International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Experimental Hybrid Technique for Enhancing the Quality of Personalized Product Recommendation System using Deep Learning

Yogesh Gurav*[1], S. K. Prashanth[2], Dr. Asma A. Shaikh[3], Dr. M. Ravichand[4], Kadam Vikas Samarthrao[5], Vinayak Biradar[6]

Submitted: 07/05/2023 **Revised**: 16/07/2023 **Accepted**: 05/08/2023

Abstract*:* Deep learning has recently gained a lot of grip in recommender systems. Hybrid recommendation systems, content-based recommendation and Deep learning were all used in multiple ways. Big data has been doing this for nearly 10 years, and the amount of available data on the network will be quickly growing. When challenged with complicated and huge data sets, it's indeed difficult for many people to obtain the necessary data rapidly. At this point, the recommendation system, including its features, is one of the most significant techniques for communicating with the large data overload issue. The development of recommendation algorithms has been aided by the growth of the e-commerce industry in particular. Traditional single recommendation algorithms are plagued by data sparsity, long-tail items and cold start. At this point, hybrid recommendation algorithms can efficiently keep away from some of the flaws of single algorithms. In response to these concerns, this paper proposes an experimental hybrid technique for enhancing the quality of the personalized product recommendation system algorithm that depends on deep learning IA-CNN to give back for a single collaborative model's limitations. To generalize and categorize the output results, first, the system employs a comprehensive approach, fusing product- and user-based collaborative filtering strategies. That is the methodology that the algorithm uses. Improved deep learning techniques are then used to capture nonlinear interactions between users and products that are more detailed and abstract. Finally, we devised experiments to test the algorithm's efficacy. Tests on the Amazon product rating dataset are performed against the benchmark algorithm, and the outcomes show that the proposed IA-CNN algorithm outperforms the on the test dataset, the benchmark algorithm was used for rating prediction.

Keywords: *Deep Learning, Recommendation System, IA-CNN Algorithm, Amazon product rating dataset, hybrid recommendation algorithm.*

1. Introduction

Today's exponential increases in the quantity of information available online, users are confronted with a serious problem of information overload [1]. Big data is still evolving and maturing. Other technologies like cloud computing have really improved our development and way of existence. Simultaneously, the data volume on the network has grown. The amount of information improved by humans during the last 10 years or so has surpassed the maximum volume of information produced in the earlier millions of years [2]. In general, when confronted with enormous amounts of data, people frequently struggle to discover what they are looking for and what they are interested in. Even if they do find it, it is unlikely that they will use it; it would be a lengthy process. A critical issue is how to provide users with useful information in an efficient manner. Google introduced search engine technology to address this issue. Users can search for items by entering keywords for content in order to find it quickly and accurately. This method, however, is only appropriate for situations in which users dynamically request information and need a good purpose. In reality, several users' information-seeking requirements are hazy, passive, and latent. Users may be unable to describe their situation or their interests, and they might not still dynamically look for the content they are seeking. Search engines are used in this case but are unable to provide users with useful information. One of the most efficient methods of resolving the issues listed above is through the use of a recommendation algorithm.

It is possible to model by analysing customer product ranking data, users' browsing history and further data such as choices of users, in order to uncover some potential personalised user requirements. Because such a recommendation algorithm is presently the most accepted on social networks and e-commerce websites, the research and application value of recommendation algorithms is

[1] *Professor, Dr. D. Y. Patil Technical Campus, Varale-Talegaon, Pune. Email: ybgurav1977@gmail.com*
[2] *Associate Professor, Dept of IT, Vasavi College of Engineering, Hyderabad. Email: sksspa21@gmail.com*
[3] *Assistant Professor, Marathwada Mitra Mandal's College of Engineering, Pune. Email: asma.ayyaj.shaikh@gmail.com*
Orcid id: 0000-0002-2474-8537
[4]*Professor of English, Mohan Babu University. erstwhile Sree Vidyanikethan Engineering College. Email: ravichandenglish@gmail.com*
Orcid Id: 0000-0002-9003-5359
[5] *Assistant Professor, Dept. of Computer Engineering, Sinhgad Institute of Technology, Lonavala. Email: vikasskadam@gmail.com*
[6]*Assistant Professor, Dept of IT, Vardhaman College of Engineering, Hyderabad, Telangana India. Email: vinayakbiradar2007@gmail.com*
** Corresponding Author Email: ybgurav1977@gmail.com*

substantial. Both collaborative filtering recommendation [3] and content-based recommendation [4] have demonstrated some success in finding the significance of users and items in traditional recommendation algorithms. However, given the fast increase in these traditional recommendation algorithms, which have been influenced by recent network data and the characteristics of "big data," they are no longer entirely appropriate in light of the current network situation.

To begin with, the information available is restricted. In today's social networks and e-commerce websites, the evaluation history of the user for a specific item is missing, as is the user's evaluation-related index, and the item has a large number of null values. Second, the algorithm does not properly handle new items added to the system or new users registered. Finally, while it is extremely rare for a user to consume a valuable item in the system, the potential profit by investors is substantial, and this type of problem is not specifically addressed by the algorithm. As a result, traditional recommendation algorithms have limited effectiveness. In order to overcome these issues, this study aims to provide a hybrid recommendation algorithm built on deep learning. The algorithm employs multi-recommendation-algorithm-fusing strategies and generalises and classifies multiple models' output results, thereby remunerating for the drawbacks of different algorithms. Then, the integration model posits that the technology of deep learning captures the more complex and fuzzy nonlinear interaction patterns between products and users. This is crucial because it solves the slow response problem and explains long tail items efficiently associated with conventional methods. Along these approaches, the algorithm's precision has been much improved. These are the most significant things that this article adds to the discussion:

i. A multi-recommendation algorithm integrated strategy model is planned, and on the dataset, pre-processing and pre-classification are applied to the model's output.

ii. The updated CNN deep learning model is accompanied by comparative experiments. IA-CNN is educated, and evaluation of advantages and disadvantages, as well as a comparative analysis with the conventional model is used.

The remaining sections of the paper are organised as described below: In the following sections, we will go over the related work done on this topic, the hybrid recommendation system's design and algorithm overview, the test, and finally, a summary of the results.

2. Related Work

A subcategory of data mining science influenced the recommendation algorithm and evolved into its own research field in the 1990s. To complete the recommendation task on the justification of clustering, an initial Group Lens has been proposed [5], and recommendation algorithm research has been ongoing. Algorithms for collaborative filtering are gaining traction. They are classified as either model-based or nearest-neighbor-based. To make recommendations for users, the latter measures the variation between customers and products in a data matrix. On this foundation, [6] proposed the item-based algorithm, that was successfully combined into Amazon's e-commerce system. The latter, besides measuring the similarity, it requires the development of a model, although it is easily extendable to massive data structures in resolving the recommended true issue. Huge amounts of data, on the other hand, can cause data sparsity and cold starts. [7] A collaborative filtering algorithm has been proposed that expands user-item capabilities by using the possible factor storage of the additional domain, resolving the restriction of only using a single special domain recommendation. Simon Funk [8] proposed, based on matrix decomposition, a collaborative filtering recommendation algorithm that significantly improved classification results. The probabilistic matrix decomposition model (PMF) [9] is the result of subsequent refinement efforts by the authors. Data sparsity has been overcome by the SVD++ model [10], MCFSAE [11], and others.

On the other hand, in the big data environment, they underperform with extremely limited rating data and are inadequate to deal with the "cold start" troubles of new products or new customers. With the expansion of online technology and social media platforms, a large volume of source data, such as personal interactions and user information, is now available. These have generated hybrid recommendation algorithms based on machine learning models, which incorporate collaborative filtering and content-based [12]. Clearly, the narrow levels of machine learning models will be incapable of learning the specific experiment inferred by customers and products. Many researchers have used deep learning to learn hidden patterns in recommendation systems because it generates outstanding results in hidden feature extraction [13].

Salakhutdinov et al. [14] became the first to suggest a collaborative filtering algorithm that is based just on a restricted Boltzmann machine (RBM) for having to discover the inherent variables of customers and products through recommender systems. [15] enhanced the classical CF technique with SVM and made progress in the research of bridge collaboration, but he neglected the source data. [16] Were using deep neural networks (DBN) and convolutional neural networks (CNN) to retrieve inherent variables from metadata. That is only useful for music data because it only considers the inherent variables of products. [17] Were using convolutional neural networks to obtain development in order to improve applicable data sets, which were integrated with a prediction model framework to make recommendations, but most of those model parameters may have to be individually changed. In this article, we propose a new deep hybrid recommendation strategy based on the first application of a Deep Belief Network (DBN) to recommender systems [18]. This method had an inherent data sparseness problem. [19] Constructed a collaborative deep learning (CDL) model to resolve this issue by incorporating the collaborative filtering technique and the Bayesian network model. DeepCoNN [20] has been proposed as a mechanism to enhance rating prediction by merging multiple CNN networks. [21] An in-depth look at the main effects that deep learning approaches have had on YouTube's video recommendation engine. Specifically, two neural networks are being used, one to generate YouTube candidate sets, and the other to rank those sets. The neural networks can operate independently of one another, and the ranking layer is not limited to the datasets used to generate the candidate sets. Experiment results show that a two-stage information retrieval system can improve recommendation impact while also providing users with a better customer experience when dealing with large amounts of data.

Research problems and possibilities are presented by the fact that deep learning techniques have been so effective in the field of personalized recommendation systems. Some progress has been made in the field of recommendation algorithms thanks to the application of deep learning algorithms, but much more might be done.

3. The Recommendation System model of IA-CNN

To address differences in recommendation results, several recommendation algorithms are typically coupled at different stages of the recommendation system. The four stages enumerated above are typically required in order for the process to consider making recommendations to customers focused on interesting items to customers under low-latency conditions. The specifications are seen in Figure 1. The recall process is primarily responsible for extracting the features of as many more accurate results as possible from the complete data collection and trying to return them to the "sort." The sorting stage is typically in charge of choosing the most extensive set of features from the testing set that conforms to the usage requirements of this process. The good ranking stage seems to be mostly responsible for ranking and classifying the present group from the collection of features in order to achieve the recommended outcomes. Estimates for items that have not been shown will deviate during the mechanism strategy stage, and problems might arise. As a result, at this stage, to filter and display products employing multiple recall techniques, re-ranking should be used.

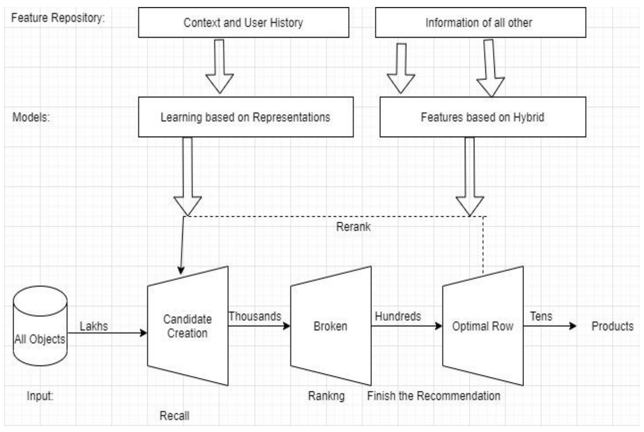

Fig 1. Process Model of a Recommendation System

i) Recall Processing Algorithm Hybrid Collaborative Filtering-Description

The traditional collaborative filtering algorithm would be enhanced during the recall phase. Collaborative filtering is a popular technology, particularly in recommender system classification, because the approach based on nearest neighbours is far more descriptive than the model-based method. The basic purpose of the nearest-neighbor collaborative filtering technique is to locate, across the entire structure, the k users and things with the most significant connections to the user or item under investigation. The primary goal is to find similarities. The majority of existing fundamental techniques are based on vector-based computation, which attempts to calculate the length between two feature vectors; the closer the distance, the stronger the correlation. We can compare products in recommendation systems by taking the customer's priority for all products as a variable or by taking all customers' choices for a single product as a vector. Its most frequently used approach is the similarity measure formula.

$$Similarity\ (m,\ n) = \frac{\sum_{j \in K} P_{m,j}}{\sqrt{\sum_{j \in K} P_{m,j}^2}} \qquad (1)$$

The correlation among user m and user n is requested from the above formula, in which K seems to be a product ranking with both customers, and Pm,j signifies customer m's rating on product j and Pm,j signifies customer n's rating on product j.

Here is the Pearson correlation equation:

$$Similarity\ (m,\ n) = \frac{\sum_{m \in M} P_{m,j} \cdot (P_{m,n} - P_n)}{\sqrt{\sum_{m \in M} (P_{m,j} - P_j)} \cdot \sqrt{\sum_{m \in M} (P_{m,n} - P_n)}} \qquad (2)$$

As in the previous equation, M signifies the number of customers who may have ranked, Pj signifies the average rating for product j, and Pj signifies the average rating for product j. The Pearson correlation coefficient, with a valuation of [1,-1], is being used to determine the measure of correlation between two customers' correlations.

To recall the dataset, this study employs hybrid collaborative filtering. The goal is to create a dataset that can be easily classified. Algorithm 1 contains a detailed description of the algorithm.

Algorithm 1:

Collaborative filtering using a hybrid approach

Input: Mounting M feature for user. Product feature J matrix, customer rating matrix Pj, customer list mI, and product list mJ are all examples of product feature matrix features.

Output: Remaining data set Data= {Data1, Data2,…,} (Data={ a1, a2, Pjm})

```
1.  Initially the similarity Matrix is empty i.e, Similarity PM and Similarity PJ
2.  Search the Product list List M and User list List J in $P_K$
3.  For m in set M and j in set J do
4.  Compute Similarity (jt, jx) and similarity (mt, mx);
5.  Measure all customers of similarity matrix Similarity PM * all products of Similarity matrix Similarity PJ
6.  Search the K-clustering M mS = {$m_1$,….., $m_k$, $m_k$+n} in Similarity PM and PJ
7.  Search the T-clustering J jS = { $j_n$, …, $j_t$, $j_t$+n} in Similarity PM and PJ
8.  To build a correlation matrix, relevant users and products are choser$Ph_{mk}^{jt}$*
9.  Repeat
10. Add similarity data in Data;
11. Until get all other Data
12. Final;
13. Return Data;
```

ii) The Rough Sorting Model is based on XGboost Classification Method

When used as a boosting technique, XGBoost should help the classification algorithm perform better [22]. This is due to the fact that regression trees form the basis of the real decision tree and that multithreading technology has the ability to boost learning performance. It has several benefits, together with large-scale data and personalised loss activities, in addition to the features of a faster rate and great results. It's an additive model made up of many base classifiers. I believe we're learning the decision tree for the tth iterative process, signified by gn(a), and we also have the following equations:

$$X_j^{(t)} = \sum_{h=1}^{t} g_h(a_j) = (t-1)_j^{(t-1)} + g_h(a_j) \qquad (3)$$

In the preceding formula, indicate the expected outcome of example $X_j^{(t)}$ behind $X_j^{(t-1)}$ tth iteration, the expected outcome of the (t-1) tth tree, and g_h (a_j) the representation of the tth tree.

The following is a description of XGBoost:

Specify the complexity of the tree:

$$g_t\,(a) = R_{s(a)},\ R\in P^Q,\ S\in P^d \rightarrow \{1,2,3,\dots,Q\} \qquad (4)$$

As in the previous equations represents a tree's configuration and R denotes the leaf strength.

To prepare for classification, Algorithm 1 is used to train the XGBoost model on the preprocessed data set. The tags are for the datasets of users and products. The goal is to make scoring model training easier while also improving the algorithm's classification efficiency. Algorithm 2 explains the algorithm in better detail.

Algorithm 2:

Classification of XGBoost

Input: Remaining data collection Data, user mL and product list jL

Output: Category for user descriptions MS, category for product descriptions JS

```
1.  Initiate with the same weight for training
2.  for sample data in Data do
3.  Initialize n times with XGBOOST to determine the erro $Error_n = (\sum Rjk\ (X_j \neq H_m y_i) / \sum R_j)$
4.  the learning rate should be updated
5.  Set the nth sample's weight to zero
6.  Repeat
7.  Create the XGBoost model based on the formulas' objective functions steps from 3 to 6
8.  Update the learning rate by obtaining the loss function's second-order temporary subset
9.  Up to Number n
10. add description label in MS and JS
11. final
12. return MS, JS
```

iii) Description of Classification Algorithm

The improved deep learning model recommendation algorithm's recall classification and processing will develop its quality. The model might be used without having to constantly evaluate its performance in terms of recall and categorization. The simulation results are invalidated by using the XGBoost classification algorithm, after the correlation has been determined independently, a similarity score has been generated, the correlation of each user to many other users and items has been evaluated, the simulation results have been clustered, the nearest neighbour set has been shaped, and the simulation results have been invalidated. The classification algorithm takes into account the customer's past actions, ratings, attributes, and interaction methods, as well as the relationships between customers and the interactions between consumers in different projects.

iv) An Improved CNN Model

The author suggested a novel deep network-CNN improved architecture (IA-CNN) that focuses on the learning algorithm for good ranking computation and can be used for real-time customer recommendation. The IA-CNN network model is depicted in Figure 2. At the first step, a hidden layer is implemented, which incorporates the hotspot matrix of the action feature from the previous layer into a dense vector. This is followed by a completely connected layer, which is constructed from two distinct feature spaces. By applying the convolutional layer to the word embeddings of the user and the object, new features are constructed to predict and capture the significantly more abstract and multidimensional user-product dialogue interaction. The convolutional layer's output can then be used by the focus pooling layer. Following that, we would go over each component of IA-CNN in detail.

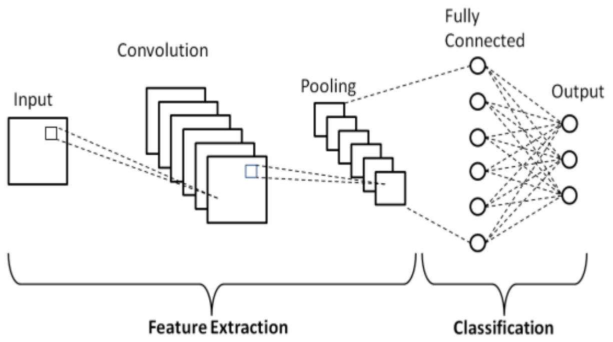

Fig. 2. Architecture diagram of IA-CNN

IA-CNN architecture is divided into two parts

a. A convolution tool is one that segregates and designates features of the product for observation and use in the feature extraction process.

b. A fully integrated layer that predicts the product category based on these features obtained in earlier phases using the outcome of the convolution process.

1. Convolutional Layer

This is the very first layer of the network, and its job is to learn about the input products by extracting features from them. The input product and a PxP filter are convolved arithmetically in this layer. Dragging the screen over the input product produces the matrix multiplication of the classifier and the input product's components with respect to the filter's size (PxP). The extracted features map contains information about the product, such as edges and corners. This convolution layer is then consumed by other layers, which learn a wide range of other attributes from the input data. The convolution layer is then consumed by other layers that also learn a wide range of other attributes about the input product.

2. Pooling Layer

It is standard procedure to use a pooling layer after one of these convolutional layers. To reduce the computational load, this layer's primary goal would be to reduce the dimension of the feature map used by the convolution layer. This is achieved by placing sole significance on the classifiers themselves and reducing the number of connections between the layers. Convolution layer tasks are classified based on how they will be carried out.

Max, the largest values are taken into account when pooling in a convolution layer. The overall factor estimate is determined by the pooling procedure within the constraints of a fixed product column size. The end outcome of the sum total pooling procedure is a computed total for all of the factors in a given segment. Connected to the convolutional layer is the FC layer via the pooling layer.

3. Fully connected Layer

The receptors, including the perceptions and weights, have been coupled from one layer to another by using a fully connected (FC) layer. Such layers are generally placed before the output neuron and represent the last several layers of an IA-CNN layout.

The earlier layers' efficiency products have been demolished and pushed towards the FC layer in this layer. The squashed vector then goes through so many more FC layers, which are usually where numerical feature operations are performed. This would be the reference point for the classification process.

4. Dropouts

Because all of the attributes have been associated with the FC layer, training and testing throughout the training dataset is frequent. Whenever a model is trained so well on training examples that it does have a negative effect, mostly on the model's efficiency while presented with new data, it is said to be overly accommodating.

A dropout layer has been implemented to deal with this problem. While the neural network is being trained, a small number of neurons are removed from this layer, leading to a more streamlined process overall. 30% of the neural network's nodes are randomly removed when the dropout threshold of 0.3 is reached.

5. Activation functions

One last important part of the CNN Algorithm is the activation function. Using this strategy, people can be put to work discovering and accurately estimating any constant or complex network link between network variables. In a summary, it calculates which design knowledge should be sent throughout the channel and what should not.

4. Experimental Study

This section contains a variety of experiments that are designed to evaluate the scientific validity of the proposed IA-CNN Algorithm. The following questions will be addressed during this experimental study:

a. Requirement1: In terms of productivity, does the IA-CNN model win over the benchmark model technique?

b. Requirement2: How perceptive is the IA-CNN model to specifications including embedding size, latent factors and degeneration?

i) Dataset

This paper measures the performance of IA-CNN using the very latest; nearly complete Amazon product dataset 2018 [23].The dataset is divided into two phases: user comment data and product metadata. This dataset contains data gathered by Amazon.com between 1996 and December of 2018. It currently has 29 different item categories and is the most accurate collection dataset for recommendation algorithm experimental tests.

The product description, which includes details namely the related products, product category, price, feature description, and product number, is the most important part of Amazon's product meta-data.

Researchers have used the Amazon review dataset for rating prediction tasks in previous works [24, 25, 26]. The customer's product of choice, the product number, and details regarding reviews, ratings, and the number of users are the most important parts of product reviews.

Because the label has been in place for so long, the original dataset is too big, and it contains a lot of unrated individuals and items. As a result, we used the Amazon-mini dataset's 5-core version in our experiment (At least five classification interactions exist for every customer or product). With only ten item classification, the volume of data within that version is approximately 0.33% of the source data; additionally, the size of the data remains extremely large, and the experimental error could be rejected due to a lack of proper datasets. Those unexpected conditions, duplicate entries, and unnecessary attributes have been eliminated from such an article prior to the word vector mapping in the particular experimental procedure to reduce the time and space required for the system to understand the data on a regular basis Only the data required for this experiment is saved, and a total of five classification models are obtained. All word removals and non-vocabulary words have been eliminated from the remark dataset. The dataset would be divided into three main sections in the following order: training, testing, and validation. The combined experimental and simulated research led to the creation of five distinct data types and four distinct data sets (1-class item, 3-class item, 5-class item, and full).

a. A comparison of experiments

To rebuild the equality of the experimental comparison, this article utilizes the very same collection of hardware and software platforms as the benchmark model. The following benchmark methods have been used for correlation: deep cooperative neural network (DCNN) (DeepCoNN), matrix factorization (MF), collaborative topic regression (CTR) and probabilistic matrix factorization (PMF). MF [27] and PMF [28, 26] are two of the most prominent datasets that have been used and very well standardized for collaborative filtering techniques. On the other hand, the significance of summary information mostly on recommendation algorithms could be confirmed. CTR [29] is a much more effective content modeling technique that models reviews and ratings while evaluating possible concepts. DeepCoNN [20] is presently a fairly advanced deep-learning-based rating prediction model. Furthermore, user-item representations have been learned by employing two simultaneous CNN models. The IA-CNN design concepts in this paper depended heavily on this method, which calculates reviews in the sharing layer using matrix factorization. We employ the free and open-source My Medialite component of a recommendation system to evaluate the model's performance for PMF and MF. The authors communicate the CTR and DeepCoNN source code on GitHub to perfect and rebuild the testing based on the available requirements set by the article's researchers.

The metrics of the model were initially fixed by a parameter sensitivity test, and then they were steadily improved upon during the experiment as intended. This was especially true for the amount of neurons in the convolutional neural network, as well as the distinctive measures of consumers and objects. The range for user-item attribute measurements is from 100 to 1000, while the range for the number of modules is from 100 to 600. After that, we set the number of modules to 200, the number of customers to 100, the packet size to 100, the dropout parameter to 0.7, the feature extraction parameter length to 4, the training rate to 0.002, the number of words to measure to 0.002, and the feature extraction parameter length to 0.002. These adjustments were made so that the maximum amount of productivity could be realized.

ii) Deep learning model experimental platform

Experiments with the deep learning approach provided in this work were run on a computer with an Intel i7-10700 processor. The model is written in Python and uses the Tensor Flow framework, while other requirements include a GTX 2070 graphics card, 32 GB of RAM, and Windows 10. The optimization algorithm employs Adam Optimizer. All algorithms have been fine-tuned except for their convergence phases; hence the training rate has been reduced to 0.002.

The Root Mean Square Error (RMSE as formula (17)) is being used to measure the accuracy of the estimation ranking since it can estimate the variation between the expected and reliable product, which would be classified as follows: the relatively small the measurement outcome, the improved the prediction accuracy and recommendation.

As the fundamental goal of the recommendation system is to produce the top-n recommendations for end users, we show that our proposed IA-CNN model makes use of evidence from both recall (formula (18)) and precision (formula (19)). The following is how they have been described:

$$Recall = \frac{TP}{TP+FN} \qquad (5)$$

$$Precision = \frac{TP}{TP+FP} \qquad (6)$$

Before recommending the top-n recommendation lists to each customer, we ranked all products based on their predicted popularity with that customer (where TP is the number of products the customer prefers in the top-N and FP is the number of products the customer uses) to assess the efficacy of the proposed hybrid model.

iii) Construction of Training Model

We try to tune and correlate the metrics in the CNN in the four categories enumerated below, because the specifications method does have a major impact on the study. The last model was constructed using experimental results.

a. Dropout's Impact

In neural network models [30] Dropout has been shown to be an effective way of dealing with over fitting. As a result, in order to test the effect of dropout, we vary the value of dropout.

Figure 3 depicts how allowing dropouts does have an impact on the decreasing number of false positives. As illustrated in Figure 3, the method may have significantly improved across all datasets. The benefit of RMSE, on the other hand, differs across datasets. Throughout many datasets, the method performs the best at approximately 0.56 or 0.64, so it accomplishes the lowest when the dropout is disabled. The effect of dropout is clearly less visible in a much bigger dataset compared to the smaller-scale sample. This is a visible occurrence. This needs to support earlier research [19] and the purpose of this system of dropout in small-scale datasets. As a result, the model's substitute value can range between 0.56 and 0.64, as seen in Figure 3.

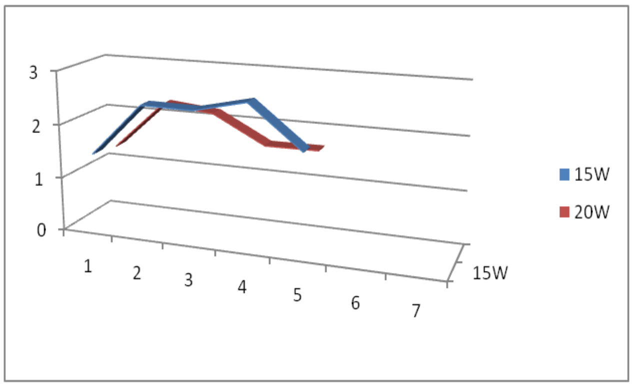

Fig. 3. The effect of Dropout

b. Dimension of Embedding

To evaluate the models responsiveness to variations in the word dimension of embedding, we assigned these values the range 100–700. As shown in Figure 4, the end result is as follows:

As illustrated in Figure 4, the model has a higher response in less than 500 word embedding measurements. The effects of sensitivity are more visible in data from a single category. This is not a difficult situation to comprehend. Here the complete data set model performs better because the recurrence of words in single class product data is low. The results also reveal that, among all datasets, efficiency is preferred at approximately 300 measurements and persists reasonably constant above 400. It is enough to make this product responsive to the word embedding factor. As a result, using larger values in the future did not result in a significant improvement in the model. Finally, 300 were selected as the word embedding.

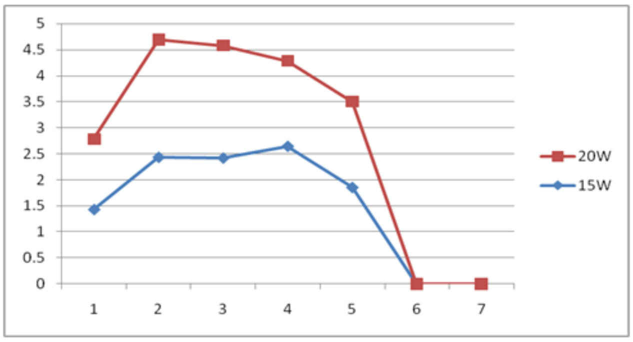

Fig 4. Effect on dimension of word embedding

c. Effect of Document Length

We started to run a verifying experiment with 5-class statistics and see how the duration of the records and information impacted the model. Modifying the duration of the document records the performance of the models and testing set. Figure 5 actually illustrates that whenever the full size of the document is set to 200, the performance is better and the spending time at this moment is much more acceptable; that is, the method is more reliable. This could furthermore demonstrate that, up until the requisite length of 400 is attained, additional document information can be effectively utilized for document concealed vector, despite the document size being bigger than 400. This could imply that hidden vectors in documents can be properly utilized for more document data up until the necessary length of 400 is attained. This is due to the fact that as the length of the document increases, so does the time required processing the information in the document. Furthermore, the model requires a significant amount of time to produce appropriate results. Accessing the information in a document, therefore, necessitates careful consideration of the trade-off between the time spent on preparation and the size of the document.

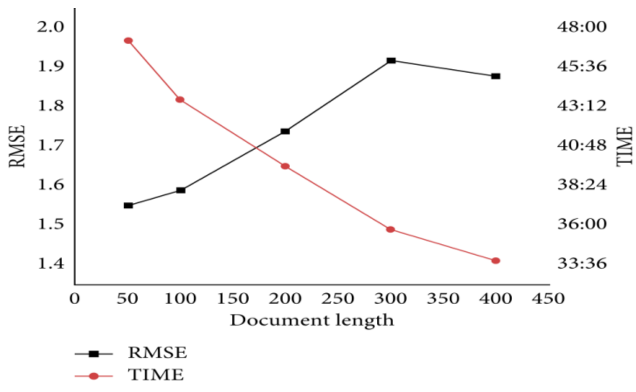

Fig 5. Effect of Document length

iv) Comparisons of Experimental analysis

In this article, we choose four important features of recommendation algorithms and run experiments to see how they perform under the conditions of the specified super standards. These four dimensions are a deep cooperative neural network (DeepCoNN), a collaborative topic regression (CTR) model, a matrix factorization (MF) model, and a hybrid matrix factorization (HMF) model. The goal of these tests is to verify the performance of the proposed IA-CNN algorithm. We may see examples of framework recommendation algorithms in the hybrid matrix factorization model and the matrix factorization model. It is possible to verify the effectiveness of review text and summary data on the recommendation algorithm when the rating matrix is constrained. This occurs when there are restrictions placed on the rating matrix. The proposed topic-based model recommendation algorithms include the collaboratively created approach. Topic models are used in the majority of recommendation algorithms that take into account review text. The proposed algorithm's advantages can be demonstrated when compared to them. The term incorporate vector technology, in particular, is being used to substitute the classification method in the preprocessing stage, with only a focus on the improving performance decided to bring by the sequence of words in the document; An example of a popular deep learning-based algorithm is DeepCoNN, a popular collaborative filtering method. While DeepCoNN serves as the foundation for the CNN algorithm suggested in this chapter, a linear correlation is also necessary. This process was repeated three times, with the product of the three outcomes used as the finished product. Experimental studies on multiple lengths of Amazon product review datasets generated the experimental results seen in Table 1.

In various datasets, the mean square error					
Models	10W	15W	20W	25W	Total RMSE
Our Methods	**2.052**	**1.431**	**1.352**	**1.263**	**1.186**
CTR	2.137	2.432	2.261	1.764	1.638
MF	2.465	2.417	2.161	1.831	1.764
HMF	2.371	2.642	1.643	1.047	1.468
DeepCoNN	2.651	1.853	1.649	1.872	1.542

Table 1. RMSE for various test sets

Naturally, we have chosen the Amazon product evaluation dataset to evaluate our product's quite compelling recall and precision with the benchmark experimental tests CTR, DeepCoNN, MF and HMF. Figures 6 and 7 have shown that even as n increases, so does the significance of recall. Our method performs them in the benchmark experimental tests for recall and precision.

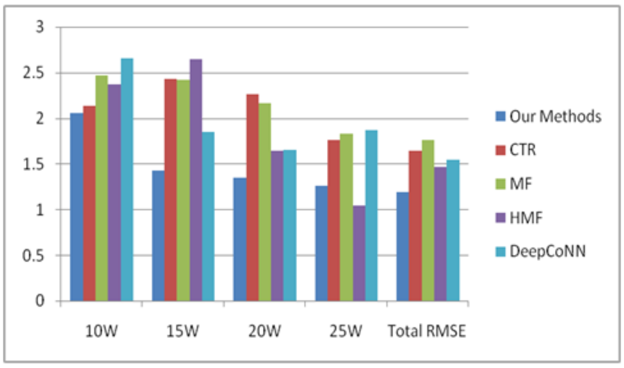

Fig 6. Comparison of Recall and Recall for Amazon Dataset

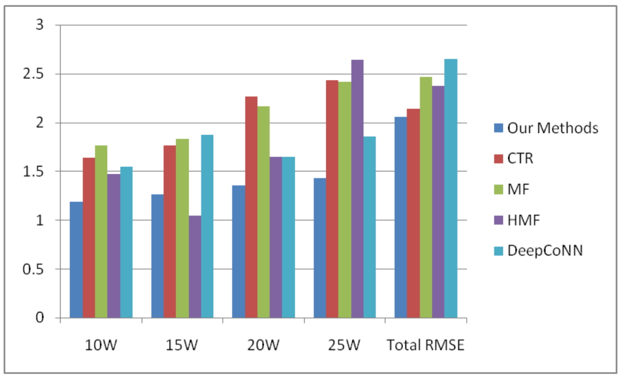

Fig 7. Comparison of Precision and Precision for Amazon Dataset

The following experiments were designed taking into account the fact that runtime is not a crucial criterion for evaluation. Several models have been used to train the 10w-full dataset, and twenty independent samples are typically selected at random from the testing machine for validation. Figure 8 displays the results of this method's temporal analysis. Under the same testing method, MF and HMF spent less time learning features when the machine learning approach began learning less features. DeepCoNN and our deep learning method, on the other hand, took slightly longer and the amount of time it takes in this article is higher than in the previous article because the algorithm will include a learning algorithm and computational complexity. Even though the recommendation could be made in 0.3 seconds, which really is an explicitly appropriate recommendation available time, suggesting that our algorithm is accurate and acceptable.

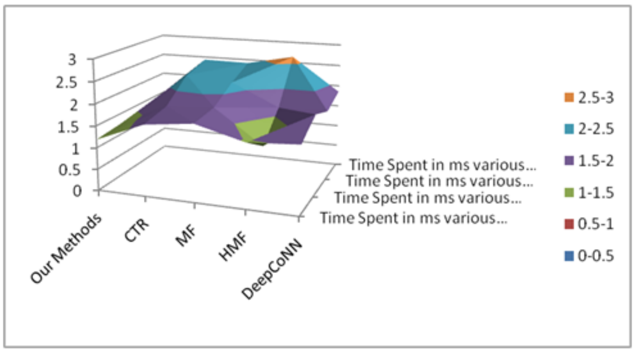

Fig 8. Comparison chart of Different Datasets

5. Conclusion

This article's objective is to evaluate how well existing personalized product recommendation algorithms deal with issues like data invariance, cold product starts, and long tail products by using the deep learning methods that have the appropriate features and can carry out complex tasks. The benefits of nonlinear transformation and its effective recognition capability can mitigate the impacts of sparse data and product cold start in existing product personalized recommendation systems. An improved CNN deep learning model is trained, a hybrid model of recommendation algorithms is developed, the models are tested, and the results are provided in this work. Although preliminary results are promising, research and use of deep learning technologies and methodology in recommendation systems are still in their development. Based on these previous improvements suggested by the experimental data, this still applies. This work proposes an experimental hybrid method for improving the quality of personalized product recommendation systems based on deep learning, but it still has phased and temporary issues and is not yet ready for use in actual recommendation systems.

References

[1] Xu, Y., Wang, Z. & Shang, J.S. PAENL: personalized attraction enhanced network learning for recommendation. Neural Comput & Applic 35, 3725–3735 (2023).

[2] X. Zeng, Y. Yang, S. Wang, T. He, and J. Chen, “A hybrid recommendation algorithm based on deep learning,” Computer Science, vol. 46, no. 01, pp. 126–130, 2019.

[3] Y. Shi, M. Larson, and A. Hanjalic, “Collaborative filtering beyond the user-item matrix,” ACM Computing Surveys, vol. 47, no. 1, pp. 1–45, 2014.

[4] S. Ruslan and A. Mnih, “Bayesian probabilistic matrix factorization using Markov chain Monte Carlo,” in Proceedings of the 25th International Conference on Machine Learning, pp. 880–887, Bled, Slovenia, 2008.

[5] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: an open architecture for collaborative filtering of netnews,” in Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, pp. 175–186, ACM, October 1994.

[6] B. Sarwar, G. Karypis, and J. Konstan, “Item-based collaborative filtering recommendation algorithms,” in Proceedings of the 10th International Conference on World Wide Web, pp. 285–295, Hong Kong, China, 2001.

[7] X. Yu, F. Jiang, J. Du, and D. Gong, “A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains,” Pattern Recognition, vol. 94, pp. 96–109, 2019.

[8] S. Funk, “Netflix update: try this at home,” 2006,https:// llsifter.org/∼simon/journal/20061211.html.

[9] A. Mnih and R. Salakhutdinov, “Probabilistic matrix factorization,” Advances in Neural Information Processing Systems, pp. 1257–1264, 2008.

[10] Y. Koren, “Factorization meets the neighborhood: a multi- faceted collaborative filtering model,” in Proceedings of the ACMSIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434, ACM, August 2008.

[11] M. Yu, T. Quan, Q. Peng, and X. Yu, “A model-based collaborate filtering algorithm based on stacked Autoencoder,” Neural Computing and Applications, pp. 1–9, 2021.

[12] S. Ahmadian, P. Moradi, and F. Akhlaghian, “An improved model of trust-aware recommender systems using reliability measurements,” in Proceedings of the 2014 6th Conference on Information and Knowledge Technology (IKT), pp. 98–103, IEEE, Shahrood, Iran, May 2014.

[13] Z. J. Sun, L. Xue, and Y. M. Xu, “Overview of deep learning,” Jisuanji Yingyong Yanjiu, vol. 29, no. 8, pp. 2806–2810, 2012.

[14] R. Salakhutdinov, A. Mnih, and G. Hinton, “Restricted boltzmann machines for collaborative filtering,” in Proceed- ings of the International Conference on Machine Learning, pp. 791–798, ACM, June 2007.

[15] X. Yu, Y. Chu, F. Jiang, Y. Guo, and D. Gong, “SVMs classification based two-side cross domain collaborative filtering by inferring intrinsic user and item features,” Knowledge- Based Systems, vol. 141, pp. 80–91, 2018.

[16] P. Chiliguano and G. Fazekas, “Hybrid music recommender using content-based and social information,” in Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2618–2622, IEEE, Shanghai, China, March 2016.

[17] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, “Convolutional matrix factorization for document context-aware recom- mendation,” in Proceedings of the 10th ACM Conference on Recommender Systems, pp. 233–240, Boston, MA, USA, September 2016.

[18] D. Chen, Research on Recommendation System Based on Deep Learning, Beijing University of Posts and Telecommunications, Beijing, China, 2014.

[19] H. Wang, N. Wang, and D. Y. Yeung, “Collaborative deep learning for recommender systems,” in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1235–1244, 2015.

[20] L. Zheng, V. Noroozi, and P. S. Yu, “Joint deep modeling of users and items using reviews for recommendation,” in Proceedings of the Tenth ACM International Conference on Web search and Data Mining, pp. 425–434, 2017.

[21] P. Covington, J. Adams, and E. Sargin, “Deep neural networks for youtube recommendations,” in Proceedings of the ACM Conference on Recommender Systems, pp. 191–198, ACM, 2016.

[22] T. Chen and C. Guestrin, “Xgboost: a scalable tree boosting system,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, San Francisco, CA, USA, 2016.

[23] Z. J. Sun, L. Xue, and Y. M. Xu, “Overview of deep learning,” Jisuanji Yingyong Yanjiu, vol. 29, no. 8, pp. 2806–2810, 2012.

[24] X. Zeng, Y. Yang, S. Wang, T. He, and J. Chen, “A hybrid recommendation algorithm based on deep learning,” Computer Science, vol. 46, no. 01, pp. 126–130, 2019.

[25] A. Mnih and R. Salakhutdinov, “Probabilistic matrix factorization,” Advances in Neural Information Processing Systems, pp. 1257–1264, 2008.

[26] Y. Koren, “Factorization meets the neighborhood: a multi- faceted collaborative filtering model,” in Proceedings of the ACMSIGKDD International Conference on Knowledge Discovery and Data

Mining, pp. 426–434, ACM, August 2008.

[27] Y. Shi, M. Larson, and A. Hanjalic, “Collaborative filtering beyond the user-item matrix,” ACM Computing Surveys, vol. 47, no. 1, pp. 1–45, 2014.

[28] S. Ruslan and A. Mnih, “Bayesian probabilistic matrix factorization using Markov chain Monte Carlo,” in Proceedings of the 25th International Conference on Machine Learning, pp. 880–887, Bled, Slovenia, 2008.

[29] C. Wang and D. M. Blei, “Collaborative topic modeling for recommending scientific articles,” in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 448–456, San Diego, CA, USA, August 2011.

[30] M. Kirienko, “Convolutional neural networks promising in lung cancer T-parameter assessment on baseline FDG-PET/ CT,” Contrast Media & Molecular Imaging 2018, 2018.

[31] Dattatraya, K.N., Rao, K.R. “Hybrid based cluster head selection for maximizing network lifetime and energy efficiency in WSN”, Journal of King Saud University - Computer and Information Sciences, 2022, 34(3), pp. 716–726

[32] Dattatraya, K.N., Raghava Rao, K, “Maximising network lifetime and energy efficiency of wireless sensor network using group search Ant lion with Levy flight”, IET Communications, 2020, 14(6), pp. 914–922.

[33] Ramkumar, J., Karthikeyan, C., Vamsidhar, E., Dattatraya, K.N.,” Automated pill dispenser application based on IoT for patient medication”, EAI/Springer Innovations in Communication and Computing, 2020, pp. 231–253.

[34] Dattatraya, K.N., Raghava Rao, K., Satish Kumar, D., “Architectural analysis for lifetime maximization and energy efficiency in hybridized WSN model”, International Journal of Engineering and Technology(UAE), 2018, 7, pp. 494–501.

[35] Dattatraya, K.N., Ananthakumaran, S., “Energy and Trust Efficient Cluster Head Selection in Wireless Sensor Networks Under Meta-Heuristic Model”, Lecture Notes in Networks and Systems, 2022, 444, pp. 715–735.

[36] Dattatraya, K.N., Ananthakumaran, S., Kiran, K.V.D., “Optimal cluster head selection in wireless sensor network via improved moth search algorithm”, Artificial Intelligence in Information and Communication Technologies, Healthcare and Education: A Roadmap Ahead, 2022, pp. 95–108.

[37] Anandpwar, W. ., Barhate, S. ., Limkar, S. ., Vyawahare, M. ., Ajani, S. N. ., & Borkar, P. . (2023). Significance of Artificial Intelligence in the Production of Effective Output in Power Electronics. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3s), 30–36. https://doi.org/10.17762/ijritcc.v11i3s.6152

[38] Christopher Davies, Matthew Martinez, Catalina Fernández, Ana Flores, Anders Pedersen. Machine Learning Approaches for Predicting Student Performance. Kuwait Journal of Machine Learning, 2(1). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/174

[39] Dhabliya, D. Security analysis of password schemes using virtual environment (2019) International Journal of Advanced Science and Technology, 28 (20), pp. 1334-1339.