Journal of Scientific & Industrial Research
Vol. 84, December 2025, pp. 1301-1309
DOI: 10.56042/jsir.v84i12.4513

Trinity Fusion: A Deep Learning based Triad Collaborative Filtering System for Product Recommendations

Mamta* & Suman Sangwan

Department of Computer Science and Engineering, Deenbandhu Chhotu Ram University of Science and Technology, Murthal, Sonipat 131 039, Haryana, India

Received 01 August 2023; revised 27 November 2025; accepted 18 December 2025

Recommender Systems (RS) enhance user experience by presenting relevant products based on their preferences and past interactions. Among various techniques, Collaborative Filtering (CF) is widely used due to its ability to generate personalized recommendations using the preferences of similar users. However, traditional CF methods, such as stochastic matrix factorization, model user–item relationships linearly and suffer from limitations including low learning efficiency, cold-start issues, and data sparsity. Traditional cooperative filtering is combined with advanced neural networks and terminology in order to solve these issues. "A Weighted Parallel Deep Triad Collaborative Filtering Model based on Singular Value Decomposition (SVD), Restricted Boltzmann Machine (RBM) and Ontology-based term weighting technique (OBTW)is proposed for significant improvement."A user-item evaluation matrix is built from scratch. Singular Value Decomposition (SVD) is merged with the user-item matrix to provide a low-rank estimate of the matrices. It is followed by developing latent characteristics to anticipate consumer tastes by integrating the matrix of user items using RBM. OBTW consists of ontology development, weighting scheme and classifier module. Root Mean Squared Error(RMSE), Mean Absolute Error(MAE), Precision, Recall, F1-Measure and Normalized discounted cumulative gain (NDCG) are used to assess the accuracy of the proposed model on the Amazon Product Reviews dataset. The proposed model is compared with recent studies and existing collaborative filtering methods such as SVD, RBM and Probabilistic matrix factorization (PMF). It achieves lower RMSE (0.9443) and MAE (0.6589), along with higher Precision (93%), Recall (95%), and F1-score (93%), demonstrating improved recommendation accuracy and effectiveness.

Keywords: Collaborative filtering, Ontology, Restricted Boltzmann machine, Recommender system, Singular value decomposition

Introduction

The RS is a highly effective method for generating decisions. The issue of sensory overload is exacerbated by the rapid expansion of online data. Many RS are being created to assist customers in this area.[1] To tailor suggestions to each individual consumer according to their personal information and past actions, recommender systems are increasingly being put to use. Such techniques were included into a wide range of online goods and services in order to improve the user experience while searching for and selecting specific media. Amazon, Netflix, YouTube, and many social networks are just some of the sites that have included recommender systems. There has been a body of scholarship on the topic of music recommendation systems, which have been used to help people find new music.[2,3]

Depending on the kind of filtering technique applied in RS, such as Collaborative Filtering (CF), Content-Based Filtering (CBF), and Demographic Filtering (DF), the input may include the consumer's assessing (numerical value), demographics (age, gender, occupation), along with the content info, that includes written evaluations of associated items rated by consumer. Forecasting and top suggestion are both possible outcomes for recommender systems. The predicted viewpoint of engaged users is represented by a quantitative value known as a forecast. A recommendation is a representation of the top N items that are most likely to be enjoyed by the currently logged-in users.

Among the many methods used by modern personalised recommender systems, collaborative filtering (CF) stands out.[3] Memory-based and model-based categories both apply to this technique. Collaborative filtering is based on the premise that in the event that two individuals possess similar tastes in previous times, they are going to have similar tastes in the future. These suggestions take into account not just user history but additionally both implicit and

*Author for Correspondence
E-mail: 19001901004mamta@dcrustm.org

explicit evaluations provided by the users themselves. The user-item matrix in the classic CF is constructed using just the users' stated preferences for individual items. Cold-start and data sparsity issues are two of CF's drawbacks. When making suggestions for a new person, item, or community, you may run into the "cold-start" problem. If a new user or product is added, the system will not have enough information to make educated guesses about the person's or product's preferences. Without any individual feedback, the user-item matrix is extremely sparse, leading to a significant decrease in recommendation accuracy.

Among the most well-known forms of collaborative filtering involves Matrix Factorization (MF). When dealing with excessively sparse data, the MF method is among the most effective.[4] By scaling the user-item interaction matrix, MF may map people and things to a shared latent component field of dimensions. Therefore, this approach effectively unearths the fundamental patterns in the data.

Knowledge Gaps

Recent studies in recommendation systems research has focused on developing Deep Neural Networks (DNNs) based on cooperative filtering.[5,6] But conventional CF strategies and deep-learning often employ a sequential learning strategy or educate the full data set all at once to perfect their algorithms. Due to information's exponential development, the system must periodically store unending flows of content. There is a constant ebb and flow to users' interests and inclinations. Both classic CF and Deep Learning need starting the training process again whenever fresh data is added to the system. They have to spend a lot of money to update and modify the variables in the model using these approaches.[6] Fuzzy interfering rules are used to improve suggestion reliability,[7] since the sparseness of data is one of CF's main drawbacks. This model begins by classifying people into those who enjoy the model, those who despise the model, and those who agree or disagree with the same things as the majority of users. Product ratings and the relationship of the recommended matrix are both significantly affected by the condition of intrusion. Many recommendation algorithms centre on data like user ratings, user histories, item details, and user/item contextual data. This data does not provide enough user-item relation exposure for reliable forecasting. By assigning various amounts of weight to each recommendation method and then adding these weights to provide a new output suggestion, hybrid weighting is a hybridization methodology that calculates the prediction value across all recommendation methods. *A Weighted Parallel Deep Triad Collaborative Filtering suggestion approach is presented to improve the cold start issue*.

Related works

The application of RBM was proven by Shetty *et al.*[8] who achieved a 6% improvement in scoring compared to Netflix's benchmark predictor methodology. It is also used as a foundation for deep learning models[10–12] and deep learning framework.[9] To deal with the two-sided nature of user and product behaviour, Shen *et al*.[13] devised a dual RBM. The model assumes that the ratings represent a latent vector of characteristics. They proposed a constrained RBM suggestion for Netflix shows and merchandise by using individual RBM for every user. There are several RBM methods of instruction, including distinct convergence and simultaneous shaping.[14] Meymandpour *et al*.[15] suggested a tweaked technique they dubbed "Lean Contrastive Divergence" in order to speed up the discovery and predictive process. Using the desired connection as a parameter to RBM and adding the additional details of the films, Tang *et al*.[16] suggested a CF model to attempt to create a hierarchy of the products. In order to propose products to a dynamic user, Zhang *et al.*[17] proposed a hybrid RS-based evolutionary algorithm employing K- Nearest Neighbour (KNN) that takes into account both latent factor and proximity data. According to Yoon *et al.*[18], the process of vectorization benefit from the use of collective models. The challenges of a slow start up, sparse matrices, and difficult recollection of recommendations for products were not addressed in the preceding research. Ma *et al*.[19] performed filtering of products based on cloud services using non-functional requirements. The study did not maintain variations in product categories thus leading to less reliable system. Allahbakhsh *et al*.[20] categorizes social ratings systems based on numeric as well as non-numeric functional requirements by aggregating fuzzy variables. Such approach failed to achieve the lowest root mean square value which leads to less efficient system. Zhu *et al*.[21] introduced graph neural network model for embedding product reviews into an array and produces relevancy scores on the basis of network edges and nodes. Chen *et al*.[22] studied influence of machine learning techniques in enhancing recommendation process. The approach made use of classifiers such as Support Vector Machine (SVM), Naïve Bayes (NB) and Linear Regression (LR) to analyze variations in products. However, the approach achieved lower precision and recall as compared to the proposed approach.

Contributions of the Study

The following is a synopsis of the intended work:

- A user-item evaluation matrix is built from scratch. Singular Value Decomposition (SVD) is merged with the user-item matrix to provide a low-rank estimate of the matrices.
- The second concern step involves developing latent characteristics to anticipate consumer tastes by integrating the matrix of user items into a deep neural network model termed “Restricted Boltzmann Machine (RBM)”. Domain-based ontology development creates a hierarchical representation of these characteristics. Connections among courses, stuff, and occurrences of the user's choices are gathered.
- Last but not least, each prediction grade is unique to the SVD and RBM recommendation methods. A weighted hybridization takes into account both of these factors, allowing the user to choose between two different weight values (W1 and W2) before producing a ranked list of suggestions.

Weighted Parallel Deep Triad Model (Proposed System)

As shown in Fig. 1, this system provides the basis for an interactive filtering method based on balanced concurrent deep triads. By this instance, the SVD and RBM are fed an input consisting of a user-item matrix along with explicit ratings. As input for SVD and RBM approach, user-item aspect data regarding the product category (movies, TV, toys and games), the title of items, minimal cost, highest cost etc. will help ease the cold-start issue related to suggestions. Likewise the SVD and RBM systems provide a prediction rating score, R_{svd} and R_{rbm}, to the final prescription result. The overall weighted mean forecast score R_f for hybridization is calculated by assigning weight values W1 and W2 to each method based on experimental results. The third approach is known as *Ontology-Based Term Weighting Technique (OBTW)* which consists of ontology development, weighting scheme and classifier module. All three approaches- **SVM, RBM and OBTW** are discussed in detail in the following sections.

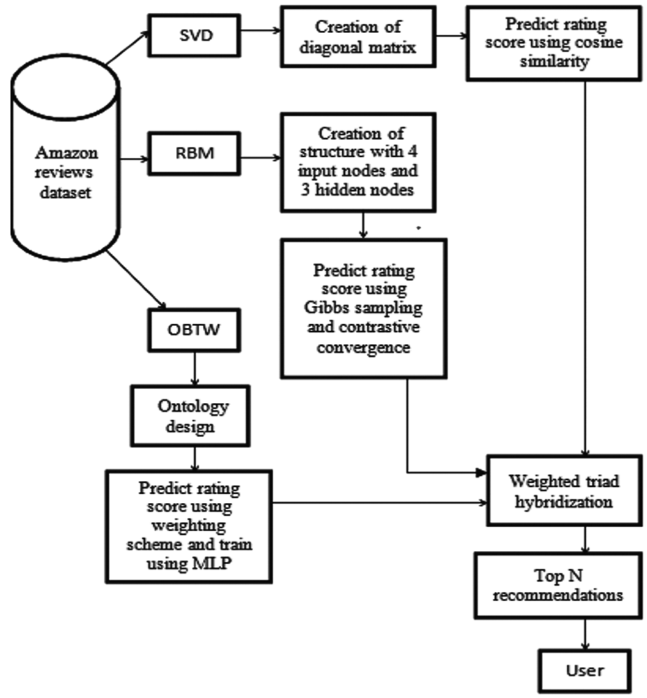

Fig. 1 — Proposed triad hybridization model

Singular Value Decomposition (SVD)

When it comes to decreasing the number of dimensions, the Singular Value Decomposition (SVD) is the gold standard. Generating a low-rank estimate is the main challenge in SVD decomposition. “Given a m $*n$matrix A, with rank r, the singular value decomposition SVD (A) is defined as”

$$SVD(A) = U * S * V \qquad \dots (1)$$

in which U and V have measurements of m by m and n by n, and they are orthogonal matrices. The matrix S has the form (m, n)(m, n), where m and n are both non-negative true values, making it a diagonal matrix. The first r variables of S (s1, s2 ...r) are consecutively positive integers in the range from $s1 \geq s2 \geq s3$ up to $\geq s_r$. The initial r columns of U are the left singular vectors of A and are denoted by the Eigen vectors of AA^T. In a similar way, the first r columns of V are the right singular vectors of A and are the Eigen vectors of A^TA. When trying to get a close approximation of matrix A at low ranks, SVD is the best method. It is generated by eliminating (r - k) columns from U and (r - k) rows from V, while keeping the first k diagonal values, which can be represented as follows:

$$A(k) = U(k) * S(k) * V(k) \qquad \dots (2)$$

The matrix A(k) that was reconstructed is a very close match to the first version matrix A. Matrix A's best-k-rank approximation under Frobenius normalisation looks like this:

$$(A - Ak)(A - Ak) = \sum_{j}^{i} (aij - \sum_{k} Uik * Sk * Vkj) \quad \text{… (3)}$$

Prediction generation using SVD

It is possible to produce a forecast on the m*n ratings matrix 'R' by computing the cosine similarity (dot products) among the m pseudo-customers $U_k.\sqrt{S_k}$ $T$and the n pseudo-productsS_k. $\sqrt{V_k T}$.

For the i^{th} buyer of the j^{th} item, the prediction score R $_{i,\,j}$is calculated through adding each row's average $\overline{ri}$ to the similarity.

$$Ri, j = \overline{\overline{ri}} + Uk\sqrt{Sk(i)} + Sk\sqrt{Vk}T(j) \quad \text{… (4)}$$

Dot product computations take O (1) time once the SVD decomposition is complete since k is a constant in the steps of generating predictions.

Restricted Boltzmann Machines (RBM)

RBM is a two-layer, randomised neural network with both exposed and concealed nodes. It consists of a surface layer consisting of units (users' product preferences), a deep layer of units (the latent components), & a prejudice unit (always active to account for ratings' varying intrinsic popularity).

In addition, each unit that is accessible is linked, in an unsecured fashion, with every unit that conceals itself, and similarly, each unit that is concealed is linked to each unit that is apparent and similarly, each prejudice unit is attached to all units that are both viewable and concealed. To facilitate understanding, the network is constrained such that the layer that is apparent and the layer that is concealed do not communicate with one another. In this scenario, each hiding node multiplies each input x by its associated weight w. This means that there are a total of 12 weights in this network (4 input nodes times 3 hidden nodes), with each input x being assigned a value between 0 and 3. The weights among both levels always take the shape of a matrix, with input nodes in the rows and nodes for output in the columns. The four inputs have been multiplied by the appropriate weights and sent to each hiding node. In order to generate only one result for every single hidden node, the activation process is fed the total of those goods, together with a bias (which ensures that some activations will occur). The structure depicting RBM nodes is shown in Fig. 2.

Each observable unit's status has been identified since the learning data encodes user settings on the objects using stochastic, binary visible units. Additionally, hidden units capture latent properties as stochastic, binary values. Using this power functioning, the network gives each potential pairing of hidden and visible vectors a certain chance. The formula for the potential energy at a given velocity and height is:

$$P(v,h) = \exp(v,h) \div z \quad \text{… (5)}$$

where, E is the system's energy, v is visible layer, h is hidden layer and Z is a normalising factor, respectively.

Prediction generation and correlation with RBM structure

Assume there are P items (such as a film, video game, or food products), N consumers, and integer ratings (from 1 to 5). When using RBM to product ratings, the first challenge is figuring out how to effectively deal with the massive amount of missing ratings. The suggested workaround is to pretend those goods' visible units the user didn't rate don't exist. In actual use, these exposed components are always disabled, making their status zero. This may also be seen as each user having their own RBM, with everyone utilising the same set of units but incorporating just the user's own Softmax values of the goods they evaluated.

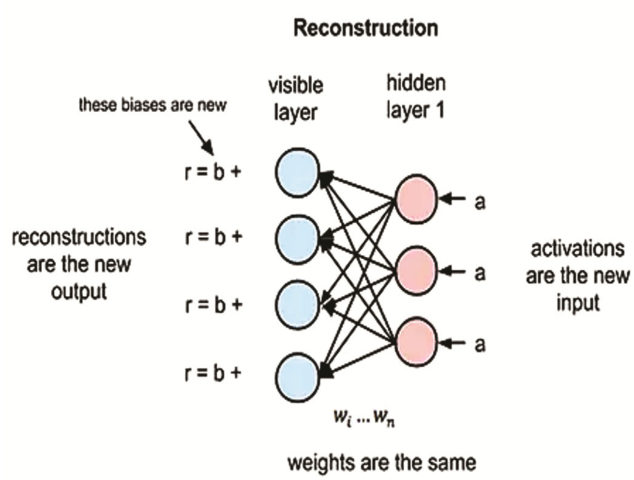

Fig. 2 — General framework of RBM with 4 input nodes and 3 hidden nodes[23]

Softmax units have been substituted for the binary visible units in Fig. 3. The RBM only contains Softmax units for goods that were rated by the respective user. Let's take a look at a few u rated p items. For simplicity, let's define visible unit V to be a K * p matrix where v_i^k = 1 if u rated product i as k and 0 otherwise.

Training convergence behaviour analysis of RBM

Let's call the binary equivalents of the user characteristics' hidden units h (j = 1 to f). Eqs (6) and (7) model the multinomial (a Softmax) used to model the columns of V.

Learning an RBM with Gibbs sampling and a Contrastive convergence step is represented below using Eqs (6) and (7):

$$P(vi = 1/h) = \exp(bi + \sum_{j=1} hjWi,j) / \sum_{i=1} \exp(bi + \sum_{j=1} hjWi,j) \quad \ldots (6)$$

$$P(hj = 1/h) = 1/1 + \exp(-bj - \sum i = 1 \sum_{k=1} viWi,j) \quad \ldots (7)$$

where, W_{ij} is how much weight is placed on the association among product i's rating k and the concealed unit j, bik is the bias of product i's rating k, and bjk is the term describing the bias of the masked unit j.

Equation (8) shows that the energy component is denoted by E (V, h).

$$E(V,h) = -\sum_{a=1}^{u} i \sum_{b=1}^{v} i \sum_{c=1}^{w} iWijhjVk - \sum_{a=1}^{u} i \sum_{b=1}^{v} iVibik - \sum_{a=1}^{u} hjbjk \quad \ldots (8)$$

Ontology-based Term Weighting Technique (OBTW)

The process comprises three parts: the creation of a dataset-related ontology, a weighting system, and a classifier module.

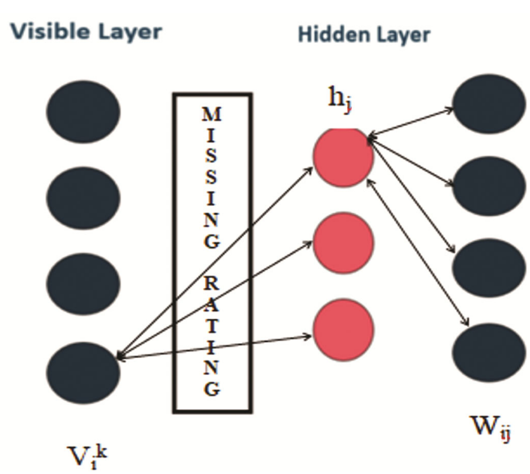

Fig. 3 — Using RBM (4 * 3), we can see how ratings of the product link to both the surface and the depths

Ontology Development

Comprehension of an abstract representation of a specific domain, such as ontology, is referred to as "conceptualization".[23,24] Information interchange along with knowledge retrieval on the internet would not be possible without ontology, which supplies the essential semantic in a diverse setting. The relationships within each entity in a space are represented by an ontology, which is a set of classes (C1, C2, ….. C_n), attributes (P1, P2, ….. P_n), and instances (I1, I2, …… I_n). A well-organized body of domain-specific knowledge may be derived from a logical description of interpersonal connections between persons.

Weighting Scheme

A terminating list of phrases eliminates filler words including articles, prepositions, conjunctions, pronouns, and verb forms. At first, we remove stop words. The next step is stemming, which gets rid of the endings of words. Each token is given a certain value after it has been tokenized. The frequency with which a word appears in the domain ontology is crucial to the suggested ontology-based term weighting strategy.

Algorithm 1: Weighting scheme

Input: Designed ontology (Fig. 4) related to domain $C_{onto}=\{C_1, C2….. C_o\}$

Parse and pre-process the designed ontology to obtain subset of terms $T_w = \{t_1, t_2 ……t_i\}$

For every term t_i in T_w**do**
For every domain C_o in C_{onto}**do**
If the term t_i exists in C_{onto} then
Add t_i to the new OntoSet
End if
For every term t_i € OntoSet**do**
Calculate semantic weight S_w of term t_i as,
$S_w = [\sum t_i * \log n / C]$ where n denotes number of instances and C denotes number of classes in OntoSet
End for
End for
End for

Output: A set of terms mapped to the domain ontology with associated weights as,
$M_t = [(t_1, w_1), (t_2, w_2)…… (t_k, w_k)]$"

Classifier model

It involves the following steps:

- Feature extraction from the designed ontology (named entities detection and filtering and semantic matching using Word Net)

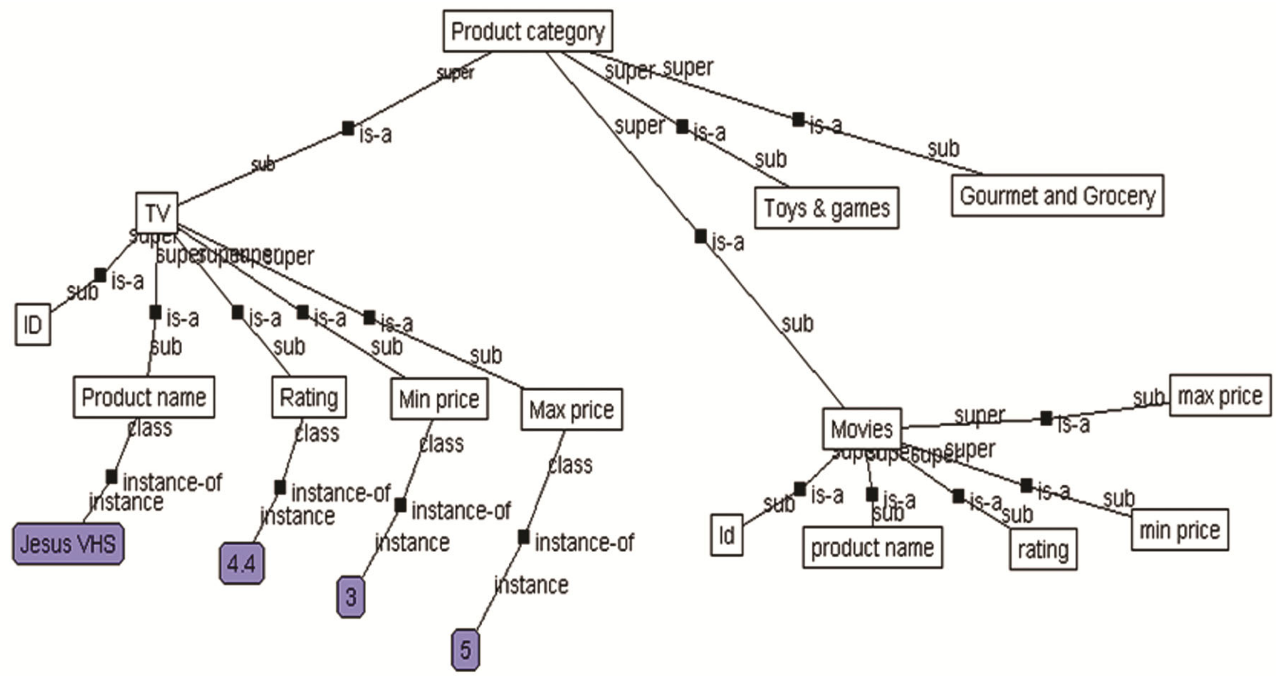

Fig. 4 — Designed ontology related to products description

- Train input data or extracted features using MLP classifier
- Perform testing of data using 5 point cross validation
- Estimate the classifier performance using Precision, Recall and F-1 measure

In conclusion, the weighted triad model provided here yields reliable results for individualising recommendations. Integrating the forecasting scores from SVM, RBM, and OBTW methods helps with the cold-start issue as well. Overall, the **complexity of the proposed model is O (t-w)** where t signifies terms associated with weights (w).

Operating Parameters

Windows 10, 64 bit operating system

Implemented in python IDE Shell version 3.11 with Tensorflow 3.0 as backend

Dataset and Evaluation Metrics

To perform evaluation, a dataset is used named Amazon Product Reviews Dataset.[25] It comprises product names, category and ratings given based on the user preferences, product identifiers and price range.

The following metrics are used in the context of our study:

$$Precision = TP/(TP+FP) \quad \ldots (9)$$

$$Recall = TP/(TP+FN) \quad \ldots (10)$$

F1-Score =

$$2(Precision * Recall)/(Precision + Recall) \quad \ldots (11)$$

where TP denotes true positives, FP means false positives and FN means false negatives.

Root Mean Squared Error (RMSE)=

$$\sqrt{1/N\sum_{u,p}(Ru-Rp)(Ru-Rp)} \quad \ldots (12)$$

Mean Absolute Error (MAE) =

$$1/N\sum_{u,p}(Ru-Rp) \quad \ldots (13)$$

Normalized Discounted Cumulative Gain (NDCG): It is computed by dividing DCG and DCG is given as:

$$DCG = \sum rel(r)/\log(r+1) \quad \ldots (14)$$

$$\text{NDCG} = 1/DCG \quad \ldots (15)$$

where, N is number of samples, R_u denotes rating by user and R_p means predicted or estimated rating. rel (r) signifies degree of relevance and r denotes ranked documents. A smaller value of RMSE and MAE indicates better performance of the method. NDCG is used to assess the effectiveness of ranking system based on the number of relevant results retrieved. Higher NDCG signifies higher retrieval effectiveness and accuracy of the system.

Results and Discussions

The parameters set to train features from the designed ontology (Fig. 4) include setting of MLP classifier with *validation set proportion 0.20 and learning rate 0.15*. After applying weighting scheme algorithm and testing of data, the following results (Fig. 5) are produced that verify the correctness of our proposed approach.

Validation of the proposed work with literature studies

In Table 1, the implemented results of the proposed model are compared with existing traditional collaborative filtering methods individually as well as recent studies for Amazon Product Reviews dataset.

The values of RMSE and MAE for the proposed method are 0.9443 and 0.6589 respectively. They are lesser than SVD (0.9654, 0.7754), RBM (1.2134,

Weigths

From INPUT to HIDDEN layer

-	Neuron "1"	Neuron "2"	Neuron "3"	Neuron "4"	Neuron "5"	Neuron "6"	Neuron "7"	Neuron "8"	Neuron "9"	Neuron "10"
Ratings	0.16473250	0.16397783	0.16367125	0.16529366	0.16475683	0.16397711	0.16530961	0.16502024	0.16444268	0.16499249
Min. Price	0.20365736	0.20342949	0.20288776	0.20403941	0.20344243	0.20356092	0.20455019	0.20453701	0.20355698	0.20404846
Max. Price	0.33069735	0.33059607	0.32978276	0.33200364	0.33180684	0.33093075	0.33182981	0.33217950	0.33059693	0.33177315
bias	0.00426923	0.00417339	0.00443205	0.00471349	0.00442679	0.00463863	0.00455458	0.00408645	0.00408177	0.00458884

From HIDDEN to OUTPUT layer

-	Movies and TV	Toys and Games
Neuron "1"	-0.22560942	0.22572879
Neuron "2"	-0.22495699	0.22567086
Neuron "3"	-0.22499093	0.22404678
Neuron "4"	-0.22668398	0.22691579
Neuron "5"	-0.22624993	0.22632878
Neuron "6"	-0.22544691	0.22537896
Neuron "7"	-0.22643033	0.22730574
Neuron "8"	-0.22692897	0.22713443
Neuron "9"	-0.22534425	0.22554590
Neuron "10"	-0.22671205	0.22651549
bias	0.09725174	-0.09788908

Fig. 5 — Testing of data using MLP as a classifier

Table 1 — Comparative analysis of the proposed approach with existing filtering methods and recent studies

Studies / Approaches	RMSE	MAE	Precision (%)	Recall (%)	F1-score (%)	NDCG
Shetty *et al.*[8]	1.0023	0.7911	76	77	77	0.63
Shen *et al.*[13]	1.2087	0.8276	73	74	74	0.69
Meymandpour *et al.*[15]	1.2311	0.8090	74	75	75	0.87
Tang *et al.*[16]	1.056	0.7478	78	79	78	0.82
Zhang *et al.*[17]	1.1187	0.8611	79	80	80	0.76
Yoon *et al.*[18]	1.1355	0.8910	83	82	81	0.67
Ma *et al.*[19]	1.1781	0.7898	82	83	83	0.23
Allahbakhsh *et al.*[20]	1.1232	1.0121	80	81	80	0.22
Zhu *et al.*[21]	1.4315	1.0341	77	78	79	0.48
Chen *et al.*[22]	1.2133	1.0563	78	79	78	0.38
SVD	0.9654	0.7754	72	77	74	0.45
RBM	1.2134	1.0203	68	69	69	0.43
PMF	1.9643	0.6987	79	76	77	0.39
Proposed	0.9443 (lowest)	0.6589 (lowest)	93 (highest)	95 (highest)	93 (highest)	0.94 (highest)

1.0203) and PMF (1.9643, 0.6987) methods which indicate the better performance of our triad approach. The values of Precision, recall and F1-score for the proposed method are 93%, 95% and 93% respectively. They are higher than SVD (72%, 77% and 74%), RBM (68%, 69%, and 69%) and PMF (79%, 76%, and 77%) methods which clearly indicate the better performance of our triad approach. Also, it is observed that the proposed model achieves the highest NDCG score (0.94) thus outperforming recent studies and existing filtering methods. Highest NDCG signifies that the proposed model retrieves the highest relevant results thus enhancing the system's efficiency.

Conclusions

On basis of the implemented results, we can clearly conclude that the proposed triad approach has better accuracy as compared to existing collaborative filtering methods including SVD, RBM and PMF. This is credited to the integration of three techniques namely SVD, RBM and Ontology based term weighting which helps in mitigating issues of traditional recommender systems such as cold start and sparsity problem. Thus, the triad hybridization of these methods provides higher performance in terms of metrics like RMSE, MAE, Precision, Recall and F-1 score. *The RMSE, MAE, Precision, Recall, F1 score* and *NDCG* values for the proposed system are 0.9443, 0.6589, 93%, 95%, 93% and 0.94 respectively. As a future aspect, this work can be extended to building a prototype for personalized search engine with the concept of ontology and multi agent systems. Ontology can be created automatically using knowledge graph embeddings which would reduce the need of manual ontology editor. To ensure dynamic variations and adaptive weighting scheme for each component, incremental learning techniques such as Bayesian distribution can be integrated. Also, learning rates can be selected based on dynamic weighting scheme with dynamic nodes embedded in knowledge graphs.

Conflicts of Interest (COI)

The authors declare that they have no conflicts of interest to report regarding the present study.

References

1 Agarwal V & Vijayalakshmi A, Recommender system for surplus stock clearance, *Int J Electr Comput Eng*, **9** (2019) 3813–3821, DOI: 10.11591/ijece.v9i5. pp 3813-3821.

2 Rahman N S A, Handayani L, Othman M S, Al-Rahmi W L, Kasim S & Sutikno T, Social media for collaborative learning, *Int J Electr Comput Eng*, **10** (2020) 1070–1078, DOI: 10.11591/ijece.v10i1.pp1070-1078.

3 Babeetha S, Muruganantham B, Ganesh S & Murugan A, An enhanced kernel weighted collaborative recommended system to alleviate sparsity, *Int J Electr Comput Eng*, **10** (2020) 447–454, DOI: 10.11591/ijece.v10i1. pp447-454.

4 Jalili M, Ahmadian S, Izadi M, Moradi P & Salehi M, Evaluating collaborative filtering recommender algorithms: A Survey, *IEEE Access*, **6** (2018) 74003–74024, doi: 10.1109/ACCESS.2018.2883742.

5 Behera D K, Das M & Swetanisha S, A research on collaborative filtering based movie recommendations: from neighborhood to deep learning based system, *Int J Recent Technol Eng*, **8** (2019) 10809–10814, doi: 10.35940/ ijrte.D4362.118419.

6 Meqdad M N, Al-Akam R & Kadry S, New prediction method for data spreading in social networks based on machine learning algorithm, *TELKOMNIKA*, **18** (2020) 3331–3338, DOI: 10.12928/telkomnika.v18i6.16300.

7 Rustam Z & Saragih G S, Prediction schizophrenia using random forest, *TELKOMNIKA*, **18** (2020) 1433–1438, DOI: 10.12928/telkomnika.v18i3.14837.

8 Shetty N P, Narula R & Tandona K, Comparison study of machine learning classifiers to detect anomalies, *Int J Electr Comput Eng*, **10** (2020) 5445–5452, DOI: 10.11591/ ijece.v10i5.pp5445-5452.

9 Vlachos M, Dunner C, Heckel R, Vassiliadis V G, Parnell T & Atasu K, Addressing interpretability and cold-start in matrix factorization for recommender systems, *IEEE Trans Know Data Eng*, **31** (2019) 1253–1266, doi: 10.1109/TKDE.2018.2829521.

10 Saga R & Duan Y, Apparel goods recommender system based on image shape features extracted by a CNN, *Proc Int Conf on Systems, Man, and Cybernetics (IEEESMC),* 2018, 2365–2369, doi: 10.1109/SMC.2018.00406.

11 Nihad E G, Mokhtar E N, Abdelhamid Z & Mohammed A, Hybrid approach of the fuzzy C-means and the K-nearest neighbors methods during the retrieve phase of dynamic case based reasoning for personalized follow-up of learners in real time, *Int J Electr Comput Eng*, **9** (2019) 4939–4950, DOI: 10.11591/ijece.v9i6.pp4939-4950.

12 Alzaqebah M, Alrefai N, Ahmed E A, Jawarneh S & Alsmadi M K, Neighborhood search methods with moth optimization algorithm as a wrapper method for feature selection problems, *Int J Electr Comput Eng*, **10** (2020) 3672–3684, DOI: 10.11591/ijece.v10i4.pp3672-3684.

13 Shen W, Wang J & Han J, Entity linking with a knowledge base: issues, techniques and solutions, *IEEE Trans Knowl Data Eng*, **27** (2015) 443–460, doi: 10.1109/TKDE.2014.2327028.

14 López E V, Luis M, Fernández M & Huete J F, Use of textual and conceptual profiles for personalized retrieval of political documents, *Knowl-Based Syst*, **112** (2016) 127–141, https://doi.org/10.1016/j.knosys.2016.09.005.

15 Meymandpour R & Davis J, A semantic similarity measure for linked data: An information content-based approach, *Knowl-Based Syst*, **109** (2016) 276–293, https://doi.org/ 10.1016/j.knosys.2016.07.012.

16 Tang B, Kay S & He H, Towards optimal feature selection in naive bayes for text categorization, *IEEE Trans Knowl Data Eng*, **28** (2016) 2508–2521, doi: 10.1109/TKDE.2016.2563436.

17 Zhang S, Yao L, Sun A & Tay Y, Deep learning based recommender system: A survey and new perspectives, *ACM Comput Surv*, **52** (2018) 1–38, https://doi.org/10.1145/3285029.

18 Yoon J H, & Jang B, Evolution of deep learning-based sequential recommender systems: from current trends to new perspectives, *IEEE Access*, **11** (2023) 54265–54279, doi: 10.1109/ACCESS.2023.3281981.

19 Ma H, Hu Z, Li K & Zhu H, Variation aware cloud service selection via collaborative QoS prediction, *IEEE Trans Serv Comput*, **14** (2021) 1954–1969, doi: 10.1109/TSC.2019.2895784.

20 Allahbakhsh M, Amintoosi H, Ignjatovic A &Bertino E, A trust-based experience aware framework for integrating fuzzy recommendations, *IEEE Trans Serv Comput*, **15** (2022) 698–709, doi: 10.1109/TSC.2019.2956937.

21 Zhu T, Sun L & Chen G, Embedding disentanglement in graph convolutional networks for recommendation, *IEEE Trans Knowl Data Eng*, **35** (2023) 431–442, doi: 10.1109/TKDE.2021.3087791.

22 Chen Yi & Huang J, Effective content recommendation in new media: leveraging algorithmic approaches, *IEEE Access*, **12** (2024) 90561–90570, doi: 10.1109/ ACCESS.2024.3421566.

23 Yadav U, Narula G S, Duhan N & Murthy B K, An overview of social semantic web framework, *Proc Int Conf Comput Sustain Global Dev (IEEE)*, 2016, 769–773.

24 Yadav U, Narula G S, Duhan N & Jain V, A novel approach for precise search results retrieval based on semantic web technologies, *Proc Int Conf on Computing for Sustainable Global Development (IEEE)*, 2016, 1357–1362

25 https://www.kaggle.com/datasets/yasserh/amazon-product-reviews-dataset (13 December 2025).