



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

# IJRSET

International Journal of Innovative Research in  
**SCIENCE | ENGINEERING | TECHNOLOGY**

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Issue 10, October 2022

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

Impact Factor: 8.118

9940 572 462

6381 907 438

[ijirset@gmail.com](mailto:ijirset@gmail.com)

[www.ijirset.com](http://www.ijirset.com)

# Experimental Hybrid Approach for Improving the Quality of Personalized Product Recommendation Systems with Deep Learning

Prof. Vishal Paranjape, Prof. Saurabh Sharma

Dept. of Computer Science, Baderia Global Institute of Engineering & Management, Jabalpur, India

**ABSTRACT:** Deep learning has recently made significant advancements in recommender systems. Various methods, including hybrid recommendation systems, content-based recommendations, and deep learning, have been employed to enhance recommendation accuracy. Over the past decade, the proliferation of big data has led to a rapid increase in the amount of available data on the internet. However, dealing with complex and large data sets makes it challenging for many users to quickly find the information they need. At this juncture, recommendation systems play a crucial role in addressing the issue of data overload. The development of recommendation algorithms has been particularly driven by the growth of the e-commerce industry. Traditional single recommendation algorithms face challenges such as data sparsity, long-tail items, and cold starts. Hybrid recommendation algorithms can effectively mitigate some of these limitations. In response to these challenges, this paper proposes an experimental hybrid approach to improve the quality of personalized product recommendations using deep learning, specifically an Improved Attention-Convolutional Neural Network (IA-CNN). This method aims to address the shortcomings of single collaborative models. The algorithm first employs a comprehensive approach, combining product- and user-based collaborative filtering strategies to generalize and categorize the output results. This methodology captures detailed and abstract nonlinear interactions between users and products through advanced deep learning techniques. Finally, experiments were designed to test the algorithm's effectiveness. The IA-CNN algorithm was evaluated against a benchmark algorithm using the Amazon product rating dataset. The results demonstrate that the proposed IA-CNN algorithm outperforms the benchmark in rating prediction on the test dataset.

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

## I. INTRODUCTION

With today's exponential growth in the amount of information available online, users face a significant problem of information overload. Big data is continually evolving and maturing, and other technologies like cloud computing have greatly improved our development and way of life. At the same time, the volume of data on the internet has surged. The amount of information generated by humans in the past decade has exceeded the total volume produced in previous millennia. This method, however, is only suitable for situations where users actively seek information with a clear purpose. In reality, many users' information-seeking needs are vague, passive, and latent. Users might be unable to articulate their interests or may not actively search for the content they desire. While search engines are used in such cases, they often fail to provide users with useful information. One of the most effective ways to address these issues is through the use of recommendation algorithms.

By analyzing customer product ranking data, users' browsing history, and other information such as user preferences, it is possible to model and uncover potential personalized user requirements. Recommendation algorithms, which are widely used on social networks and e-commerce websites, have substantial research and application value. Both collaborative filtering recommendations and content-based recommendations have shown success in identifying the relevance of users and items in traditional recommendation algorithms. However, with the rapid growth of these traditional recommendation methods, influenced by recent network data and the characteristics of "big data," they are no longer fully suitable for the current network environment.

The remaining sections of the paper are organized 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.

## II. RELATED WORK

In the 1990s, a subfield of data mining science influenced the recommendation algorithm, eventually evolving into its own research area. An early effort in this field was the Group Lens, proposed to complete recommendation tasks based on clustering, leading to ongoing research in recommendation algorithms. Collaborative filtering algorithms have gained popularity and can be classified into model-based and nearest-neighbor-based methods. Nearest-neighbor algorithms measure differences between customers and products in a data matrix to make recommendations, while model-based algorithms develop models and are scalable to large data structures, addressing the recommendation problem more effectively. On this foundation, the item-based algorithm was successfully integrated into Amazon's e-commerce system.

However, large data volumes can lead to issues such as data sparsity and cold starts. To address these limitations, collaborative filtering algorithms have been proposed that expand user-item capabilities by incorporating latent factor models from additional domains, overcoming the restriction of using a single special domain recommendation. For instance, Simon Funk proposed a collaborative filtering algorithm based on matrix decomposition, significantly improving classification results. This work led to the development of the probabilistic matrix factorization (PMF) model. Further advancements like the SVD++ model and MCFSAE have also addressed data sparsity issues.

Despite these improvements, traditional models struggle in the big data environment, particularly with limited rating data and cold start problems for new products or users. The expansion of online technology and social media platforms has introduced vast amounts of data, such as personal interactions and user information, leading to the development of hybrid recommendation algorithms that combine collaborative filtering and content-based methods using machine learning models. However, basic machine learning models often fail to capture the complex interactions between users and products.

Deep learning has emerged as a powerful tool for extracting hidden patterns in recommendation systems. Salakhutdinov et al. were pioneers in proposing a collaborative filtering algorithm based on a restricted Boltzmann machine (RBM) to uncover latent variables of users and products. Further enhancements include integrating classical CF techniques with SVM and employing deep neural networks (DBN) and convolutional neural networks (CNN) to retrieve inherent variables from metadata, though these methods had their limitations. For instance, deep neural networks were mainly effective for specific data types, such as music.

Recent efforts include using CNNs to improve applicable data sets and integrate them with prediction models. However, these models often required parameter adjustments. To address these challenges, a new deep hybrid recommendation strategy was proposed, utilizing a Deep Belief Network (DBN) to overcome data sparseness. Additionally, the collaborative deep learning (CDL) model was developed to combine collaborative filtering with Bayesian networks. Another advancement, DeepCoNN, used multiple CNN networks to enhance rating predictions. A significant application of deep learning in recommendation systems is seen in YouTube's video recommendation engine, which employs two neural networks—one for generating candidate sets and another for ranking them. This two-stage information retrieval system has shown to improve recommendation impact and user experience when handling large data volumes.

The success of deep learning techniques in personalized recommendation systems presents numerous research challenges and opportunities. While deep learning algorithms have advanced recommendation algorithms, there remains significant potential for further development and application in this field.

## III. THE RECOMMENDATION SYSTEM MODEL OF IA-CNN

To reconcile differences in recommendation results, multiple algorithms are usually integrated at various stages within the recommendation system. The process typically involves four stages to deliver recommendations of relevant items to customers efficiently. These stages are depicted in Figure 1. The recall stage is responsible for extracting features from the entire dataset to find as many accurate results as possible, which are then forwarded to the sorting stage. In the sorting stage, the most relevant set of features from the testing data is selected to meet the process's requirements. The ranking stage then ranks and classifies the features to produce the final recommendations. During the mechanism strategy stage, there might be discrepancies in the estimates for items not yet displayed, which can lead to potential issues. To mitigate this, re-ranking should be applied at this stage to filter and present products using various recall

techniques.

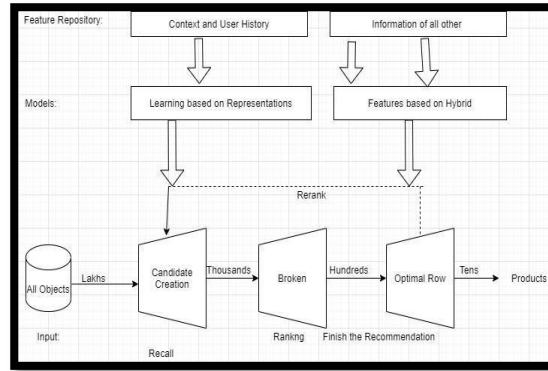


Fig 1. Process Model of a Recommendation System

**i) Recall Processing Algorithm Hybrid Collaborative Filtering-Description**

In the recall phase, traditional collaborative filtering algorithms can be enhanced. Collaborative filtering is widely used

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

in recommender systems due to its effectiveness, particularly the nearest-neighbor approach, which is often more descriptive than model-based methods. The core objective of nearest-neighbor collaborative filtering is to identify the k users or items with the strongest connections to the user or item being examined. The focus is on finding similarities. Most existing fundamental techniques rely on vector-based calculations, aiming to measure the distance between two feature vectors; a smaller distance indicates a stronger correlation. In recommendation systems, products can be compared by treating a customer's preferences for all products as a variable or by considering all customers' choices for a single product as a vector. The similarity measure formula is the most commonly used method for this purpose.

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 P<sub>m,j</sub> signifies customer m's rating on product j and P<sub>n,j</sub> signifies customer n's rating on product j.

Here is the Pearson correlation equation:

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

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

**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 t<sup>th</sup> iterative process, signified by g<sub>n</sub>(a), and we also have the following equations:

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

In the preceding formula, indicate the expected outcome of example I behind the t<sup>th</sup> iteration, the expected outcome of the (t-1) t<sup>th</sup> tree, and g<sub>h</sub>(a<sub>j</sub>) the representation of the t<sup>th</sup> tree.  $X_j^{(t)}$       $X_j^{(t-1)}$

The following is a description of XGBoost: Specify the complexity of the tree:

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

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

### iii) Description of Classification Algorithm

The enhanced deep learning model for recommendation algorithms will improve its recall classification and processing quality. This model can be utilized without the need for continuous performance evaluation in terms of recall and classification. However, simulation results are invalidated when using the XGBoost classification algorithm. After determining the correlation independently, generating a similarity score, and evaluating each user's correlation with other users and items, the simulation results are clustered. The nearest neighbor set is then formed, and the simulation results are invalidated. The classification algorithm considers customer history, ratings, attributes, interaction methods, and the relationships between customers, as well as interactions across different projects.

### iv) An Improved CNN Model

The author proposed a new deep learning architecture, the Improved Attention-Convolutional Neural Network (IA-CNN), designed for effective ranking computation and real-time customer recommendations. The IA-CNN model is illustrated in Figure 2. Initially, a hidden layer integrates the hotspot matrix of action features from the previous layer into a dense vector. This is followed by a fully connected layer, which is derived from two separate feature spaces. By applying the convolutional layer to the word embeddings of both users and items, new features are generated to predict and capture more abstract and multidimensional interactions between users and products. The output from the convolutional layer is then processed by the attention pooling layer. Next, each component of the IA-CNN will be examined in detail.

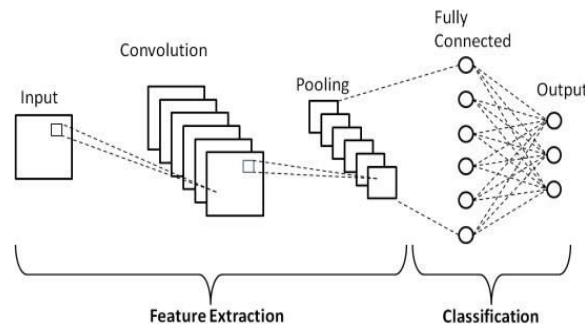


Fig. 2. Architecture diagram of IA-CNN

- A convolution tool is one that segregates and designates features of the product for observation and use in the feature extraction process.
- A fully integrated layer that predicts the product category based on these features obtained in earlier phases using the outcome of the convolution process.

## IV. 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:

- Requirement1: In terms of productivity, does the IA-CNN model win over the benchmark model technique?
- Requirement2: How perceptive is the IA-CNN model to specifications including embedding size, latent factors and degeneration?

### i) Dataset

This paper evaluates the performance of the IA-CNN model using the latest, nearly comprehensive Amazon product dataset from 2018. The dataset is split into two parts: user comment data and product metadata. Collected by Amazon.com from 1996 to December 2018, it includes 29 distinct item categories and is highly accurate for experimental tests of recommendation algorithms. The product description, encompassing details such as related

products, category, price, feature description, and product number, is a crucial element of Amazon's product metadata. Previous research has utilized the Amazon review dataset for rating prediction tasks, focusing on elements such as the customer's product choice, product number, review details, ratings, and the number of users. Due to the extensive size of the original dataset and the presence of many unrated individuals and items, we utilized the 5-core version of the Amazon-mini dataset for our experiment. This version includes at least five interactions for each customer or product. It contains approximately 0.33% of the original dataset's volume, yet remains quite large. To mitigate experimental errors caused by inadequate datasets, we removed unexpected conditions, duplicate entries, and irrelevant attributes before performing word vector mapping. This was done to streamline the data processing and reduce time and space requirements. Only the data necessary for the experiment was retained, resulting in five classification models. All irrelevant words and non-vocabulary terms were removed from the review dataset. The dataset was divided into three main parts: training, testing, and validation. The combined experimental and simulated research produced five distinct data types and four distinct datasets: 1-class item, 3-class item, 5-class item, and full.

### a. A comparison of experiments

#### ii) Deep learning model experimental platform

To ensure consistency in the experimental comparison, this study uses the same hardware and software platforms as the benchmark model. The benchmark methods employed for comparison include Deep Collaborative Neural Network (DCNN) or DeepCoNN, Matrix Factorization (MF), Collaborative Topic Regression (CTR), and Probabilistic Matrix Factorization (PMF). MF and PMF are well-established techniques for collaborative filtering, having been widely used and standardized. CTR, on the other hand, is a more effective content modeling approach that incorporates review and rating data to evaluate potential concepts. DeepCoNN is a sophisticated deep learning-based rating prediction model that uses two concurrent CNN models to learn user-item representations. The IA-CNN design presented in this paper heavily relies on this approach, which involves calculating reviews in the sharing layer through matrix factorization. For evaluating PMF and MF models, we use the open-source My Medialite component of a recommendation system. The CTR and DeepCoNN source codes are available on GitHub, allowing researchers to refine and reconstruct the tests according to the requirements specified in this article.

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:

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

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

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 illustrates the effect of implementing dropout on reducing the number of false positives. As shown, the method has notably improved across all datasets. However, the benefit of RMSE varies among datasets. The method generally performs best with an RMSE around 0.56 to 0.64, achieving its lowest values when dropout is disabled. The impact of dropout is less pronounced in larger datasets compared to smaller ones, which is a noticeable trend. This supports

previous research [19] regarding the effectiveness of dropout in small-scale datasets. Consequently, the model's performance can fluctuate between 0.56 and 0.64, as depicted in Figure 3.

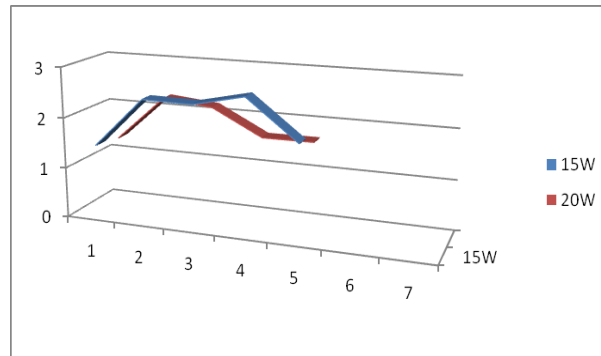


Fig. 3. The effect of Dropout

#### 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.

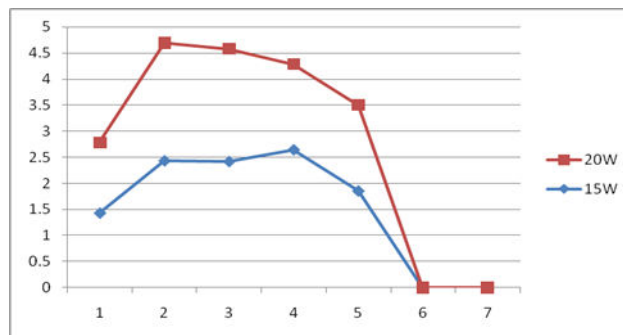


Fig 4. Effect on dimension of word embedding

#### c. Effect of Document Length

We conducted a verification experiment using 5-class statistics to examine how the duration of records and information affects the model's performance. By adjusting the duration of the document records, we assessed the impact on both model performance and testing accuracy. Figure 5 shows that when the document size is set to 200, the model's performance improves, and the processing time becomes more manageable, indicating greater reliability. This suggests that additional document information can be effectively utilized for document vector representation up to a length of 400. Beyond this length, while the document size increases, the hidden vectors can still be effectively employed, but processing time also rises. Consequently, the model requires more time to deliver accurate results. Therefore, balancing the trade-off between preparation time and document size is crucial for accessing and utilizing document information efficiently.

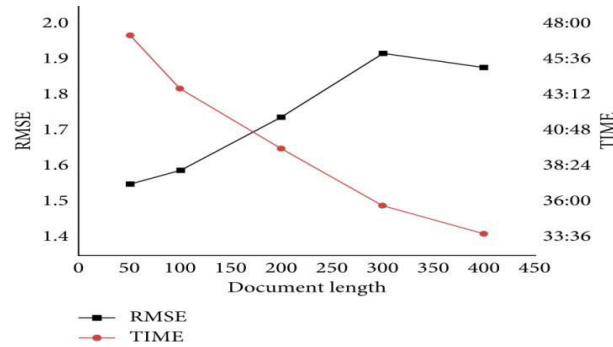


Fig 5. Effect of Document length

iv) Comparisons of Experimental analysis

In this article, we examine the performance of four key recommendation algorithm features under specified super standards. The four models tested are the Deep Collaborative Neural Network (DeepCoNN), Collaborative Topic Regression (CTR), Matrix Factorization (MF), and Hybrid Matrix Factorization (HMF). The aim of these experiments is to evaluate the performance of the proposed IA-CNN algorithm. Examples of recommendation frameworks can be observed in both the hybrid matrix factorization and matrix factorization models. The effectiveness of review text and summary data on recommendation algorithms is assessed, particularly when the rating matrix is constrained. Topic-based recommendation algorithms, which include the collaborative approach and use topic models for review text, are compared with the proposed IA-CNN algorithm to highlight its advantages. The IA-CNN algorithm incorporates vector technology to replace the classification method in the preprocessing stage, focusing on improving performance through word sequence analysis in documents. DeepCoNN, a widely used deep learning-based collaborative filtering method, serves as the basis for the CNN algorithm discussed in this chapter, though linear correlation is also considered. This process was repeated three times, with the combined results of these trials forming the final outcome. Experimental results from various lengths of Amazon product review datasets are presented in Table 1.

In various datasets, the mean square error					
Models	10W	15W	20W	25W	Total RMSE
Our Methods	<b>2.052</b>	<b>1.431</b>	<b>1.352</b>	<b>1.263</b>	<b>1.186</b>
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.



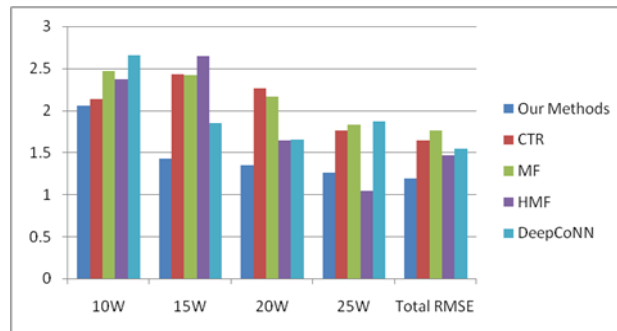


Fig 6. Comparison of Recall and Recall for AmazonDataset

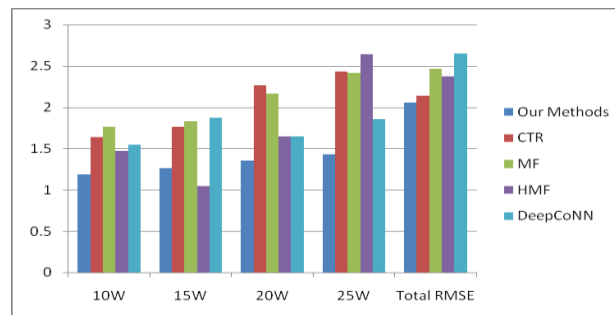


Fig 7. Comparison of Precision and Precision for AmazonDataset

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.

## V. CONCLUSION

The goal of this article is to assess how effectively current personalized product recommendation algorithms address challenges such as data invariance, cold starts for new products, and long tail products using deep learning techniques with relevant features capable of handling complex tasks. Nonlinear transformations and their efficient recognition capabilities are shown to alleviate the effects of sparse data and cold starts in existing personalized recommendation systems. This study involves training an improved CNN deep learning model, developing a hybrid recommendation model, testing these models, and presenting the results. Although the initial findings are promising, the application of deep learning technologies and methodologies in recommendation systems is still evolving. Based on the advancements highlighted by the experimental data, this work proposes a hybrid approach to enhance the quality of personalized product recommendation systems. However, it remains in a developmental phase and is not yet suitable for implementation in practical 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://lmsifter.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 ACM SIGKDD 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 collaborative 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 *Proceedings 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 recommendation," 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.



**INNO SPACE**  
SJIF Scientific Journal Impact Factor  
**Impact Factor: 8.118**



**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details