerged as a pivotal tool for enhancing the overall shopping experience for customers [1]. Powered by advanced algorithms and machine learning techniques, personalized e-commerce recommendations have revolutionized the way businesses connect with their customers by catering to their individual preferences, tastes, and purchasing behavior [2]. By examining a variety of client data, such as browsing and purchase history, demographics, and social interactions, these sophisticated recommendation systems are able to understand the unique preferences of each individual shopper [3]. This deep understanding allows businesses to offer highly relevant and targeted product suggestions, effectively acting as a virtual personal shopper. The benefits of personalized recommendations are twofold [4]. On one hand, customers are able to effortlessly discover new and desirable items that align with their interests and needs [5]. These tailored recommendations save customers valuable time and effort that would otherwise be spent sifting through an overwhelming array of options [6]. Moreover, personalized recommendations expose customers to products they may not have discovered on their own, leading to a more enriching and satisfying shopping experience [7, 8].

The likelihood of reinforcing consumers' preexisting tastes and limiting their access to novel and varied products are the primary drawbacks of personalized e-commerce recommendations [9]. Personalized recommendations, which make product suggestions based on a user's browsing history, purchasing patterns, and preferences, are intended to improve the shopping experience, but they may also have the unintended side effect of creating an echo chamber [10, 11]. Users may miss out on learning about new and alternative possibil-

[*] Department of Information Systems, College of Computer Science and Information Systems, Najran University, Najran 61441, Saudi Arabia (aaalqhatni@nu.edu.sa).

[†] School of science engineering and environment, University of Salford, Manchester, United Kingdom (s.khan138@salford.ac.uk); Department of Electrical and Computer Engineering, Lebanese American University, Byblos, Lebanon

ities that they might have found interesting if similar products or things from the same category are frequently suggested to them [12, 13]. Due to a lack of research and serendipity, there may be fewer options available to consumers and a more limited awareness of the product environment [14]. Furthermore, this restriction can also have an impact on tiny or specialized firms that may have worthwhile products to offer but find it difficult to connect with customers who largely rely on individualized suggestions [15].

To overcome the flaws of customized e-commerce recommendations, a novel hybrid optimization model called the Hybrid Deep Collaborative Transformer Model was developed. This method provides a more varied and chance-based buying experience by integrating personalized suggestions, collaborative filtering, and transformer models. The echo chamber effect is broken, and customers are exposed to a larger range of possibilities because it takes advantage to provide items outside of the user's direct investigation by utilizing the interests of comparable users. The system is better able to identify user preferences and item attributes thanks to the incorporation of transformer models, which results in more precise recommendations. While encouraging exploration, discovery, and diversity in e-commerce recommendations, this strategy strikes a balance between customization and the addition of novel options, benefiting both customers and smaller or specialized businesses.

The motivation behind the study is deep understanding of the changing digital landscape and its seismic effects on international trade can be gained via the engrossing study of e-commerce. Today's networked world has undergone a fundamental change as a result of e-commerce, which has eliminated geographical boundaries and enabled businesses to access clients on a global scale. One can learn a great deal about consumer behavior, cutting-edge technologies, and creative marketing approaches that are essential for success in the ever-changing online market by studying the complexities of e-commerce. Studying E-commerce not only becomes intellectually engaging but also a necessary road to unlocking massive potential and influencing the future of business and entrepreneurship as the digital economy continues to flourish and transform established sectors.

Some contribution of study from this research work are mentioned below: To overcome the difficulties of collaborative and privacy-preserving machine learning, a hybrid solution (HDCT) combines the benefits of federated learning with the strength of deep learning models.

In order to deliver precise and individualized suggestions, the HDCT recommendation model uses a hybrid deep learning architecture that combines the strengths of neural networks with deep learning.

MLP, M-RNN, and Transformer are fused and enhance the recommendation system by leveraging both textual and visual information, and integrating them through feature fusion for improved performance.

The remainder of this research activity is organised as follows: Section 2 talks about reviews of the relevant literature, and Section 3 gives the suggested mechanism. The experimental findings are presented in Section 4. This study is concluded in Section 5.

2. Literature Review.

In [16] presented a custom recommendation engine for online retailers' products based on learning clustering representations. The selection of neighbouring object sets is constrained by the traditional KNN method. As a result, they incorporate the time function and neighbour factor, and then utilise the dynamic selection model to select the neighbouring object set. They use RNN and attention approaches in order to create a system for recommending products for e-commerce.

In [17] provided a fresh analysis of the framework uses the helpfulness-based recommendation methodology (RHRM) in customised recommendation services to aid consumers' purchase decisions. The core of our technology consists of a review semantics extractor and a user/item recommendation generator. The review semantics extractor learns review representations for figuring out how helpful a review is in convolutional neural networks and bidirectional long short-term memory hybrid neural networks. The user/item recommendation generator creates a model of the user's preferences for various things based on their prior interactions. Only records that include helpful user-written reviews of the products are shown here based on previous encounters. Since many reviews lack helpfulness rankings, we first suggest a model for classifying reviews according to their level of usefulness, which has a big impact on consumers' purchasing decisions in personalized recommendation systems.

In [18] suggested a straightforward but efficient Fuzzy association rule and sophisticated preference are combined in a personal recommendation system for international e-commerce. Using fuzzy association rules, it is possible to prevent the creation of a hybrid recommendations model based on intricate user preference characteristics while still allowing for the customised recommendation of products based on user behaviour preferences. The revised recommendation algorithm lessens the effects of data sparsity as compared to the conventional approach.

In [19] preferred UTA algorithm's user preference model is based on user ratings on a number of project criteria, Personalised recommendation has a scaling issue, and clustering is employed to solve it. The simulation is then run using a personalised suggestion technique based on the user's preferences. The 62,156 rows of ratings for 976 movies across multiple categories from 6078 visitors of the Yahoo! Movies website make up the simulation data.

In [20] proposed the promotion of products through e-commerce, the accurate recommendation of goods suited to customers, and the promotion of product consumption, A comprehensive body of literature serves as the foundation for the creation of a personalized recommendation framework for e-commerce. a cloud computing platform that makes use of Hadoop. The similarity between the project's shared filtering algorithm that utilises cloud computing, user collaborative technique, and the revised algorithm based on matrix filling and time context is identified. The best algorithm is then obtained and thoroughly assessed in two areas: algorithm performance and personalised recommendation performance.

In [21] discussed the JD.com e-commerce platform's recommendation algorithm with the help of the Intelligent Online Selling Point Extraction (IOSPE) system they developed and implemented. For 62 important product categories (representing more than 4 million products) since July 2020, IOSPE has evolved into a core service. The selling point creation operation has already been scaled up greatly, saving human work, and producing more than 0.1 billion selling points.

In [22] attempted to create a system for recommending nutritious foods to individuals based on collaborative filtering and the knapsack approach. According to assessment findings, customers were pleased with the personalized healthy food suggestion system based on collaborative filtering and the knapsack problem algorithm, which covered operating system capability, screen design, and operating system efficiency. Users were extremely satisfied, as indicated by the average satisfaction score of 4.20 for the entire sample. Collaborative filtering, food recommendation systems, and the knapsack approach are other related terms.

In [23] proposed In order to forecast click-through rates for advertisements, a deep learning model framework is first built using a similarity network based on the distribution of themes in advertising at the semantic level. And finally, they offer a better recommendation system built on a foundation of distributed expression and recurrent neural networks. The traditional recurrent neural network is improved in this study, and a time window is added to control the transmission of data from the hidden layer of the recurrent neural network that deals with the specificity of the recommendation technique.

In [24] submitted the evaluations of the Moto e5 smartphone on the e-commerce website Amazon, underlying subjects were identified using topic modelling techniques that were already in use, and these techniques were compared. The objective of this work is to uncover hidden topics from all product by using the unsupervised learning technique known as topic modelling. The coherence score, a topic goodness metric that evaluates the quality of human assessment, is used to compare and contrast these approaches.

In [25] improved e-commerce's Service (QoS) and experience (QoE) quality. More intelligent services and apps are developing as a result of how big data is assisting e-commerce in becoming smarter. Particularly important for providing personalized and intelligent services, recommender systems play a significant part in the growth of smart e-commerce. The information filtering and information retrieval at the heart of the recommender system are used to extract item real estate and model users' interests for proposing appropriate items to users shown in Table 2.1.

3. Proposed methodology. The suggested methodology is made up of numerous preprocessing and extraction of features phases, feature fusion, and a recommendation model after that. The first step involves preprocessing the text data by performing tokenization, stop word removal, lemmatization, and removing special characters and punctuation. Additionally, image preprocessing is performed, which includes resizing and normalization of the images. In the second step, features are extracted from different sources. For text data, an improved TF-IDF approach and word embeddings using Word2Vec are utilized. Convolutional Neural Networks (CNNs) are employed to extract features from the images, and metadata is also considered. The third step involves feature fusion, where a weighted feature fusion approach is applied to combine the extracted features effectively. Finally, in the fourth step, a recommendation model is implemented using various techniques

Table 2.1: Problem identification

Author and citation	year	Aim	Methodology	Problem Methodology
[16]	2019	Design a learning clustering representation -based personalized recommendation system for online retailers' products	Combine RNN and attention methods, add Neighbor factorand time function, use dynamic selection model to choose neighboring object set	Constrained selection of nearby object sets in the traditional KNN approach
[17]	2021	To provide a fresh analysis of the framework of the helpfulness -based recommendation methodology (RHRM) to assist consumers' purchase choices in customized recommendation services.	cc A convolutional neural network and a bidirectional long short-term memory hybrid neural network to learn	Personalized recommendation systems, consumers' purchasing decisions can be influ -enced significantly by helpful user-written
[18]	2020	Develop a cross-border e-commerce personalized recommendation algorithm combining preference and fuzzy association rule	Personalize recommendations based on user behavior preferences	Reduce the impact of data sparsity in cross-border e-commerce personalized recommendations
[19]	2019	Utilize clustering and UTA algorithm to address scalability issue and provide personalized recommendation	Perform simulations and suggest personalized recommendations based on user preference	Address the scalability issue of personalized recommendation, build a user preference model, and provide personalized suggestions
[20]	2021	Construct a personalized recommendation framework for e-commerce based on a large body of literature and comparison of algorithms	Cloud computing and Hadoop, User collaborative algorithm, and Improved algorithm based on matrix filling	Construct a personalized recommendation framework for e-commerce, compare different algorithms and assess their performance
[21]	2022	cc Discuss the Intelligent Online Selling Point Extraction (IOSPE) system and its implementation in an e-commerce platform	Scale up selling point creation operation, save human work, and generate a large number of selling points	Develop and deploy IOSPE system to support e-commerce recommendations, improve efficiency, and generate a large number of selling points
[22]	2021	To create a system for recommending nutritious foods to individuals	Collaborative filtering and the knapsack approach	Personalized healthy food suggestions
[23]	2020	cc Construct a deep learning model framework for advertising click-through rate prediction	Utilize similarity networks and text recurrent neural networks to improve recommendation algorithm	Enhance conventional recurrent neural networks to address the specificity of the recommendation algorithm
[24]	2020	Uncover hidden topics from product reviews using topic modelling techniques	Compare and evaluate different topic modelling techniques	Use unsupervised learning (topic modelling) to identify hidden topics in product reviews and assess technique quality
[25]	2021	Improve Quality of Service (QoS) and Quality of Experience (QoE) in e-commerce	Utilize recommender systems for personalized and intelligent services	Enhance e-commerce by providing smarter and more intelligent services using recommender systems

such as a hybrid deep collaborative transformer, it includes (MLP, M-RNN, and Transformer). By utilising both textual and visual information and fusing them through feature fusion for greater performance, this holistic strategy seeks to improve the recommendation system showun in Fig. 3.1.

3.1. Preprocessing. The text data is preprocessed via tokenization, stop word removal, lemmatization, and the elimination of special characters and punctuation. Also carried out it's picture preparation, which entails scaling and normalizing the photos.

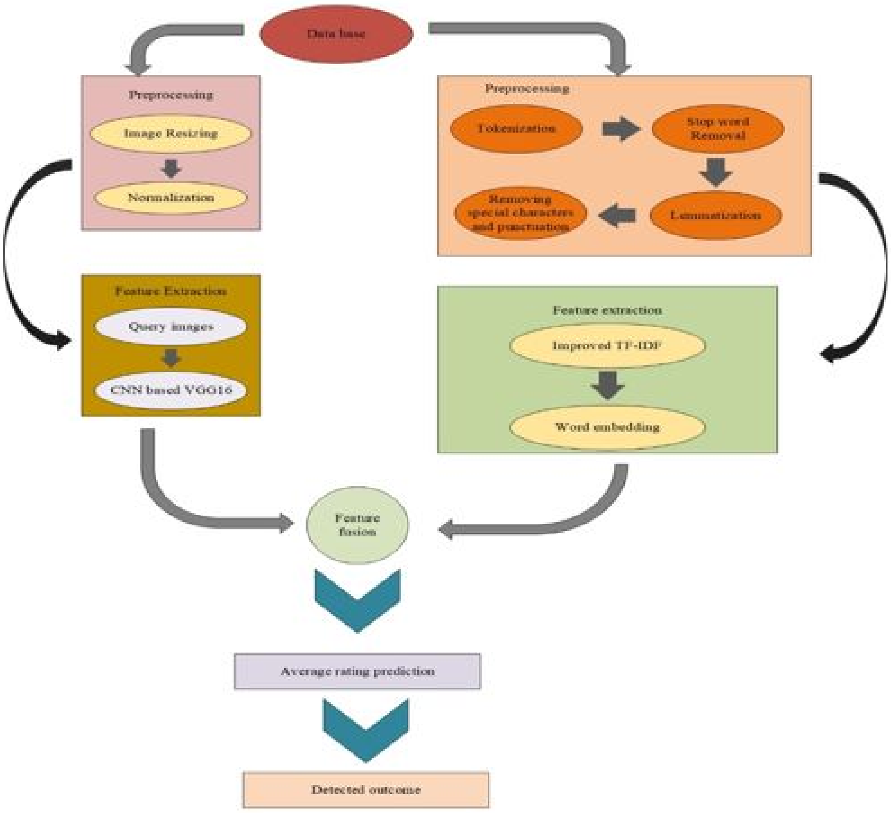

Fig. 3.1: Overall architecture diagram

3.1.1. Tokenization. To distinguish the contents, tokens, also known as words, are utilized. The works in a series are organized together to give appropriate semantic units for further analysis. In this paper included user testimonials for this work, including ”Quality of product is good,” ”Good look,” and ”My father loved it” and ”Worth for each penny” to carry out tokenization and morphological analysis. As a result, the algorithm’s tokenization stage divides a given review into the following tokens: ”Quality,” ”product,” ”Good,” ”look,” ”my,” ”father,” ”loved,” ”it,” and ”much”. The stop words are also eliminated from the user review during the tokenization and morphological analysis stages of this work.

3.1.2. Stop word removal. Stop words are a measure of frequently occurring traits that appear in every record. Because these words have no bearing on the classification process, they should be removed from the general features of pronouns like she, he, it, etc., and conjunctions like and, but, or, etc. To reduce the quantity of the data, the stop word should be eliminated. To increase performance, a stop word needs to be eliminated. Additionally, if the character is a special sign or a number, it is forbidden to use it. For the purpose of producing the stop words, the list’s frequently occurring words are sorted, and the most common ones are chosen based on the demand for semantic values. Once such terms have been chosen, they need to be removed. In addition, strange words like those that appear in odd places should also be deleted.

3.1.3. Lemmatization. Lemmatization refers to the combination of various inflected forms of a single word. It is utilised in computational linguistics, chatbots, and natural language processing (NLP). Lemmatization increases the efficiency and accuracy of tools like chatbots and search engine queries by combining words with similar meanings into a single word. Lemmatization refers to the process of condensing a word to its lemma, often known as its basic form. For instance, the verb ”running” would be known by the term ”run.”

Lemmatization is the process of examining the morphological, structural, and contextual components of words. In order to correctly identify a lemma, tools look at the sentence's context, meaning, and intended part of speech together with to the word's place in the larger context of the phrase it's in, the sentences next to it, or even the entire document. Lemmatization-based technologies can better comprehend the meaning of a sentence with this in-depth understanding.

A word is reduced to its lemma through lemmatization. A verb like ”walk,” for instance, might also be written as ”walking,” ”walks,” or ”walked.” The letters ”s,” ”ed,” and ”ing” that indicate inflection are eliminated. These words are grouped as a lemma through lemmatization, meaning ”walk”. Depending on the context, ”saw” could be understood in a variety of ways. For instance, the word ”saw” can be decomposed into the lemma ”see” or ”saw.” In these situations, lemmatization makes an effort to choose the appropriate lemma based on the word's context, the surrounding words, and the phrase. Other words, like ”better,” could be reduced to a lemma like ”good.”

3.1.4. Removing Special Character and Punctuation. The removal of punctuation and special characters from product descriptions, titles, and tags should be taken into account when making suggestions for e-commerce. Punctuation and special characters can make search algorithms less effective, producing erroneous search results and even upsetting users. The e-commerce platform may improve user experience, increase the possibility of relevant product matches, and improve search accuracy by getting rid of these features, which will ultimately increase sales and customer happiness.

3.1.5. Image preprocessing. Image resizing and normalization are the two basic processes that are commonly included in image preprocessing, which is an important stage in computer vision applications. Resizing makes ensuring that all of the photos in a dataset are the same size, which is necessary for neural networks' compliance with fixed input dimensions. Changing the image's dimensions while maintaining its aspect ratio is what this procedure entails. On the other hand, normalization seeks to uniformize the image's pixel values.

1. **Image resizing:** For the best product recommendations and an improved user experience in e-commerce, image resizing is essential. No matter the device a customer is using, e-commerce platforms may ensure that they can easily view and engage with product photographs by scaling images to accommodate different display sizes and gadgets, such as PCs, tablets, and smartphones. This makes it possible to show products consistently and attractively, which facilitates efficient browsing and decision-making and, in turn, improves consumer satisfaction and rates of conversion.
2. **Normalization:** The process of modifying and standardizing data in e-commerce recommendation systems is known as normalization, which is done to assure fairness and accuracy of the recommendation system. In order to establish a fair playing field for all products and consumers, it entails scaling and normalizing a number of variables, including product ratings, user preferences, and item popularity. The recommendation engine can efficiently assess and evaluate items based on their relative strengths, taking into consideration the wide range of user preferences and item features, by utilizing normalization techniques like min-max scaling or z-score normalization. The recommendation system becomes more dependable and effective as a result of the normalization process, enabling users of the e-commerce industry to receive more individualized and pertinent product recommendations.

3.2. Feature Extraction. Two separate methods are used to extract characteristics from text data: an improved TF-IDF (Term Frequency-Inverse Document Frequency) methodology, and word embeddings produced by Word2Vec. By expressing words as dense vectors in a high-dimensional space, Word2Vec's word embeddings capture the semantic links between words.

3.2.1. Improved Term Frequency-Inverse Document Frequency. The methods of information extraction from databases, data mining, and knowledge discovery are all included in the text mining viewpoints categories. Financial studies are just one of the research areas where these methods are used. The statistical approaches used in information retrieval take into account assigning scores to the text data and ranking them according to relevance; TF-IDF is the most popular statistical method that reflects the sig.

Step 1: Term Frequency (TF) Calculation:

(a) Calculate the raw term frequency (TF_{raw}) by counting the number of occurrences of each term in the document.

(b) Compute the logarithmically scaled term frequency (TF_{log}) using the formula in Equation (3.1):

$$TF_{\log(t,d)} = 1 + \log\left(TF_{raw(t,d)}\right) \tag{3.1}$$

Step 2: Inverse Document Frequency (IDF) Calculation:
(a) Calculate the document frequency ($DF(t)$) by counting the number of documents containing each term.
(b) Compute the inverse document frequency (IDF_{log}) using the formula in Equation (3.2),

$$IDF_{\log(t)} = \log(N/DF(t)) \tag{3.2}$$

Step 3: Term Frequency Normalization (proposed):
(a) Equation (3.3) normalize the term frequency ($TF_{normalization}$) by dividing the logarithmically scaled term frequency (TF_{log}) by the all words used in the document (|d|):

$$TF_{norm}(t,d) = TF_{\log(t,d)/|d|} \tag{3.3}$$

Step 4: Weight Function (proposed):
(a) Define a frequency-based weight that assigns weights to terms based on specific criteria. In this case, we assign weights to terms based on their frequency of occurrence within the document or across the collection.
(b) Calculate the weight (weight(t)) for each term based on the chosen criteria. One common approach is to use a term frequency-based weight. Here's the Equation (3.4) for assigning weights based on term frequency:

$$weight(t) = TF_{norm(t,d)} \tag{3.4}$$

In this Equation (3.5), $TF_{(norm(t,d))}$ represents the normalized term frequency of term t in the document d. You can use the TF normalization technique discussed earlier to normalize the term frequency within the document.

$$TF_{\text{norm }(t,d)} = TF_{\log}(t,d)/\max(\log(t,d)) \tag{3.5}$$

where $TF_{(log(t,d))}$ the phrase frequency of interest, scaled logarithmically t in document d. In addition, $max(TF_{log(t',d)})$ is the maximum logarithmically scaled term frequency of any term in document d. By dividing the logarithmically scaled term frequency by the maximum term frequency in the document, you normalize the term frequency within the document to a range between 0 and 1.
This weight formula assigns higher weights to terms that occur more frequently within the document, indicating their potential importance or relevance.

Step 5: TF-IDF Calculation:
(a) Calculate the modified TF-IDF in Equation (3.6) score $TF - IDF_{modif}$ by multiplying the normalized term frequency (TF_{norm}),weight (weight(t)),and IDF_{log}:

$$TF - IDF_{\text{modifiend }(t,d)} = TF_{\text{normalization }(t,d)} \times \text{ weight } (t) \times IDF_{\log(t)} \tag{3.6}$$

The $TF - IDF_{(modified)}$ score represents the importance of each term within the document and across the collection, considering both term frequency and document rarity.

3.2.2. Word2Vec. Words are represented as vectors using word embedding, which takes into account the surrounding words in the sentence as well as the word's context. Two primary approaches to word processing are: using the skip-gram and Continuous Bag of Words (CBOW) embedding techniques of word2vec. Through the use of context, CBOW is able to anticipate words. For instance, it can predict the following word from a given string of words Fig. 3.2 .

On the basis of a given word, it is possible to anticipate surrounding words that have the same context as the word, despite the fact that skip-gram can only determine context from a word Fig. 3.3

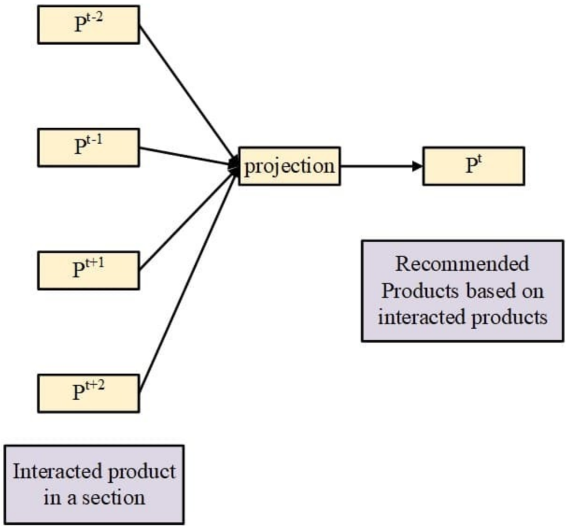
P^{t-2}
P^{t-1}
projection
P^{t}
P^{t+1}
P^{t+2}
Recommended Products based on interacted products
Interacted product in a section

Fig. 3.2: Using the CBOW model as an example to recommend products

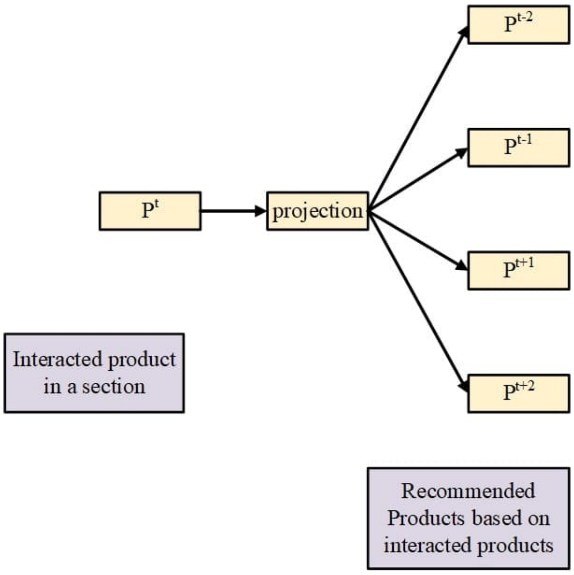
P^{t-2}
P^{t-1}
P^{t}
projection
P^{t+1}
Interacted product in a section
P^{t+2}
Recommended Products based on interacted products

Fig. 3.3: An example of a product recommendation N-skip-gram model

In order to analyses if a consumer Using the Word2Vec model, it is possible to determine, for example, whether a user is satisfied with a product after using it. Word2Vec approaches are frequently used for sentiment analysis. Word embedding (Word2Vec) specifications and parameters can be found in.

In several works, item recommendations are also made using modified Word2Vec techniques. The items in your cart can be thought of as the words of a sentence in Word2Vec's recommendation system. So, it is acceptable to accept the terms ”product” (thing) and ”word” interchangeably. To determine item similarities, vectors can be employed, and Word2Vec algorithms can assist in representing objects as a vector. Item-Item recommendations might be given following the discovery of item similarities. Instead of using word embedding in this study, Each session serves as a context (sentence) for the product embedding that we create from the data. A similar methodology is used in our Word2Vec RS. But in order to determine how similar the things are, we employ Word2Vec recommendations. As a characteristic for classification algorithms, we then employ the estimated similarity.

3.2.3. Extract features by images . Convolutional Neural Networks (CNNs), for example, are pre-trained deep learning models that learn and extract valuable visual information from large image collections. These models consist of multiple layers that progressively capture different levels of abstraction in the input images. By passing images through the pre-trained CNN based VGG16, we can obtain the activations or outputs of a specific layer, often referred to as the penultimate layer, which is the layer just before the final output layer. Extracting the output from this penultimate layer provides a high-level representation of the input images, capturing abstract and semantically rich features Following that, other activities, including image classification, can be accomplished using these derived features, object detection, or as input to other machine learning models for downstream tasks. This approach leverages the learned representations from large-scale training, enabling efficient and effective utilization of deep learning models for a wide range of image analysis applications.

3.2.4. Metadata. Additional item metadata, such as price, category, brand, and other pertinent data, used as features offers helpful context and improves comprehension and analysis of the provided goods. Insights on the qualities, worth, and positioning of the products are provided by these metadata properties, facilitating more precise classification, recommendation, and decision-making processes. Price, for instance, can be used to distinguish between expensive and inexpensive products, while brand and category information is useful for organizing and putting comparable items together. Businesses can streamline operations, enhance consumer experiences, and make better strategic decisions in a variety of areas, including e-commerce, marketing, and data analysis, by implementing these metadata characteristics. It is believed that the features obtained via Metadata are F_3 features.

3.3. Weighted feature fusion approach. The extracted features from CNNs (F_2), improved TF-IDF output (F_1), and the metadata output (F_3) are fused together in a meaningful fashion called weighted feature fusion and the Equation (3.7) are shown as below.

$$Weightfeaturefusion = W(F_1F_2F_3) \tag{3.7}$$

3.4. Recommendation Model. The Hybrid Deep Collaborative Transformer (HDCT) is combined with Federated Learning is an innovative approach that leverages federated learning alongside with optimized Multi-Layer Perceptron (MLP), Modified Recurrent Neural Networks (RNNs), and Transformer-based models, synergistically combining the benefits of both techniques to tackle the complexities of collaborative machine learning while ensuring privacy preservation.

1. Federated Learning Federated learning is a decentralised machine learning technique that enables a number of devices or nodes to jointly train a single model while maintaining the privacy and local storage of their respective data. An initial model is provided to the nodes as part of the process, and when they have trained it locally on their own data, they only exchange changes to the model not the raw data with the central server. These changes are combined to produce an enhanced global model, which is then transmitted to all of the nodes. Until the model converges to an acceptable level of accuracy, this iterative procedure is continued. Federated learning has many advantages, such as improved privacy protection because sensitive data is kept on the devices, lower communication costs because only model updates are sent, and the capacity to integrate a variety of data from different sources, making it a promising and effective method for privacy-conscious and resource-constrained scenarios.

   Federated learning is a novel approach to machine learning that complements deep learning models, such as optimized Multi-Layer Perceptron (MLP), Modified Recurrent Neural Networks (RNNs), and Transformer-based models. Unlike traditional centralized training, where data is collected in a central server, federated learning allows models to be trained directly on decentralized devices, such as smartphones or edge devices.
2. Multi-Layer Perceptron

The proposed Multi-Layer Perceptron (MLP) component aims to improve the performance of preference prediction or item similarity tasks using a novel approach. It starts with the Input Layer, which accepts a fused feature vector representing the input data. This fused feature vector is then passed through multiple Hidden Layers, each consisting of fully connected neurons with non-linear activation functions (such as sigmoid). The introduction of non-linear activation functions makes it possible for the network to discover intricate patterns and connections in the data. To optimize the activation function of the MLP, a new self-improved Bacterial Foraging Optimization (SI-BFO) algorithm is proposed. This optimization technique helps fine-tune the parameters of the activation functions, enabling the MLP to achieve better generalization and convergence properties during the training process. The SI-BFO method offers an efficient and effective way to find optimal configurations for the activation functions, further enhancing the performance of the MLP.

The MLP is utilized for this study's objectives. Three different layer types are characteristic of MLP, an instance of artificial neural network that is feed-forward (ANN):

- Input layer,
- Hidden layer and
- Output layer.

MLP can separate data that cannot be separated linearly if it is appropriately designed. This characteristic makes MLP useful for regressing and classifying a variety of problems. problems. Each artificial neuron (AN), the fundamental component of each MLP, is created using the activation function. The activation function (Y) determines the output of each AN are shown in Equation (3.8):

$$Y = f(u) \tag{3.8}$$

3. Logistic sigmoid Activation function

As follows is the formula for the logistic sigmoid activation functions are shown in Equation (3.9):

$$f(x) = \frac{1}{\left(1 + e^{-x)}\right.} \tag{3.9}$$

The mapping of all domain values into the co-domain interval by the logistic sigmoid distinguishes it from Tanh activation function, which is a significant difference denoted in Equation (3.10):

$$0 < f(x) < 1 \tag{3.10}$$

Positive output value will occur when a logical sigmoid is applied. Tanh activation function may not always be applied in this situation.

To optimize the activation function of MLP hear we use Self improved Bacterial Foraging Optimization.

3.4.1. Bacterial Foraging Optimization. Accurately mimics the key processes that E. coli uses to produce while it hunts for food, namely chemotaxis, reproduction, elimination and dispersal. Bacterial Foraging Optimization flow chart is given below Fig. 3.4.

Step 1: Using a random number generator, random coordinates can be assigned within a predetermined range to initialise the population of bacteria with random positions Equation (3.11).

$$\Theta_i(j, k, l) \tag{3.11}$$

Step 2: Fitness computation Equation (3.12)

$$F = min(E) \tag{3.12}$$

1. Chemotaxis

Each bacteria travels gradually towards the goals by swimming and tumbling due to its ability to get away from dangerous items quickly and get nutrients. Bacteria travel continuously with defined run lengths while choosing one route at random within the search space, on the one hand. After falling,

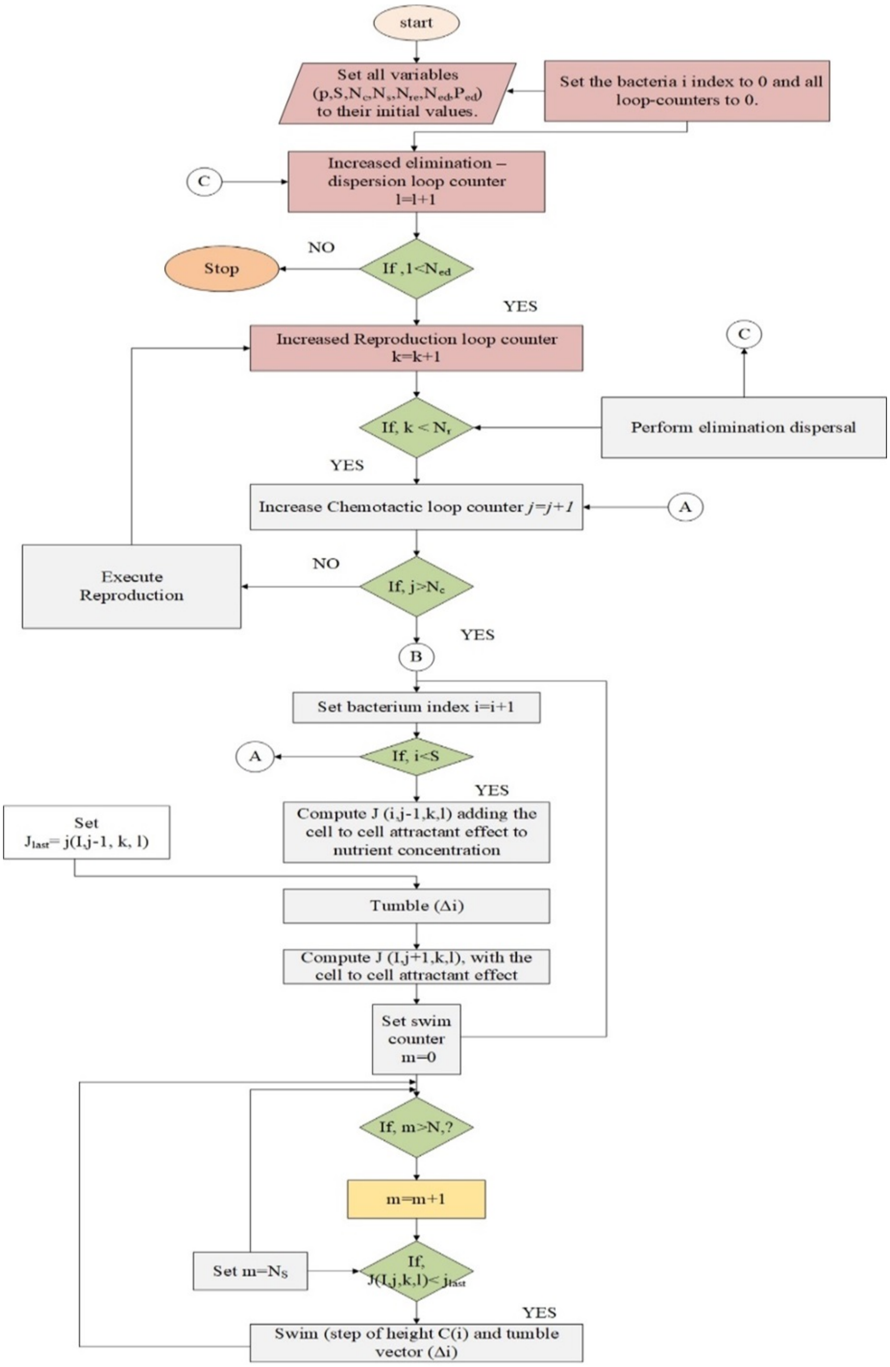
start
Set all variables (p,S,Nc,Ns,Nre,Ned,Ped) to their initial values.
Set the bacteria i index to 0 and all loop-counters to 0.
C
Increased elimination – dispersion loop counter l=l+1
NO
Stop
If ,1<Ned
YES
C
Increased Reproduction loop counter k=k+1
If, k < Nr
Perform elimination dispersal
YES
Increase Chemotactic loop counter j=j+1
A
NO
Execute Reproduction
If, j>Nc
YES
B
Set bacterium index i=i+1
A
If, i<S
YES
Set Jlast= j(I,j-1, k, l)
Compute J (i,j-1,k,l) adding the cell to cell attractant effect to nutrient concentration
Tumble (Δi)
Compute J (I,j+1,k,l), with the cell to cell attractant effect
Set swim counter m=0
If, m>N,?
m=m+1
Set m=NS
If, J(I,j,k,l)< jlast
YES
Swim (step of height C(i) and tumble vector (Δi)

Fig. 3.4: Bacterial Foraging Optimization

bacteria could not continue swimming in the same direction until their updated position became worse or until the number of possible moves reached the Nc limit. The new position of the bacterium i during the j+1th chemotaxis, kth reproduction, and lth elimination and dispersal, where $\theta^i(j, k, l)$ denotes the bacterium's previous position, C(i) denotes the step size, and i(i) denotes a random direction vector with all of its elements falling between -1 and 1. And the Equation (3.13) is combined by Lens Opposition-Based Learning.

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i) \times \Delta(i) \times \frac{\Delta T(i)}{(\Delta(i)^2 + \varepsilon)} \tag{3.13}$$

Initialize the fitness value J(i,j,k,l) to evaluate its performance.
Generate a random direction vector $\Delta(i)$ with elements ranging from -1 to 1.
Generate a random direction vector $\Delta T(i)$ with elements ranging from -1 to 1.
Update the position of bacterium i using Lens Opposition-Based Learning.
Update the step size for bacterium i are mentioned in Equation (3.14)

$$C(i) = C(i)(1+\delta) \tag{3.14}$$

2. Reproduction
   Following the central tenet of Darwin's "Survival of the Fittest" theory, the BFO algorithm's reproduction process reflects the fact that healthier bacteria are more likely to have the remarkable capacity for reproduction to sustain the entire swarm population, while undernourished individuals will ultimately be wiped out. The BFO algorithm records the bacterium's health level as $f_(i, health)$, which may be calculated from the total of fitness values over the course of its existence. In relation to this, the relevant mathematical statement may be shown as Equation (3.15)

$$f_{i,\text{ health }} = \sum_{j=1}^{N_c} J(i,j,k,l) \tag{3.15}$$

   where J(i,j,k,l) is the fitness value of the bacterium i in the j^{th} chemotaxis, k^{th} reproduction, l^{th} elimination and dispersal, and Nc is the total number of chemotaxises that the bacterium i undergoes over its career. When the health value of each bacterium is ranked in ascending order, half of the healthier bacteria (Sr=SS/2) can split into two bacteria with identical positions while the remaining bacteria are discarded.
   If $i \leq Sr$ it is denoted by healthier bacteria
   In this algorithm Triangle Walk Strategy is additionally combined with reproduction of the search agent hear we denoted the Equation (3.16)

$$\theta^i(j,k+1,1) = \theta^i(j,k,1) + \lambda \times \left(\theta^i(j,k,1) - \theta^i(j,k-1,1)\right) \tag{3.16}$$

   In this algorithm Levy Flight Walk Strategy is additionally combined with reproduction of the search agent hear we denoted the Equation (3.17)

$$\theta^i(j,k+1,1) = \theta^i(j,k,l) + \alpha \times L \times \Delta(i) \tag{3.17}$$

   Update the positions of bacteria based on the chosen strategy.
3. Elimination and Dispersal Actually, because of the constant, drastic shift in its environment, bacteria may be exposed to a variety of unanticipated dangers, such as the invasion of harmful substances or a change in the local area's dynamic temperature. As a result, when confronted with these unfavorable and unexpected circumstances, some germs must spread out as quickly as possible to a more favorable place. Based on it, the bacteria i randomly migrates to a new site $\Delta^{`}$ following the reproductive process with a specific probability P_{ed}, if not it stays in the existing position.
   For each bacterium i, In a uniform distribution between 0 and 1, a random number R is produced. If the generated number R is less than or equal to the dispersal threshold P_{ed}, the bacterium performs dispersal to a new position Δ^i using the Levy Flight Walk Strategy. A mathematical simulation of the movement of organisms based on heavy-tailed distributions is the Levy Flight Walk Strategy in Equation 3.18.

$$\theta^i = \theta^i(j,k,l) + \alpha X L X \Delta'_{(i)} \tag{3.18}$$

Else: Remain in the current position $\Delta^i(j,k,l)$.

4. Termination Repeat steps 3-5 until a termination condition is met, up to a certain number of iterations. Return the best solution found during the iterations.

3.4.2. Modified RNN. In a sequential user-item interaction-based recommendation system, the input layer processes the historical interaction data, capturing the sequential nature of user actions. The embedding layer transforms the user and item IDs into dense representations, enabling the model to capture latent features. LSTM layers are employed to effectively model temporal dependencies in the interactions, allowing the system to learn from past behaviors and preferences. An appropriate loss function, like mean squared error (MSE), is defined to quantify the prediction error and guide the model towards better recommendations. The fusion layer combines the final hidden states of the user and item embeddings, creating a joint representation that captures their interactions and preferences. Finally, the output layer leverages this fused information to predict user preferences or item similarities, providing personalized Providing appropriate suggestions for users based on their prior behaviour and previous interactions. The combination of these layers enables the recommendation system to learn from sequential data, handle temporal dynamics, and generate accurate predictions to enhance the overall user experience in the e-commerce platform.

3.4.3. Collaborative Filtering. In order to offer consumers items or services, Ratings, past purchases, and browser history are all used in personalised recommendation services. Furthermore, customers who have trouble selecting between different products and services would find a solution like this that provides personalised recommendations useful. Global companies like Netflix, Amazon, and Google make money by providing tailored suggestion services in e-commerce to assist users in making decisions.

The Collaborative Filtering component utilizes an autoencoder architecture to effectively capture user-item interactions in a recommendation system. The autoencoder comprises three key components: the Encoder, Decoder, and Output Layer. The Encoder transforms user and item IDs into lower-dimensional representations, effectively compressing the information while preserving essential patterns and relationships. The Decoder then reconstructs the original input from these encoded representations, allowing the autoencoder to learn the underlying structures in the data. Finally, the Output Layer leverages the learned representations to predict user preferences for items or similarities between items, aiding in generating personalized recommendations. By employing this autoencoder-based approach, the Collaborative Filtering component can efficiently extract meaningful features from user-item interactions, providing recommendation systems with more precise and customized recommendations.

Cosine similarity. Finding things that are comparable to those that the target user has liked or interacted with during the recommendation process and recommending those items are both steps in the process. Cosine similarity is one similarity measure that can be used to determine how similar two things are showed in Eq. (3.19).

$$\theta = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| \cdot ||\vec{b}||} \tag{3.19}$$

3.4.4. Item-based collaborative filtering. The idea behind the suggestion method known as "item-based collaborative filtering" is that buyers tend to favor goods that are similar to those they have already loved. This strategy uses the idea of item similarity to offer tailored recommendations. The technology finds things that have been often eaten or highly rated in tandem by examining the previous behavior and preferences of consumers. This information is then used to generate recommendations by suggesting products that are comparable to those the user has already indicated an interest in. Since this approach relies on item relationships rather than explicit user preferences, it has a number of benefits, including scalability and accuracy. Item-based collaborative filtering, which takes item similarity into account, enhances the system's ability to capture subtleties and offer pertinent recommendations.

3.5. Feature fusion. The collaborative filtering component, along with the outputs of the MLP, RNN, and Transformer-based Model components, can be integrated to produce an extensive ensemble model. A fusion approach that combines the individual model outputs, is used to achieve this integration. By combining the predictions from many models, we can take use of each architecture's advantages and provide a final prediction that is more reliable and accurate. By maximizing the aggregate intelligence of the multiple models, this ensemble technique enhances generalization and performance on a range of tasks and datasets.

4. Result and discussion. In this section, the results of the suggested procedures are compared to those of the current methods. The implementation is done with Python. Myntra Fashion Product Dataset comprising 14,481 samples was utilized, accompanied by 14,330 metadata entries, and subjected to a rigorous data cleaning process. The feature extraction phase resulted in 6,581 relevant features. For model training and evaluation, the dataset was split into two different sets: 70% for training and 30% for testing in one experiment, and 80% for training and 20% for testing in another experiment. This separation allowed for the assessment and comparison of the models' performance under different learning rate scenarios [26].

Several metrics, including Mean Squared Error (MSE), Mean square relative error (MSRE), Normalized Mean Squared Error (NMSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE), are used in this e-commerce recommendation system to evaluate the accuracy and efficacy of the recommendations. These metrics give us a way to gauge how closely the real user preferences match the projected recommendations, giving us important information about the performance of the system.

4.1. Performance matrices. Evaluation is done using error measures such as Mean Absolute Percentage Error (MAPE), Mean Square Error (MSE), Normalized Mean Square Error (NMSE), Mean Square Relative Error (MSRE), and Root Mean Square Error (RMSE).

MSE: Calculated is the average squared difference between the expected and actual values. The MSE loss, which is also used to direct model training, is used to evaluate a model's performance. The better the model fits the data, the lower the MSE shoun in Equation (4.1)

$$MSE = \frac{1}{N}\left(A_v - P_v\right)^2 \tag{4.1}$$

Where, A_v is the target variables actual value P_v , where N is the total number of values, and is the target variables predicted value.

MSRE: The Mean Square Relative Error (MSRE) is a metric used to measure the accuracy of a predictive model. It is particularly useful when evaluating models that make continuous predictions, such as regression models shown in Equation (4.2).

$$MSRE = \sqrt{\left(\sum^{\left[(\mathrm{y}_{\text{true}} - \mathrm{y}_{\text{pred}})^2 / \mathrm{y}_{\text{true}}{}^2\right]/\mathrm{n}}\right)} \tag{4.2}$$

NMSE: The Normalized Mean Square Error (NMSE) is a metric for comparing two signals that is frequently used to assess how well a prediction algorithm is working. It is described as the difference between the variance of the target signal and the MSE of the forecast signal shoun in Equation (4.3)

$$\text{NMSE } = \left(\frac{\mathbf{1}}{w}\right) * \operatorname{sum}\left(\frac{\left(A_v - P_v\right)^2}{\operatorname{var}\left(A_v\right)}\right) \tag{4.3}$$

RMSE: In order to evaluate how different disease symptoms affect the effectiveness of the treatments used at various disease severity levels, RMSE is used in this study in Equation 4.4

$$RMSE = \sqrt{\frac{1}{N}\left(A_v - P_v\right)^2} \tag{4.4}$$

MAPE: The MAPE formula is used (individually for each period) by multiplying the demand by the total number of distinct absolute errors in Equation (4.5)

$$MAPE = \frac{1}{T_j}\sum_{j=1}^{T_j}\left|\frac{A_v - F_v}{var(A_v)}\right| \tag{4.5}$$

Table 4.1: Metrices-Learn Rate (70%)

	BFO	EO	DBN	CNN	RNN	Proposed (HDCT)
MSE	0.3360	0.3237	0.3110	0.3264	0.3090	0.2971
MSRE	0.3007	0.2797	0.2883	0.3186	0.3319	0.2763
NMSE	0.4453	0.4224	0.4195	0.4515	0.4487	0.4013
RMSE	0.3604	0.3382	0.3878	0.3674	0.3893	0.3222
MAPE	0.3293	0.3172	0.3048	0.3199	0.3029	0.2911

Table 4.2: Metrices-Learn Rate (80%)

	BFO	EO	DBN	CNN	RNN	Proposed (HDCT)
MSE	0.2618	0.2499	0.2640	0.2718	0.2515	0.2403
MSRE	0.2262	0.2684	0.2577	0.2427	0.2332	0.2234
NMSE	0.3690	0.3919	0.3944	0.3889	0.3665	0.3506
RMSE	0.2050	0.2433	0.2336	0.2199	0.2113	0.2025
MAPE	0.2995	0.3119	0.2629	0.2710	0.2820	0.2597

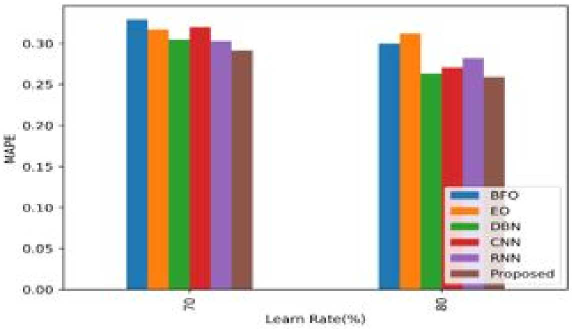

Fig. 4.1: Mean Squared Error analysis graph

where T_j is the total number of occurrences for the summation iteration, A_v is the actual value, F_v is the anticipated value.

4.2. Mean Squared Error. Table 4.1 shows that the techniques are represented by the models BFO, EO, DBN, CNN, RNN, and PROPOSED (HDCT) that are supplied. These approaches' respective mean squared error (MSE) values are 0.3360, 0.3237, 0.3110, 0.3264, 0.3090, 0.2971. It is clear that among the dataset's most recent 70%, the proposed approach has the lowest error consumption.

Table 4.2 shows that the models BFO, EO, DBN, CNN, RNN, and PROPOSED (HDCT) provided represent several methods. The corresponding mean squared errors (MSE) for these methods are 0.2618, 0.2499, 0.2640, 0.2718, 0.2515, and 0.2403. It is evident that among the most recent 80% of the dataset, the suggested strategy has the lowest error consumption.

The graph represents the 70% and 80% of the learn rate of in Mean Squared Error are represented in Fig. 4.1.

4.3. Mean square relative error. The models BFO, EO, DBN, CNN, RNN, and PROPOSED (HDCT) that are provided are shown in Table 4.1 to reflect the methodologies. Mean square relative error values for these methods are 0.3007, 0.2797, 0.2883, 0.3186, 0.3319, and 0.2763, respectively. It is evident that among the

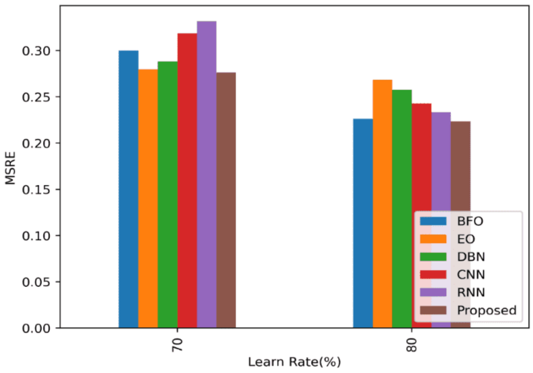

Fig. 4.2: Mean Squared relative Error analysis graph

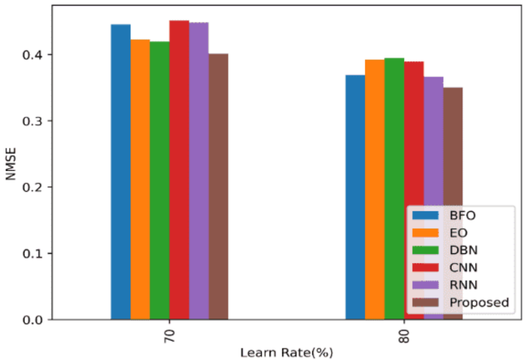

Fig. 4.3: Normalized mean square error

most recent 70% of the dataset, the suggested strategy has the lowest error consumption.

Table 4.2 demonstrates that the offered models BFO, EO, DBN, CNN, RNN, and PROPOSED (HDCT) represent several approaches. For these techniques, the associated Mean square relative error is 0.2262, 0.2684, 0.2577, 0.2427, 0.2332, and 0.2234. It is clear that the suggested technique has the lowest error consumption among the most recent 80% of the dataset.

The graph represents the 70% and 80% of the learn rate of in Mean square relative error are represented in Fig. 4.2.

4.4. Normalized mean square error. The models BFO, EO, DBN, CNN, RNN, and PROPOSED (HDCT) that are provided are shown in Table 4.1 to reflect the methodologies. Normalized mean square error values for these methods are 0.4453, 0.4224, 0.4195, 0.4487, 0.4013 respectively. It is evident that among the most recent 70% of the dataset, the suggested strategy has the lowest error consumption.

Table 4.2 shows that the available models BFO, EO, DBN, CNN, RNN, and PROPOSED (HDCT) represent several methodologies. The corresponding Normalized mean square error for these methods are 0.3690, 0.3919, 0.3944, 0.3889, 0.3665, and 0.3506. It is evident that among the most recent 80% of the dataset, the suggested technique has the lowest error consumption.

The graph represents the 70% and 80% of the learn rate of in Normalized mean square error are represented in Fig. 4.3.

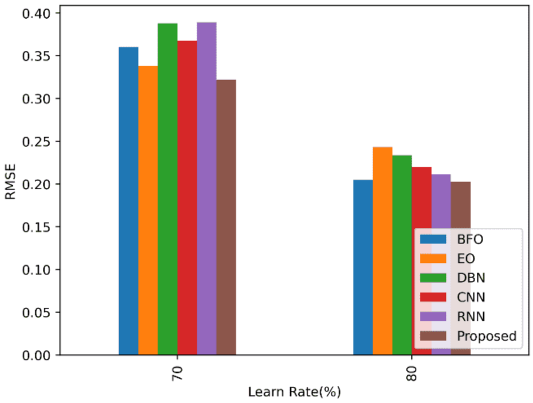

Fig. 4.4: Root Mean square error analyzing graph

4.5. Root Mean square error. Table 4.1 displays the models BFO, EO, DBN, CNN, RNN, and PROPOSED (HDCT) that are offered in order to illustrate the approaches. These methods' respective Root Mean Square Error values are 0.360416, 0.338242, 0.387806, 0.367417, 0.389381, and 0.322242. It is clear that the suggested technique has the lowest error consumption among the most recent 70% of the dataset.

The accessible models BFO, EO, DBN, CNN, RNN, and PROPOSED (HDCT) represent various methods, as shown in Table 4.2. For each of these approaches, the associated Root Mean Square Error is 0.2995, 0.3119, 0.2629, 0.2710, 0.2820, and 0.2597. It is clear that the suggested technique has the lowest error consumption among the most recent 80% of the dataset.

The graph represents the 70% and 80% of the learn rate of in Root Mean square error are represented in Fig. 4.4.

4.6. Mean absolute percentage error. Table 4.1 displays the models BFO, EO, DBN, CNN, RNN, and PROPOSED (HDCT) that are offered in order to illustrate the techniques. These methods' respective mean absolute percentage error values are 0.3293, 0.3172, 0.3048, 0.3199, 0.3029, and 0.2911. It is clear that the suggested technique has the lowest error consumption among the most recent 70% of the dataset.

Table 4.2 shows that the available models BFO, EO, DBN, CNN, RNN, and PROPOSED (HDCT) represent several methodologies. The corresponding Normalized mean square error for these methods are 0.2995, 0.3119, 0.2629, 0.2710, 0.2820, 0.2597. It is evident that among the most recent 80% of the dataset, the suggested technique has the lowest error consumption.

The graph represents the 70% and 80% of the learn rate of in Root Mean square error are represented in Fig. 4.5.

4.7. Existing result analysis . In the existing result analysis, two root mean square error (RMSE) values are presented: one labelled as "[5]" and the other as "PROPOSED."

RMSE: The first RMSE value, 0.2200, is associated with an unidentified method or model denoted by "[5]." Without further context or information, it is unclear what this value represents or what specific technique or model was used to obtain it.

PROPOSED: The second RMSE value, 0.2025, is associated with the method or model labelled as "PROPOSED." This value likely represents the root mean square error achieved by the proposed method or model in a certain experiment or analysis.

5. Conclusion. In conclusion, the Hybrid Deep Collaborative Transformer (HDCT), which has been suggested for e-commerce suggestions, performs well and exceeds other models already in use. But it's critical to recognize and deal with any mistakes or restrictions that can occur while optimizing. The HDCT approach can be improved and strengthened by carefully identifying and correcting these errors in e-commerce suggestions.

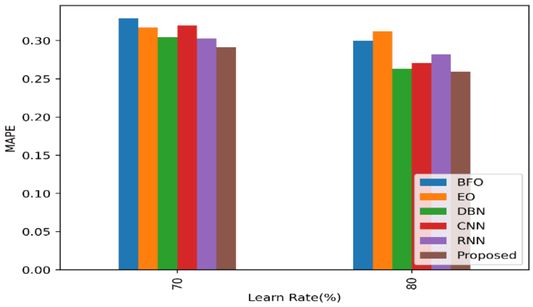

Fig. 4.5: Mean absolute percentage error Analysis

This focus on the little things and ongoing development guarantees that the HDCT model will continue to be dependable and effective in giving precise and individualized recommendations to e-commerce users. The identification and resolution of potential errors contribute to the overall robustness and trustworthiness of the HDCT method, enhancing its practicality and data for the RS is collected from the Myntra fashion product dataset ability in dataset e-commerce scenarios. As the field of e-commerce continues to evolve, it is imperative to remain vigilant in refining and advancing recommendation models like HDCT, ensuring optimal performance and customer satisfaction in the ever-growing digital marketplace.

Data Availability. The data availability statement is mentioned in the paper.

REFERENCES

[1] Ping, N.L. *Constructs for artificial intelligence customer service in E-commerce.* In 2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS) pp. 1-6, IEEE, 2019.

[2] Habib, A., Irfan, M. and Shahzad, M. *Modeling the enablers of online consumer engagement and platform preference in online food delivery platforms during COVID-19.* Future Business Journal, 8(1), 6, 2022.

[3] Liao, S.H., Widowati, R. and Hsieh, Y.C. *Investigating online social media users' behaviors for social commerce recommendations.* Technology in Society, 66, p.101655, 2021.

[4] Battisti, S., Agarwal, N. and Brem, A. *Creating new tech entrepreneurs with digital platforms: Meta-organizations for shared value in data-driven retail ecosystems.* Technological Forecasting and Social Change, 175, p.121392, 2022.

[5] Zhao, Q., Zhang, Y., Friedman, D. and Tan, F. *E-commerce recommendation with personalized promotion.* In Proceedings of the 9th ACM Conference on Recommender Systems pp. 219-226, 2015.

[6] Ehikioya, S.A. and Guillemot, E. *A critical assessment of the design issues in e-commerce systems development.* Engineering Reports, 2(4), p.e12154, 2020.

[7] Sharma, S., Singh, S., Kujur, F. and Das, G. *Social media activities and its influence on customer-brand relationship: an empirical study of apparel retailers' activity in India.* Journal of Theoretical and Applied Electronic Commerce Research, 16(4), pp.602-617.

[8] Deng, S., Tan, C.W., Wang, W. and Pan, Y. *Smart generation system of personalized advertising copy and its application to advertising practice and research.* Journal of Advertising, 48(4), pp.356-365.

[9] Tolstoy, D., Nordman, E.R., Hånell, S.M. and Özbek, N. *The development of international e-commerce in retail SMEs: An effectuation perspective.* Journal of World Business, 56(3), p.101165.

[10] Cui, Z., Xu, X., Fei, X.U.E., Cai, X., Cao, Y., Zhang, W. and Chen, J., 2020. Personalized recommendation system based on collaborative filtering for IoT scenarios. IEEE Transactions on Services Computing, 13(4), pp.685-695, 2020.

[11] Shahbazi, Z., Hazra, D., Park, S. and Byun, Y.C. *Toward improving the prediction accuracy of product recommendation system using extreme gradient boosting and encoding approaches.* Symmetry, 12(9), p.1566, 2020.

[12] Cui, Z., Xu, X., Fei, X.U.E., Cai, X., Cao, Y., Zhang, W. and Chen, J. *Personalized recommendation system based on collaborative filtering for IoT scenarios.* IEEE Transactions on Services Computing, 13(4), pp.685-695, 2020.

[13] Sohn, K. and Kwon, O. *Technology acceptance theories and factors influencing artificial Intelligence-based intelligent products.* Telematics and Informatics, 47, p.101324, 2020.

[14] Nguyen, A.T., Parker, L., Brennan, L. and Lockrey, S. *A consumer definition of eco-friendly packaging.* Journal of Cleaner Production, 252, p.119792, 2020.

[15] Grewal, D., Hulland, J., Kopalle, P.K. and Karahanna, E. *The future of technology and marketing: A multidisciplinary perspective.* Journal of the Academy of Marketing Science, 48, pp.1-8, 2020.

[16] Wang, K., Zhang, T., Xue, T., Lu, Y. and Na, S.G. *E-commerce personalized recommendation analysis by deeply-learned clustering.* Journal of Visual Communication and Image Representation, 71, p.102735, 2020.

[17] Li, Q., Li, X., Lee, B. and Kim, J. *A hybrid CNN-based review helpfulness filtering model for improving e-commerce recommendation Service.* Applied Sciences, 11(18), p.8613, 2021.

[18] Xiang, D. and Zhang, Z. *Cross-border e-commerce personalized recommendation based on fuzzy association specifications combined with complex preference model.* Mathematical Problems in Engineering, pp.1-9, 2020.

[19] Chen, Y. *Research on personalized recommendation algorithm based on user preference in mobile e-commerce* Retraction of Vol 18, Pg 837, 2020.

[20] Wang, J. and Zhang, Y. *Using cloud computing platform of 6G IoT in e-commerce personalized recommendation.* International Journal of System Assurance Engineering and Management, 12, pp.654-666, 2020.

[21] Guo, X., Wang, S., Zhao, H., Diao, S., Chen, J., Ding, Z., He, Z., Lu, J., Xiao, Y., Long, B. and Yu, H. *Intelligent Online Selling Point Extraction for E-Commerce Recommendation.* In Proceedings of the AAAI Conference on Artificial Intelligence Vol. 36, No. 11, pp. 12360-12368, 2022.

[22] Thongsri, N., Warintarawej, P., Chotkaew, S. and Saetang, W. *Implementation of a personalized food recommendation system based on collaborative filtering and knapsack method.* International Journal of Electrical and Computer Engineering, 12(1), pp.630-638, 2020.

[23] Zhou, L. *Product advertising recommendation in e-commerce based on deep learning and distributed expression.* Electronic Commerce Research, 20(2), pp.321-342, 2020.

[24] Chehal, D., Gupta, P. and Gulati, P. *Implementation and comparison of topic modeling techniques based on user reviews in e-commerce recommendations.* Journal of Ambient Intelligence and Humanized Computing, 12, pp.5055-5070, 2021.

[25] Zhang, Y., Abbas, H. and Sun, Y. *Smart e-commerce integration with recommender systems.* Electronic Markets, 29, pp.219-220, 2019.

[26] Dataset Available: (https://www.kaggle.com/datasets/hiteshsuthar101/myntra-fashion-product-dataset)

Edited by: Polinpapilinho F. Katina
Special issue on: Scalable Dew Computing for future generation IoT systems
Received: Dec 8, 2023
Accepted: Apr 1, 2024