Int. J. Nonlinear Anal. Appl. 15 (2024) 7, 197–214
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2023.30855.4511

A deep neural network-based approach in tag recommender system to overcome users' Cold Start

Mahdi Bazargani[a], Sasan H. Alizadeh[b,*]

[a]Department of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

[b]Faculty of Information Technology, ICT Research Institute (Iran Telecommunication Research Center), Tehran, Iran

(Communicated by Seyyed Mohammad Reza Hashemi)

Abstract

Recommender systems are used in various fields such as movies, music, and social networks. Recommender systems aim to provide attractive offers to users according to their performance in the system. The most popular recommender systems are content-based models and collaborative filtering methods. One of the most important challenges and problems in recommender systems is the challenge of users' cold start. So far, various methods such as machine learning algorithms, optimization approaches, and statistical methods, have been proposed by other researchers in improving internet marketing strategy and overcoming the cold-start problem, which despite having numerous applications, still could not solve the start problem. This article will investigate the problem of cold start users' by presenting a recommendation model based on a deep neural network and considering the problem of improving the internet (network) marketing strategy. In this article, the relevant simulation is done on the popular Movielens dataset, which is from 2015, and the evaluations of the methods presented on this dataset are compared

Keywords: Cold Start, Deep Neural Network, Recommender system
2020 MSC: 68T07

1 Introduction

Tagging has emerged as one of the most efficient methods of associating metadata with objects (e.g., videos, texts) in Web 2.0 applications. In contrast to arbitrary keywords assigned to objects by users, tags represent a simpler, less expensive, and more natural way to organize content. Several previous studies have proposed tag recommenders to help users efficiently explore and organize web resources. However, most of these methods require a sufficient amount of user-tagged data, which may In addition, recent studies have shown that, among other textual features such as titles, user comments, and descriptions, tags are more effective in supporting information retrieval services, such as search [12], automatic classification [6], and content recommendation [9]. The tagging process can use a tag suggestion service. These types of services allow users to select some recommended tags or use new ones. Besides improving the user experience, the tag suggestion will increase the quality of the generated tags and finally improve the quality of word-of-mouth retrieval services, where tags are considered as data sources. Moreover, the obvious benefits of improving the description of objects, helping services such as browsing, direct use of recommended tags in search [10],

*Corresponding authors
Email addresses: mbzir@iauz.ac.ir (Mahdi Bazargani), s.alizadeh@itrc.ac.ir (Sasan H. Alizadeh)

Received: March 2023 *Accepted:* July 2023

and expanding queries [18] are also very popular cases of using tags. Despite item recommendation, which usually aims to match the target user's interests, tag suggestion applies to describing, summarizing, and organizing the content of objects, and matching the user's interests to personalized suggestions. Therefore, the design of recommenders is challenging and requires special solutions whose methods vary item recommendation methods. For example, text retrieval, knowledge retrieval, and cognitive meanings play an essential role in determining the tag. In addition, the effectiveness of the recommendation is very important, considering that poor recommendations, in addition to reducing or eliminating user satisfaction, ultimately affect the performance of various IR services that use tags as the data source they use, which will cause irreparable damage. Many novel methods for label recommendation are designed with the assumption that an initial set of labels exists on the target object. However, the effectiveness of these methods in a cold start scenario where there is no initial tag (even with the possible presence of other attributes of the target object such as title and description) becomes a bit difficult. To solve this problem, previous works extract the candidate expressions for tagging directly from the text of the target object or similar/related objects and from the statistical properties of word occurrence such as term frequency and inverse document frequency to rank the tags for the suggestion. However, these features, individually, may not be good enough to effectively rank candidates for tags, especially when using small and possibly low-quality regular texts associated with objects. In this work, different syntactic patterns (e.g., syntactic dependencies between words in a sentence) are analyzed from text related to objects that are used to find and suggest tags. In addition, two types of input content (containing text related to objects) and preferences (preferring a tag for an item) are used to train deep neural networks to develop a robust hybrid model for suggesting tags for items with an overriding approach. The cold start is provided. One of the strategic areas is the internet marketing area, which has been investigated a lot in different societies recently. In internet marketing, different and diverse products are offered to the internet market. These products can be films, clothes, cosmetics, electronic components, etc [17]. In the category of internet marketing, to present a product in the internet market, it must be easily retrieved by search engines. The more easily the products are retrieved in these search engines, the easier they can be offered to the applicants of this product and capture a higher share of this internet market. Product tags are the most essential part of a product used by search engines. Tags or labels express the nature of the product and the important characteristics of the product. In addition, tags are indexed by search engines and are known as product descriptors. In addition, the more suitable tags are intended for products, the better they can be recognized by search engines. And get more share of internet marketing. In this article, a tag proposal is presented to improve the internet retrieval of products. In this article, in part 2, the works done in the past are examined, in part 3, the proposed method and relevant details are explained, finally, in part 4, the results are obtained, and in part 5, the conclusion and future suggestions are examined.

2 Related Works

In the Table 1, a summary of previous research records is presented. According to the review of the previous methods and observing the results obtained and the performance of each of the previous methods, it was observed that these methods have challenges and problems such as insufficient accuracy, high error, and the time-consuming process of information processing. Hence, in this research, a tag proposal model based on a deep neural network is presented to overcome the cold start problem and solve the challenges of other methods.

To assess the accuracy of recommendation systems, various measures have been developed and discussed in literature. The most commonly used measures for evaluating the accuracy of a recommendation method are Mean Average Error (MAE), Root Mean Squared Error (RMSE), Coverage, Precision, F1-measure, and Recall, as listed in Table 2. While there are other measures available for evaluating system accuracy, they are not as widely used or discussed in detail by researchers. MAE and RMSE are two measures that calculate the difference between the actual rating of an item and the predicted rating made by a recommendation method. They do this by finding the average distances between them.

3 Architecture of the proposed model

By applying evolutionary algorithms to optimize artificial deep neural networks (DNN), we can improve their predictive capabilities in various applications [1, 5, 8]. In this article, a recommender system based on deep neural networks, or DNN, is presented. The general flowchart of the proposed method is given in Figure 1, which is explained in the following sections.The purpose of this article is to provide a tag recommender system that recommends tags for different items based on the information in the access, on the other hand, these tags can cause more sales and improve the internet marketing. strategy because in identifying the item to customers, information retrieval systems and search

Table 1: Features of papers that use in Recommender System to alleviate cold-start problem.

Ref.	Major Improvement	Techniques	Dataset	Abstract Summary
[17]	Precision	MFS-LDA	Wikipedia	presents MFS-LDA, a novel multi-feature space tag recommendation model that addresses the cold-start problem in tag-based recommendation systems
[20]	Validation AUC	DNN	Social (XING's)	A combination of content-and neighbor-based models won both offline and online phases.
[4]	Performance	NSPR(DNN Family)	CiteUlike & Yahoo Movies	A probabilistic modeling approach called Neural Semantic Personalized Ranking significantly outperforms the state-of-the-art baselines.
[7]	Accuracy	Personalized Preference on Non-functional Attributes	QWS	The addition of users' personalized preferences on non-functional attributes alleviates these limitations and improves recommendation accuracy.
[21]	Precision	a novel contextual multiarmed bandit model BiUCB	MovieLense	A novel bandit algorithm called binary upper confidence bound can deal with the item-user-cold-start problem where there is no information about users and items.
[19]	Performance	Fuzzy Linguistic	No Standard dataset is used	Remove Cold-start Problem from REFORE
[22]	Accuracy	DNN	Netflix	Two recommendation models were proposed for cold start items.
[24]	Quality	Matrix factorization	WHUT Digital Communication Engineering Co	A tag-based interactive framework to make the resource recommendation for different users.
[26]	Accuracy & Quality	DUTS	CiteUlike	The proposed method is better than the compared methods at the accuracy of tag recommendation.
[23]	Accuracy	SIMWORD	SO & Math	The proposed method outperforms several existing methods in terms of recommendation accuracy.
[2]	Precision	NLP & L2R	Bibsonomy, Movielens LastFM, Elo7	improving tag recommendation in a cold start scenario by analyzing the syntactic patterns of the text associated with Web 2.0 objects
[16]	Quality	co-SVD model & Matrix Factorization	Movielens	proposes a personalized recommendation approach that utilizes matrix co-factorization techniques to incorporate user-generated tags and temporal dynamics into the prediction model
[11]	Performance	attention-based Capsule Network(ACN)	TPA & AG	Proposed the architecture of the attention-based Capsule Network(ACN)

Table 2: Evaluation measures used in the main papers.

Ref.	Evaluation Measure					
	MAE	RMSE	Precision	Precision	Recall	F1-Measure	Other
[17]			✓				
[20]							✓
[4]					✓		✓
[7]	✓						
[21]			✓				✓
[24]		✓	✓		✓		
[26]							✓
[23]					✓		
[16]	✓	✓	✓		✓	✓	
[25]							✓
[11]			✓		✓	✓	

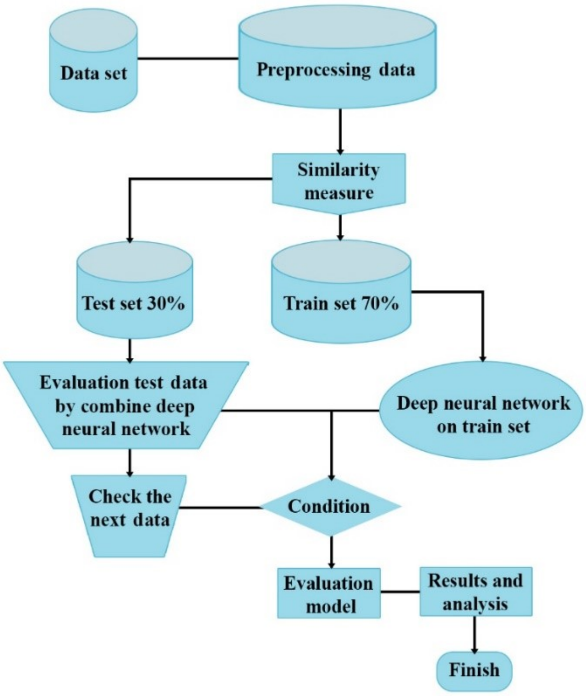


Figure 1: Flowchart of the proposed method.

engines have had a very successful performance. Also, to implement this system, first, the data set is entered into the proposer's system. In the pre-processing stage, first, non-numeric variables such as protocol type have been converted to numeric. To perform this conversion, there are two binary and histogram-based methods, and because there are flaws in the binary method, the Target-based method was used. This method converts any non-numeric variable into a number, which is the probability of repeating the same histogram. On the other hand, the data should be normalized at this stage and should be in the zero range so that the suggestions have higher accuracy. After this step, the demographic information of the contacts is used to calculate the similarity between the contacts. Then, to test and train random forest and deep neural network algorithms, separate training data samples are considered. Training samples are used to train a deep neural network and random forest algorithms, and test samples are used to evaluate the model. In the end, models of deep neural networks and random forest algorithms, based on training and test samples, are created, and these models provide labels for new inputs.

4 Description of the proposed method

4.1 Used datasets

In this article, to analyze and evaluate the outputs, simulation has been done with the MovieLens dataset. The output is analyzed, compared, and evaluated with other methods performed on this data set. From the MovieLens dataset, the data of 2013, which is about 100 MB, has been considered. In these files, the number of 1,000,209 records are related to the ratings registered by users to the movies, the number of 3900 movie samples and the number of 6040 user samples, each user has voted for at least 20 movies.

4.2 Data preprocessing

The pre-processing stage is such that non-numerical variables such as the type of protocol are converted into numerical variables and since there are two binary and histogram methods, the histogram method is used because of its advantage over the binary one, and actually this number is also considered as the probability of repeating the corresponding histogram. After concatenating the data, normalization of the data is done so that the data is in the range of zero and one, the range of the data must be in a suitable range because the characteristics that have a higher value have an impact and this makes suggestions not be presented with proper precision. Also, normalization makes better training in neural networks because the training of large weights is not done well in neural networks. Placing the data in the range of zero and one, is one of the normalization methods.

$$X_{NO} = 2 \times (X - X_{\min})/(X_{\max} - X_{\min}) - 1 \tag{4.1}$$

4.3 Measuring user similarity

If two audiences, u and v rate two items i and j and the values of this rating are r_{ui} and r_{vj}. Based on the ranks (r_{ui}, r_{vj}), the similarity formula of two users u and v is defined as follows:

$$\mathrm{Sim}(u, v) = \mathrm{Sim}_1(r_{ui}, r_{vj}) \tag{4.2}$$

On the other hand, the similarity of i and j should be taken into account when calculating the similarity of the user, because usually, the items are different. This similarity of the item $Sim_{item}(i, j)$ is used to increase the accuracy of the user's similarity and weight is considered. So the following relationship is defined:

$$\mathrm{Sim}(u, v) = \mathrm{Sim}_{\mathrm{item}}(i, j) \cdot \mathrm{Sim}_1(r_{ui}, r_{vj}) \tag{4.3}$$

To calculate the similarity between two users u and v who rate more than one object, all pairs of ratings given by two users u and v should be considered. As a result, the sum of all possible pairs of items for user similarity is calculated:

$$\mathrm{Sim}(u, v) = \sum_{i \in J} \sum_{j \in J} \mathrm{Sim}_{\mathrm{item}}(i, j) \cdot \mathrm{Sim}_1(r_{ui}, r_{vj}) \tag{4.4}$$

4.4 shows that even if the items do not have a rank, this Sim calculation is done and in fact, it is a measure of user similarity and is a solution for situations where the items have the same rating.

To consider the difference between users, the asymmetric factor $\mathrm{Sim}_2(u, v)$ and to calculate the priority of the model, another weighting factor $\mathrm{Sim}_3(u, v)$ is used in the similarity calculation model. The final formula for calculating user similarity is given below:

$$\mathrm{Sim}(u, v) = \mathrm{Sim}_2(u, v) \cdot \mathrm{Sim}_3(u, v) \cdot \sum_{i \in Iu} \sum_{j \in Iv} \mathrm{Sim}_{item}(i, j) \cdot \mathrm{Sim}_1(r_{ui}, r_{vj}) \tag{4.5}$$

The Sim_1 function is expressed based on the PPC model to provide a better fit for the non-linear relationship between users. The expanded PPC is defined as a Simi function, which is as follows:

$$\mathrm{Sim}_1(r_{ui}, r_{vj}) = \mathrm{proximity}(r_{ui}, r_{vj}).\mathrm{significance}(r_{ui}, r_{vj}).\mathrm{singularity}(r_{ui}, r_{vj}) \tag{4.6}$$

Where the proximity function gives a similarity value considering the absolute difference between r_{ui} and r_{vj}. The importance function gives the effectiveness of pair ranking evaluation in the final result, in such a way that the ranking of pairs with a greater distance from the mean is more important. This pair of points is more important. The unique function expresses the difference in the score of a pair compared to other scores.

$$\mathrm{proximity}(r_{ui}, r_{vj}) = 1 - 1/(1 + \exp(-|r_{ui} - r_{vj}|)) \tag{4.7}$$

$$\mathrm{significance}(r_{ui}, r_{vj}) = 1/(1 + \exp(-|r_{ui} - r_{med}| - |r_{ui} - r_{med}|)) \tag{4.8}$$

$$\mathrm{singularity}(r_{ui}, r_{vj}) = 1 - 1/(1 + \exp(-|(r_{ui} - r_{vj})/2 + (\mu_i - \mu_j)/2|)) \tag{4.9}$$

On the other hand, μ_i and μ_j are the averages of item i and item j. The meaning that the function $Sim_1(r_{ui}, r_{vi})$ returns cannot be used separately, because the user u and v show the rate of two different items i and j, but if i and j are similar, Sim_1 returns a large value in the sense that u and $v$are similar, otherwise, the large value returned by Sim_1 indicates that u and v are not similar, so it should be $\mathrm{Sim}_{item}(i, j) \cdot \mathrm{Sim}_1(r_{ui}, r_{vj})$ should be used to calculate the similarity of two users. However, this function plays an important role in calculating the similarity of users.

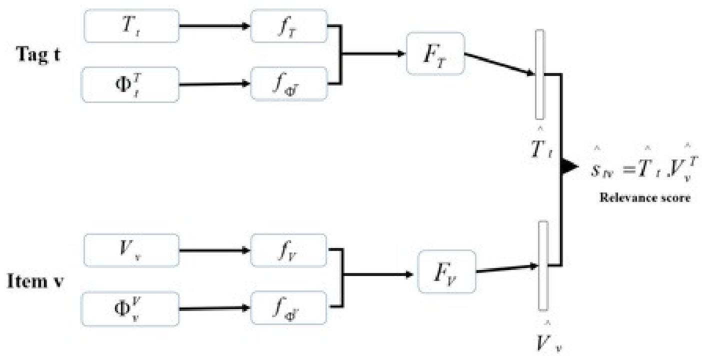


Figure 2: Changed Dropout method architecture for tag recommendation.

4.4 Using hybrid deep neural network

This part describes the modified dropout method and also deals with the integration of deep neural networks and random forest machine learning algorithm. In the basic CF problem, there is a set of N users $U = \{u_1, ..., u_N\}$ and a set of M items $V = \{v_1, ..., v_M\}$ that a user is recommended a set of items. Here, the goal is to suggest a tag. Instead of a set of N users, a set of$N$ tags $T = \{t_1, ..., t_N\}$ is considered.

R_{tv} it means the preference of item v by label t. A matrix $N \times M$ called R is considered, which represents the feedback of labels for items. The value of relevance in the direct situation is numerical, but in the case of indirect feedback, R is binary. In situations where information about preferences is not available answer is$R_{tv} = 0$. It also provides $T(t) = \{t \in T \mid R_{tv} \neq 0\}$ the labels that are considered for item v. On the other hand, it presents $V(v) = \{v \in V \mid R_{tv} \neq 0\}$ the items that are considered for the t tag. When the preferred information is not available, especially in the cold start, the values that are considered are $V(v) = 0, T(t) = 0$.

In many situations, information is available for tags and items, for which different types of data can be considered, such as text, audio, video, image, etc. The information of items that have these tags in the past can be considered as tags. This data can be an effective guide for the recommender model, especially in situations where there is no preferred information and a cold start scenario would have existed.

The content information can be represented by a fixed-length feature vector, after related transformations, it showed thatΦ^t and Φ^v are the content features for tags and items, where $\Phi^t(\Phi^v)$ is the content feature vector for the tag t (item v). Also, if the content information exists but does not have it, the feature vector becomes zero. Finally, the preference information of R and the content information Φ^t and Φ^v are used to train the recommender model with high accuracy.

In the dropout model, for the model to be able to deal with hot and cold starting situations, preference and content information is used as input, so to propose a label, the input of the model is checked first. There are two methods for entering preferred information (R). The first method is to enter the rows and columns of R as input, which is not an effective method because the more items and tags, the more rows and columns of the matrix. Another method is in latent models, the approximate preferences matrix is the product of two low-rank matrices T and V. This method is called latent representation.

$$R_{tv} \approx T_t V_v^T \tag{4.10}$$

T_t and V_v are the latent representations of tag t and item v. U and V are matrices with low dimensions with rank $D << min(N, M)$, this low rank reduces the complexity of the DNN model because the activation size in the first latent layer depends on the size of the input, so the inputs $[T_t, \Phi_t^T]$ and $[V_v, \Phi_v^V]$ are considered for each tag t and item v Also, to convert the input into a hidden display, the DNN model is used, which uses the preference and content information, in the first stage T_t, the preferences and content Φ_t^T pass through the DNNs f_T and f_{Φ^T} and the output of the DNNs are combined and entered to F_t. Finally, the output is F_t a hidden display of T_t.

The above steps are also done for items, so f_V and f_{Φ^v} are used to create F_v and finally create $\hat{V}_v$. The relevance score is calculated $\hat{s_{tv}} = \hat{T}_t \hat{V_v^T}$ in the way that tags use this relevance score. It also shows the degree of relevance of a tag to an item. Figure 2 shows the complete architecture of this method.

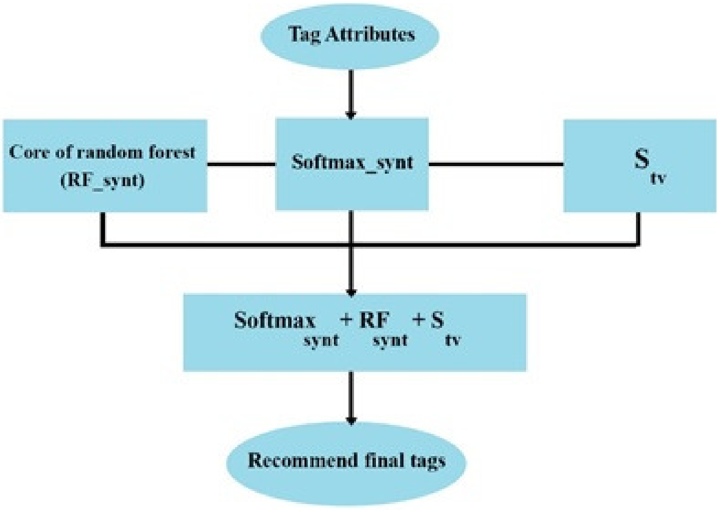


Figure 3: Architecture of the proposed hybrid deep network method.

This paper uses two techniques, deep network, and random forest, which obtain candidate labels from three sources:

1. Other textual characteristics of the desired item (for example, title and description).
2. Labels that occur in the educational data with the words in the text.
3. Labels got from similar items.

In random forest, non-linear learning is a set of decision trees, each of which is trained with subsets of size and distinct from the training set, randomly with replacement to reduce the correlation between them. It is also possible to improve the label suggestion by using dropout in DNN. Also, since RFsynt is the best method compared to other methods, it is considered the primary proposer, so a new version is created as Softmax with syntactic features or called Softmaxsynt.. As a result, a linear combination of the scores of the two methods has been made to combine the advantage points, which is called Resynt + Softmaxsynt. In Figure 3, RFsynt combines features that are effective in label quality, such as syntactic features. It also summarizes the connections between these new methods. RFsynt recommendations are further extended by Softmaxsynt. Therefore, the scores of these two methods are combined by the formula $RF_{synt} + Softmax_{synt} + S_{tv}$. In the following, the operation of the deep network is explained in more detail. At this stage, the random forest algorithm is combined with the core of the deep neural network, and with the assistance of each other, they provide the tag. As shown in Figure 3, RFsynt combines various tag quality features, including syntactic features. The recommendations provided by RFsynt are further expanded using Softmaxsynt. Finally, the scores of these two methods are combined by the formula RFsynt + Softmaxsynt + Stv. The way the deep network works is explained further. It is at this stage that the random forest algorithm is combined with the core of the deep neural network and they provide tags with the help of each other.

4.5 Layers of deep neural network

Softmax is a function that has high efficiency in classification because it calculates the probability of classes with high accuracy. It uses the following formula to calculate this probability. The following formula:

$$p(y_i|x_i;W) = e^{f_{y_i}} / \sum_j e^{f_{y_j}} = Ce^{f_{y_i}} / \sum_j e^{f_{y_j}} = e^{f_{y_i}+\log C} / \sum_j e^{f_{y_j}+\log C} \tag{4.11}$$

In this formula, the class y_i is calculated for each tagx_i and a constant coefficient like $logC$ is added to this fraction for simplicity, and the value of this coefficient is $\log C = -\max_j f_j$ [15].

Data reconstruction in deep neural network models in the neural network does not require much pre-processing of the data and has more generalization on the test data. Traditional neural network models with various layers consider weights to generalize to monitoring data but do not perform well in test data. Therefore, deep network models do not have generalization capabilities, only the monitoring information has been changed to provide labels. These models are usually called deep belief networks and models based on deep neural network encoder-decoder or

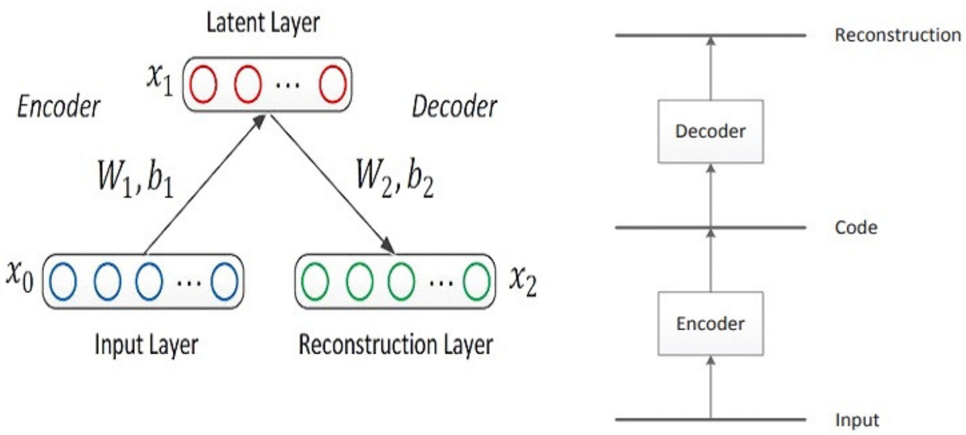


Figure 4: Automatic encoder layer in deep neural network model.[15]

stack auto-encoder models. Deep network models are classified into two types, unsupervised and supervised, such as tag categories. Unsupervised models are used to reach a space of features where there is not much dependence between data. Supervisory models are trained in classification or regression, just like neural and supervisory networks. In deep network models developed to learn new features, the number of layers can be increased, and also in learning nonlinear data transformation.

In this model, there is an encoder and a decoder in each layer, which has a linear transformation in each layer and an activation function for new representation and a transformation for data reconstruction, and in the automatic encoder model, there is a stack, an example of which is presented in Figure 4. As shown in Figure 4; Coding is done by multiplying the inputs by the weight, and they are converted to decoding transfer functions and reconstructed. On the other hand, probabilistic generative models can also perform decoding automatically. Therefore, it can be stated that the deep network models in each layer can reconstruct data with the help of the distribution model in each layer, to maintain the feature space, so there is not much pre-filling in the neural network on the model. On the other hand, traditional neural network models consider weights to generalize monitoring information, but they are not very effective for test data. It can be said that deep network models, do not have generalization capability and only the data display is changed to find generalization with monitoring information or tag.

4.6 Training of models using the method of maximizing the logarithm of the Contrastive Divergence

In the probabilistic generative model, the generator estimates the distribution of characteristics in the base state and does not use distance monitoring information. To teach in the approach of the maximum likelihood function of the data, the parameters of the model should be maximized with the probability of occurrence of the data and considering the parameters. Also, in model training, he solved the following optimization problem, which is equivalent to maximizing the probability of observing the data:

$$(\underset{\theta}{\arg\min}) - \mathcal{L}(\theta|X) \equiv (\underset{\theta}{\arg\min}) - \ln(\mathcal{L}(\theta|(x)^1, \cdots, (x)^S) \tag{4.12}$$

For a specific example $(x)^i$ the precision value is obtained as follows:

$$\mathcal{L}(\theta|x^{'}) = p(x^{'}) = \frac{1}{Z}p^{'}(x^{'}) \tag{4.13}$$

The Z parameter specifies the normalizer constant. To solve this optimization problem, gradient reduction is used. By taking the gradient from the logarithm of both sides, the following relationship is defined

$$\ln(\mathcal{L}(\theta|x^{'})) = \ln(p^{'}(x^{'})) - \ln(Z) \tag{4.14}$$

The existing integral is a time-consuming problem, and on the other hand, the influence of Z is in the optimization problem. Also, due to the existence of a possible distribution, Gibbs sampling is used to estimate the integral, and this sampling requires many steps to achieve an unbiased estimate; Therefore, it is not practical. Also, Hinton's approach with the idea of using training samples and taking samples from the hidden and overt layers, in this approach, a simpler approximation was made by minimizing the confrontation divergence. Gibbs CD sampling algorithm with K step: A training sample of $x^0 = x^I$ should be chosen to be repeated for $I = 1, \cdots, K$.

$$\mathbf{E}[h_k^{l-1}] = \mathbf{E}_{p(h_k|x)}[h_k|\mathbf{E}[x^{l-1}]] \tag{4.15}$$

$$\mathbf{E}[x_i^l] = \mathbf{E}_{p(x_i|h)}[x_i|\mathbf{E}[h^{l-1}]] \tag{4.16}$$

Since the training method of this model, based on gradient reduction, is completely scalable. Briefly the change the gradient in parameters can be described below:

$$\nabla_\theta^L \tau \propto \mathbb{E}_{\text{Data}} \partial(-\mathbf{E}(x))/\partial\theta - \mathbb{E}_{\text{Model}} \partial(-\mathbf{E}(x))/\partial\theta \tag{4.17}$$

E_{Data} with training data and EModel with the parameter j of the model has been estimated in such a way that to estimate E_{Model} the current parameters that are from the CD-1 algorithms of the collection Training data are obtained and used.

4.7 Performance evaluation criteria of recommender system in marketing strategy

The improvement of the sales strategy can be reviewed from the point of view of a recommender system. On the other hand, in this article, the recommender system has become a problem of machine learning and neural network, as in other research, quantitative performance criteria have been used for prediction. It is also possible to check the performance qualitatively, using other websites, which will be discussed in the next chapter. In the end, all the quantitative measures such as MSE, RMSE , MAE , and MAPE are presented along with the graphs of the simulation results, the relationships of the quantitative measures are given below [3]

1. Root Mean Square Error(RMSE)

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\text{Actual}_i - \text{Recommend}_i)^2} \tag{4.18}$$

2. Mean Absolute Percentage of Error(MAPE)

$$MAPE = (\frac{1}{N}\sum_{i=1}^{N} \mid (\text{Actual}_i - \text{Recommend}_i)/Actual_i \mid) * 100 \tag{4.19}$$

3. Mean Absolute Error(MAE)

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\text{Actual}_i - \text{Recommend}_i| \tag{4.20}$$

5 Experimental results

5.1 Analysis of the results of the proposed method

After the simulation of the proposed method in this article, results have been obtained, and in this part, the results of the proposed method are compared with other methods. It should be noted that all the evaluations of the proposed method are applied to film products and the results are discussed. In this article, the relevant simulation has been done on the popular Movielens dataset and the evaluations of the methods presented on this dataset have been compared. The Movielens dataset provides data that includes more than one million points recorded by different users of various movie products. In this part, a set of users registered in the system is separated into different users in stages, clustering is done, and then the desired samples are selected and applied to the recommender system based on a deep neural network. Finally, the movie products predicted by the recommender system and the main products voted by the selected users as training samples are calculated with the evaluation criteria of Mean Actual Error or MAE, Mean Square of Error or RMSE, and compared with other methods. In this article, different stages are defined according to different weights. Then the number of users is increased many times and the results are evaluated.

Table 3: Evaluation parameters of the proposed method

Parameters	Values
Algorithms compared	Quadratic decision tree , multifaceted decision tree, random classification, naïve bayes
Weights	age, gender, and job characteristics
Beta parameter	0.8

Table 4: Evaluation stages of the proposed method with other methods.

levels	weights
Step 1	W1 = 0.6 , W2 = 0.3 , W3 = 0.1
Step 2	W1 = 0.3 , W2 = 0.6 , W3 = 0.1
Step 3	W1 = 0.3 , W2 = 0.1 , W3 = 0.6

5.2 Evaluation parameters of the proposed method

The set of parameters considered to evaluate the proposed method in this article is summarized in Table 3.

In this part, the proposed method is compared with various classification and forecasting algorithms and methods, which are: Decision tree algorithm, Naïve Bayes, and random classification algorithm, which are two of the proposed algorithms in the field of classification. Also, evaluations are evaluated with various weights. The beta parameter, which is described in detail in the fourth section, is also set to 0.8 Scores also have numerical values between 1 (lowest score) and 5 (highest score). Demographic characteristics of age, gender, and occupation are defined for each user. In this article, several stages or scenarios are defined, which are shown in the Table 4.

As can be seen, different weights are obtained for age, gender, and job characteristics. The above steps are defined with different weights. Weights that have more features are more focused on the desired feature and have a greater impact on the degree of similarity. In other words, features that have more weight are more important and play a more important role in determining similarity.

5.3 Analysis of the results of the proposed method

In Table 5, the average amount of prediction errors using the evaluation criterion of the average real error, and the square of the average error in the proposed method compared to other methods have been calculated. The stage considered for the following results is stage 1.

As can be seen, the proposed method with the number of 100 users as input, according to the predictions that have been made, has a real average error of 0.35 and a mean square error of 0.59. As can be seen, the accuracy of the proposed method is much more favorable than the other methods stated in the table above.

Also, in table 6, the average amount of prediction errors using the evaluation criterion of the average real error and the square of the average error in the proposed method compared to other methods have been calculated by considering step number 2.

As can be seen, the proposed method with 100 users as input has MAE = 0.35 and RMSE = 0.59 according to the predictions made. As can be seen, the accuracy of the proposed method is much better than the other methods mentioned in the table above. In the table (6), the average prediction errors have been calculated using the evaluation criteria of the average real error and the square of the average error in the proposed method compared to other methods, considering step 3.

As can be seen, the proposed method with the number of 100 users as input has MAE = 0.35 and RMSE = 0.59 according to the predictions made. As can be seen, the accuracy of the proposed method is much better than the

Table 5: Comparison of the average real error and square average error in the proposed method with other methods with the number of users = 100 (step 1)

Method	Random classification	Naïve Bayes	Multimodal decision tree	Quadratic decision tree	Proposed Method
MAE	0.92	0.89	0.9	0.86	0.35
MSE	1.19	1.13	1.12	1.09	0.59

Table 6: Comparison of the average real error and square average error in the proposed method with other methods with the number of users = 100 (step 2)

Method	Random classification	Naïve Bayes	Multimodal decision tree	Quadratic decision tree	Proposed Method
MAE	0.92	0.89	0.91	0.86	0.35
MSE	1.2	1.14	1.15	1.09	0.59

Table 7: Comparison of the average real error and square average error in the proposed method with other methods with the number of users = 100 (step 3)

Method	Random classification	Naïve Bayes	Multimodal decision tree	Quadratic decision tree	Proposed Method
MAE	0.92	0.9	0.9	0.86	0.35
MSE	1.29	1.14	1.14	1.09	0.59

other methods stated in the table above. In the Table 7, the average amount of prediction errors are calculated using the evaluation criterion of the average real error and the square of the average error in the proposed method compared to other methods. The stage considered for the following results is stage 1.

As can be seen, the proposed method with the number of 500 users as input has MAE = 0.76 and RMSE = 1.03 according to the predictions made. As can be seen, the accuracy of the proposed method is much better than the other methods mentioned in the table above. Also, in the Table 8, the average amount of prediction errors using the evaluation criterion of the average real error and the square of the average error in the proposed method compared to other methods have been calculated by considering step number 2.

As can be seen, the proposed method with the number of 500 users as input, according to the predictions that have been made, has an average true error equal to 0.76 and an average square error equal to 1.03. As can be seen, the accuracy of the proposed method is much better than the other methods stated in the table above. In the Table 9, the average prediction errors have been calculated using the evaluation criterion of the average real error and the square of the average error in the proposed method compared to other methods, considering step 3.

As can be seen, the proposed method with the number of 500 users as input has MAE = 0.76 and RMSE = 1.03 according to the predictions made. As can be seen, the accuracy of the proposed method is far better than the other methods stated in the Table 9. In the table 10, the average amount of prediction errors are calculated using the evaluation criteria of the average real error and the square of the average error in the proposed method compared to other methods. The stage considered for the following results is stage 1.

As can be seen, the proposed method with the number of 900 users as input has MAE = 0.73 and RMSE = 0.95 according to the predictions made. As can be seen, the accuracy of the proposed method is much better than the other methods stated in the table. (10) Also, in Table 11, the average amount of prediction errors using the evaluation criteria of the average real error and the square of the average error in the proposed method compared to other methods have been calculated by considering step number 2.

As can be seen, the proposed method with the number of 900 users as input has MAE = 0.73 and RMSE = 0.95 according to the predictions made. As can be seen, the accuracy of the proposed method is far better than the other methods stated in the Table 11. In the Table 12, the average prediction errors have been calculated by using the evaluation criterion of the average real error and the square of the average error in the proposed method compared to

Table 8: Comparison of the average real error and square average error in the proposed method with other methods with the number of users = 500 (step 1)

Method	Random classification	Naïve Bayes	Multimodal decision tree	Quadratic decision tree	Proposed Method
MAE	0.87	0.835	0.845	0.83	0.76
MSE	1.1	1.03	1.09	1.03	1.03

Table 9: Comparison of the average real error and square average error in the proposed method with other methods with the number of users = 500 (step 2)

Method	Random classification	Naïve Bayes	Multimodal decision tree	Quadratic decision tree	Proposed Method
MAE	0.87	0.839	0.85	0.83	0.76
MSE	1.1	1.06	1.09	1.06	1.03

Table 10: Comparison of the average real error and square average error in the proposed method with other methods with the number of users = 500 (step 3)

Method	Random classification	Naïve Bayes	Multimodal decision tree	Quadratic decision tree	Proposed Method
MAE	0.87	0.84	0.86	0.83	0.76
MSE	1.09	1.06	1.08	1.06	1.03

Table 11: Comparison of the average real error and square average error in the proposed method with other methods with the number of users = 900 (step 1)

Method	Random classification	Naïve Bayes	Multimodal decision tree	Quadratic decision tree	Proposed Method
MAE	0.86	0.83	0.83	0.83	0.73
MSE	1.04	1.01	1.01	1.01	0.95

Table 12: Comparison of the average real error and square average error in the proposed method with other methods with the number of users = 900 (step 2)

Method	Random classification	Naïve Bayes	Multimodal decision tree	Quadratic decision tree	Proposed Method
MAE	0.82	0.82	0.82	0.82	0.73
MSE	1.04	1.01	1.01	1.01	0.95

Table 13: Comparison of the average real error and square average error in the proposed method with other methods with the number of users = 900 (step 3)

Method	Random classification	Naïve Bayes	Multimodal decision tree	Quadratic decision tree	Proposed Method
MAE	0.82	0.82	0.82	0.82	0.73
MSE	1.04	1.01	1.01	1.01	0.95

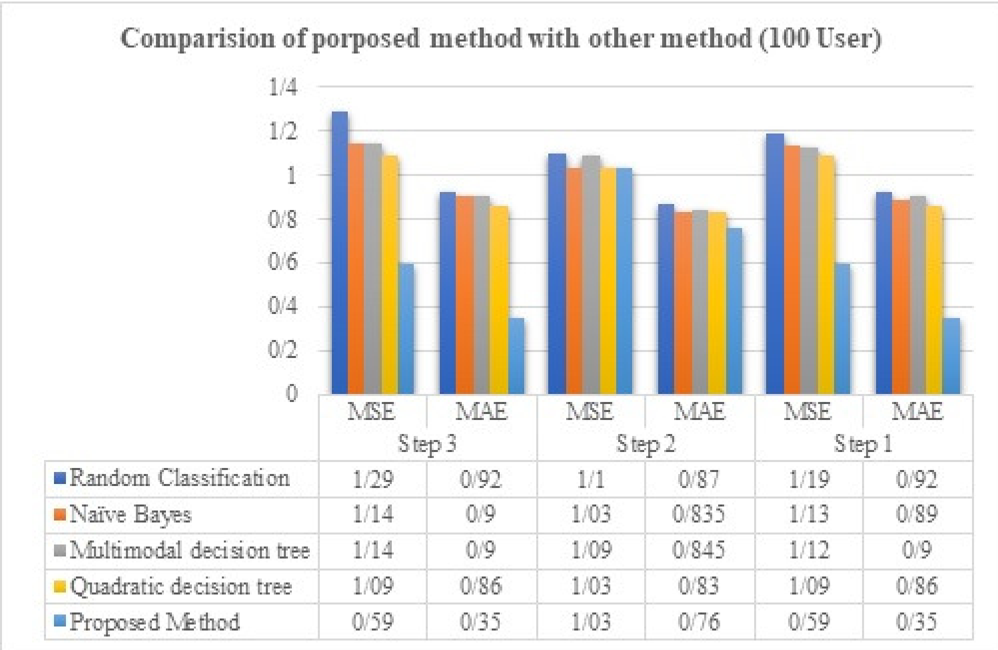


Figure 5: Evaluation of the proposed method with other methods with the number of 100 users.

other methods, taking step 3 into account.

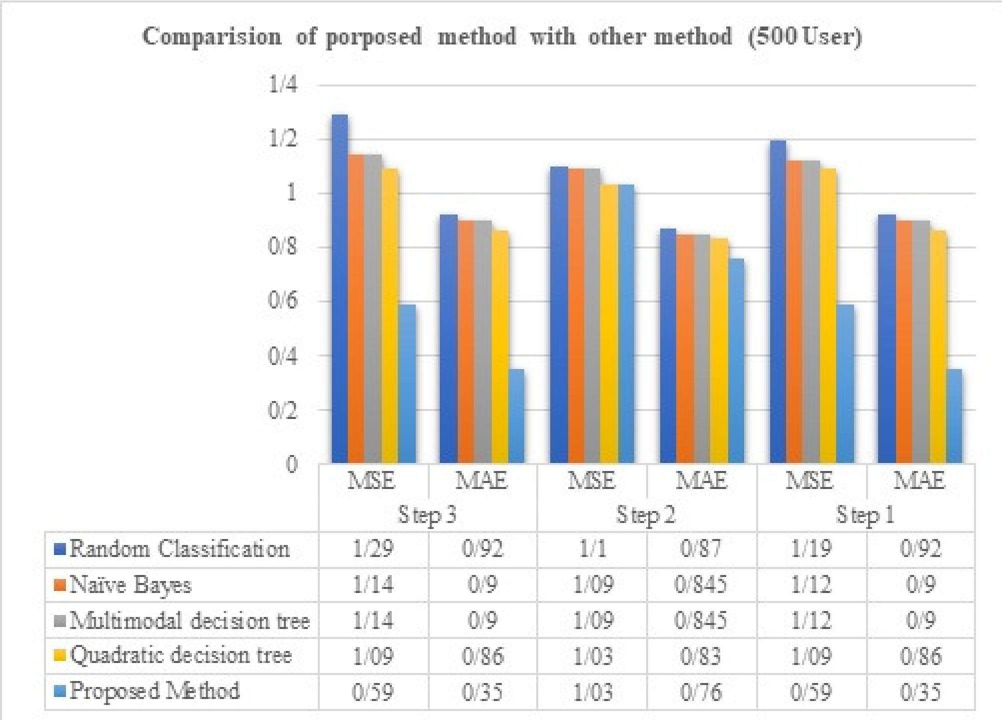
As can be seen, the proposed method with the number of 900 users as input has MAE = 0.73 and RMSE = 0.95 according to the predictions that have been made. As can be seen, the accuracy of the proposed method is much better than the other methods stated in the Table 13.

Figure 5 shows a summary of the evaluation of the proposed method based on the average true error criterion with other methods with the number of 100 users. Also, Figure 6 provides a summary of the evaluations related to steps 1, 2, and 3 with the number of 500 users. Finally, We can observe a summary of the evaluations related to steps 1, 2, and 3 with the number of 900 users in Figure 7.

According to the evaluations made on the proposed method and comparison with other recent research that was done in 2014 and 2015, in Figure 8 the actual error rate (average of the actual error) of the proposed method with other basic methods such as Algorithms of a decision tree, Naive Bayes, random classification [13] and also algorithms such as SVD and ApproSVD have been compared [27].

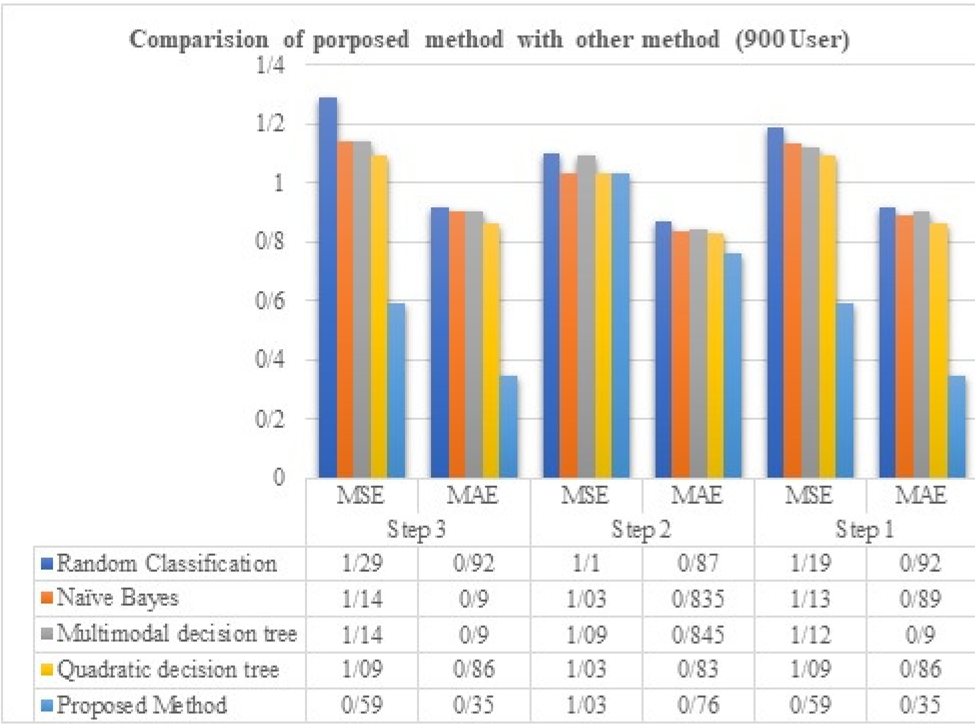
Also, RMSE of the proposed method in comparison with other methods is shown in the Figure 9. As can be seen from the figure above, the MAE in the proposed method is equal to 0.35, which compared to other methods of a decision tree, Naïve Bayes, random classification, SVD method, and ApproSVD method is about 0.51, 0.54, 0.57, and 0.425 respectively. and has improved by 0.43.

The amount of RMSE in the proposed method is equal to 0.59, which has improved by 0.5, 0.54, 0.6, 0.385, and 0.39, respectively, compared to other methods of a decision tree, Naïve Bayes, random classification, SVD method, and ApproSVD method. As can be seen, the proposed method has performed better than other methods. In another experiment, the results of the proposed method have been evaluated and compared with the method presented in the article [14]. In this paper,authors presented a neural network for a movie recommendation system. Table 14 shows the comparison of the proposed method using the deep neural network and random forest methods with the methods



	Step 3 MSE	Step 3 MAE	Step 2 MSE	Step 2 MAE	Step 1 MSE	Step 1 MAE
Random Classification	1/29	0/92	1/1	0/87	1/19	0/92
Naïve Bayes	1/14	0/9	1/09	0/845	1/12	0/9
Multimodal decision tree	1/14	0/9	1/09	0/845	1/12	0/9
Quadratic decision tree	1/09	0/86	1/03	0/83	1/09	0/86
Proposed Method	0/59	0/35	1/03	0/76	0/59	0/35

Figure 6: Evaluations related to steps 1, 2, and 3 with the number of 500 users.



	Step 3 MSE	Step 3 MAE	Step 2 MSE	Step 2 MAE	Step 1 MSE	Step 1 MAE
Random Classification	1/29	0/92	1/1	0/87	1/19	0/92
Naïve Bayes	1/14	0/9	1/03	0/835	1/13	0/89
Multimodal decision tree	1/14	0/9	1/09	0/845	1/12	0/9
Quadratic decision tree	1/09	0/86	1/03	0/83	1/09	0/86
Proposed Method	0/59	0/35	1/03	0/76	0/59	0/35

Figure 7: Evaluations related to steps 1, 2, and 3 with the number of 900 users.

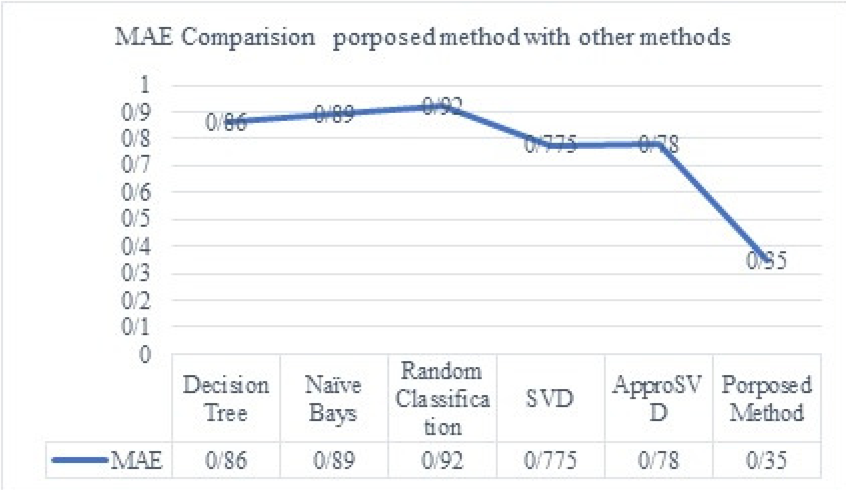


Figure 8: MAE Comparison.

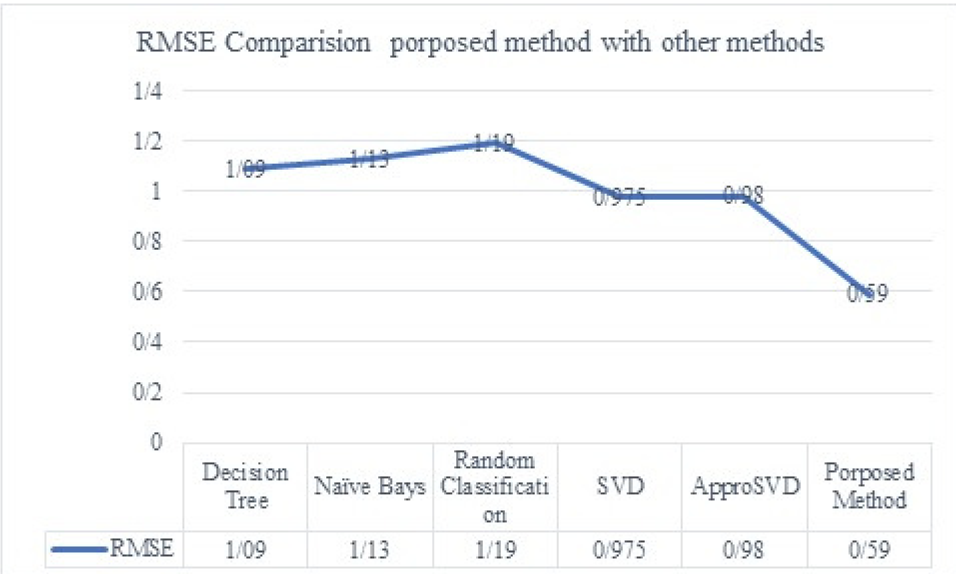


Figure 9: RMSE Comparison.

Table 14: The comparison based on recommended method on DNN with Scikit-Learn and Tensor Flow methods.

Method	Type of recommender system	MAE	Processing Time(s)
Scikit -learn	Typical recommender system	0.88	31.49
TensorFlow		0.88	104.336
Scikit -learn	Recommender system based on neural network	1.43	6264.01
TensorFlow		0.76	136.28
Proposed Method	Based on the proposed deep neural network	0.35	99.021

presented in the article [14] such as Scikit-Learn, and TensorFlow. Scikit-Learn is a basic machine learning tool with many ready-to-call libraries and functions and powerful and fast math capabilities. TensorFlow is a tool that has emerged in recent years with the development of deep learning.

As can be seen from Table 14, the MAE of the proposed method compared to other methods Scikit-Learn and TensorFlow basedon neural network and without applying neural network performs better. The processing time of the proposed method is less than other methods.

6 Conclusion

In the category of internet marketing, to present a product in the internet market, it must be easily retrieved by search engines. The more easily the products are retrieved in these search engines, the easier they can be offered to the applicants of this product and capture a higher share of this internet market. Product tags are the most basic part of a product used by search engines. Tags or labels express the nature of the product and the important characteristics of the product. In addition, tags are indexed by search engines and are known as product descriptors. In addition, the more suitable tags are intended for products, the better they can be recognized by search engines. And get more share of internet marketing. In this article, tag recommendations are provided to improve the internet retrieval of products.

The main goal of the current paper is to present a hybrid tag recommendation model to improve Internet (network) marketing strategy by using DNNs to overcome the cold start problem in online store networks of products such as movies suitable for new users with acceptable accuracy as suggestions by using a combination of suggestion systems. It has been proposed as well as the use of data mining techniques such as a deep neural network. In this research, the researcher has been able to present videos based on tags for the new users using similarity criteria and deep learning algorithms, which is more accurate than other methods that have been done so far. Therefore, when a new user enters the system considering that he has not rated any product and is facing a cold start, the proposed method first uses the clustering algorithm and using the capabilities of the boosting method, based on the demographic information of the user, to select the desired cluster for the user determines and then extracts the closest similar users using a composite similarity measure. Finally, by using a prediction strategy such as a deep neural network and based on the scores provided by neighboring users to each product and combining with the degree of similarity of neighboring users with the new user based on the weighting applied in the previous step, the desired score for each product by the new user It is predicted and after sorting, it presents related products. According to the simulation done to provide attractive products to new users who are facing a cold start, the proposed method provides relevant and attractive products to new users at the right time with acceptable accuracy compared to other methods. Therefore, another major problem is the issue of trust, which is caused by not paying attention to old users compared to new users.

The main goal and final output of this research is a tag recommender system for products that can suggest a series of tags for different objects and products based on the information available from that object. These tags can be very useful for introducing objects to information retrieval systems and search engines, increasing sales, and improving internet marketing strategy. Therefore, as can be seen from Figure (1), to implement the proposer system proposed to provide tags to users, first the dataset or data set is entered into the proposed system. Then the pre-processing process is done on the data. In the pre-processing stage, non-numerical variables such as protocol type should be converted into numerical variables. For this type of transformation, there are two main binary and histogram-based methods (Target-based), which, due to the disadvantages of the binary method, the Target-based method is used. In this method, each non-numerical variable is replaced by a number, and then the probability of its repetition (the corresponding histogram) is considered a numerical value. At this stage, considering that the accuracy of the suggestions should be increased, the data is normalized so that their values are between 0 and 1. After data normalization, the similarity between users is calculated. For this purpose, matching relationships based on users' demographic information are used. In the next step, training and testing samples are separated for training and testing deep neural networks and random forest algorithms. Training samples are used to train a deep neural network and random forest algorithms, and test samples are used to evaluate the produced models. Finally, based on the training and experimental samples, models of deep neural networks and random forest algorithms are produced. With the help of generated models, tags are proposed for new entries.

References

[1] D L. Abd Al-Nabi and S Sh. Ahmed, *Deep learning for big weather data analyzing and forecasting*, Int. J. Nonlinear Anal. Appl. **In Press** (2023), doi: 10.22075/ijnaa.2023.30013.4313.

[2] F.M. Belém, A.G. Heringer, J.M. Almeida, and M.A. Gonçalves, *Exploiting syntactic and neighbourhood attributes to address cold start in tag recommendation*, Inf. Process. Manag. **56** (2019), no. 3, 771–790.

[3] B. Doucoure, K. Agbossou, and A. Cardenas, *Time series prediction using artificial wavelet neural network and multi-resolution analysis: Application to wind speed data*, Renewable Energy **92** (2016), 202–211.

[4] T. Ebesu and Y. Fang, *Neural semantic personalized ranking for item cold-start recommendation*, Inf. Retriev. J. **20** (2017), no. 2, 109–131.

[5] M. Fadavi Amiri, M. Hosseinzadeh, and S.M.R. Hashemi, *Improving image segmentation using artificial neural networks and evolutionary algorithms*, Int. J. Nonlinear Anal. Appl. **In Press** (2023), doi: 10.22075/ijnaa.2023.30232.4371.

[6] F. Figueiredo, H. Pinto, F. Belém, J. Almeida, M. Gonçalves, D. Fernandes, and E. Moura, *Assessing the quality of textual features in social media*, Inf. Process. Manag. **49** (2013), no. 1, 222–247.

[7] K.K. Fletcher, *A method for dealing with data sparsity and cold-start limitations in service recommendation using personalized preferences*, 2017 IEEE Int. Conf. Cogn. Comput. (ICCC) (Honolulu, HI, USA), IEEE, June 2017, pp. 72–79.

[8] A. Ghanbari Sorkhi, M. Iranpour Mobarakeh, S.M.R. Hashemi, and M. Faridpour, *Predicting drug-target interaction based on bilateral local models using a decision tree-based hybrid support vector machine*, Int. J. Nonlinear Anal. Appl. **12** (2021), no. 2, 135–144.

[9] I. Guy, N. Zwerdling, I. Ronen, D. Carmel, and E. Uziel, *Social media recommendation based on people and tags*, Proc. 33rd Int. ACM SIGIR Conf. Res. Dev. Inf. Retriev., 2010, pp. 194–201.

[10] M.-H. Hsu and H.-H. Chen, *Efficient and effective prediction of social tags to enhance web search*, J. Amer. Soc. Inf. Sci. Technol. **62** (2011), no. 8, 1473–1487.

[11] K. Lei, Q. Fu, M. Yang, and Y. Liang, *Tag recommendation by text classification with attention-based capsule network*, Neurocomputing **391** (2020), 65–73.

[12] X. Li, L. Guo, and Y.E. Zhao, *Tag-based social interest discovery*, Proc. 17th Int. Conf. World Wide Web, 2008, pp. 675–684.

[13] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, *Facing the cold start problem in recommender systems*, Expert Syst. Appl. **41** (2014), no. 4, 2065–2073.

[14] C.-H. Lin and H. Chi, *A novel movie recommendation system based on collaborative filtering and neural networks*, Adv. Inf. Network. Appl.: Proc. 33rd Int. Conf. Adv. Inf. Network. Appl. (AINA-2019) 33, Springer, 2020, pp. 895–903.

[15] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F.E. Alsaadi, *A survey of deep neural network architectures and their applications*, Neurocomputing **234** (2017), 11–26.

[16] L. Luo, H. Xie, Y. Rao, and F.L. Wang, *Personalized recommendation by matrix co-factorization with tags and time information*, Expert Syst. Appl. **119** (2019), 311–321.

[17] M.A. Masood, R.A. Abbasi, O. Maqbool, M. Mushtaq, N.R. Aljohani, A. Daud, M.A. Aslam, and J.S. Alowibdi, *MFS-LDA: A multi-feature space tag recommendation model for cold start problem*, Program **51** (2017), no. 3, 218–234.

[18] V. Oliveira, G. Gomes, F. Belém, W. Brandao, J. Almeida, N. Ziviani, and M. Gonçalves, *Automatic query*

expansion based on tag recommendation, Proc. 21st ACM Int. Conf. Inf. Knowledge Manag., 2012, pp. 1985–1989.

[19] A. Tejeda-Lorente, J. Bernabé-Moreno, C. Porcel, and E. Herrera-Viedma, *Using Bibliometrics and Fuzzy Linguistic Modeling to Deal with Cold Start in Recommender Systems for Digital Libraries*, Advances in Fuzzy Logic and Technology 2017 (Janusz Kacprzyk, Eulalia Szmidt, Slawomir Zadrożny, Krassimir T. Atanassov, and Maciej Krawczak, eds.), vol. 643, Springer International Publishing, Cham, 2018, Series Title: Advances in Intelligent Systems and Computing, pp. 393–404.

[20] Ma. Volkovs, G.W. Yu, and T. Poutanen, *Content-based neighbor models for Cold Start in Recommender Systems*, Proc. Recomm. Syst. Challenge 2017 (Como Italy), ACM, August 2017, pp. 1–6.

[21] L. Wang, Ch. Wang, K. Wang, and X. He, *BiUCB: A contextual bandit algorithm for cold-start and diversified recommendation*, IEEE Int. Conf. Big Knowledge (ICBK) (Hefei, China), IEEE, 2017, pp. 248–253.

[22] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, *Collaborative filtering and deep learning based recommendation system for cold start items*, Expert Syst. Appl. **69** (2017), 29–39.

[23] Y. Wu, S. Xi, Y. Yao, F. Xu, H. Tong, and J. Lu, *Guiding supervised topic modeling for content based tag recommendation*, Neurocomputing **314** (2018), 479–489.

[24] Q. Xie, F. Xiong, T. Han, Y. Liu, L. Li, and Z. Bao, *Interactive resource recommendation algorithm based on tag information*, World Wide Web **21** (2018), 1655–1673.

[25] Z. Xu, D. Yuan, T. Lukasiewicz, Ch. Chen, Y. Miao, and G. Xu, *Hybrid deep-semantic matrix factorization for tag-aware personalized recommendation*, ICASSP 2020-2020 IEEE Int. Conf. Acoustics Speech Signal Process. (ICASSP), IEEE, 2020, pp. 3442–3446.

[26] H. Yu, B. Zhou, M. Deng, and F. Hu, *Tag recommendation method in folksonomy based on user tagging status*, J. Intell. Inf. Syst. **50** (2018), 479–500.

[27] X. Zhou, J. He, G. Huang, and Y. Zhang, *SVD-based incremental approaches for recommender systems*, J. Comput. Syst. Sci. **81** (2015), no. 4, 717–733.