IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 30, 2019, accepted August 19, 2019, date of publication August 26, 2019, date of current version September 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2937518

Customer Reviews Analysis With Deep Neural Networks for E-Commerce Recommender Systems

BABAK MALEKI SHOJA AND NASSEH TABRIZI
Department of Computer Science, East Carolina University, Greenville, NC 27858, USA

Corresponding author: Babak Maleki Shoja (malekishojab16@students.ecu.edu)

This work was supported in part by the National Science Foundation under Grant 1560037.

ABSTRACT An essential prerequisite of an effective recommender system is providing helpful information regarding users and items to generate high-quality recommendations. Written customer review is a rich source of information that can offer insights into the recommender system. However, dealing with the customer feedback in text format, as unstructured data, is challenging. In this research, we extract those features from customer reviews and use them for similarity evaluation of the users and ultimately in recommendation generation. To do so, we developed a glossary of features for each product category and evaluated them for removing irrelevant terms using Latent Dirichlet Allocation. Then, we employed a deep neural network to extract deep features from the reviews-characteristics matrix to deal with sparsity, ambiguity, and redundancy. We applied matrix factorization as the collaborative filtering method to provide recommendations. As the experimental results on the Amazon.com dataset demonstrate, our methodology improves the performance of the recommender system by incorporating information from reviews and produces recommendations with higher quality in terms of rating prediction accuracy compared to the baseline methods.

INDEX TERMS Recommender system, review, deep neural networks, recommendation, matrix factorization, latent Dirichlet allocation.

I. INTRODUCTION

The exponential growth of data and information on the Internet confronts us with information overload. This results in a tremendous amount of information that makes it hard for people to make choices between an enormous number of movies, books, web pages, and other products which poses a challenge to user's ability to efficiently access required data [1], [2]. Evaluating even a small portion of such data seems to be impractical, increasing the need for automatic recommender systems with the capability of suggesting relevant items as well as new items to the customers and clients [3], [4]. Besides, personalization and customization for users and providing suggestions in the ever increasing information is a crucial and challenging problem for online service providers such as e-learning. Recommender systems are a branch of information filtering systems that try to predict users' preferences for an item and provide personalized suggestions based on this analysis for a particular user. In other words, recommender systems help users to find products or services they need based on analysis of user preferences using client profiles and their similarities or finding products or services that are similar to those clients who have already expressed interest in [5]. Recently, there is an increasing trend in employing this approach to various areas, including music, book, social tags, and products. Several e-commerce companies such as Amazon employ recommender systems and related tools to enhance the recommendations to their customers with the primary purpose of increasing overall profits [4], [6]–[9].

Generating proper recommendations to the user requires information about the user's characteristics, preferences, and needs [9]. Recommender systems mainly consider the overall rating a customer gives to items and latent factor models such as matrix factorization (MF) are widely used to predict ratings. However, there are drawbacks for using MF models

The associate editor coordinating the review of this article and approving it for publication was Hualong Yu.

such as cold-start problem, considering only the customer overall satisfaction, and sparsity. As the literature on the MF methods show, numerous researches are devoted to tackling the weaknesses of MF methods by incorporating side information such as tags [10], [11], visual features [12], and social relations [13], [14]. Customer reviews are one of the critical resources in developing recommender systems. A written part of the review of a rating includes essential information on what the customer thinks about the product.

Consequently, researchers suggest many models that exploit reviews with ratings for improving the recommendations. Some of these models are discussed in [15]–[18]. Sentiment analysis is one of the conventional approaches toward the analysis of customer reviews. It is mainly to predict whether the attitude of a piece of text is positive or negative, supported or opposed [19]. Semantic analysis is employed to analyze customer reviews [19]–[21] for different objectives such as to measure e-commerce service quality [22]. Some recent studies try to use customer reviews in developing recommender systems. The approaches they utilized include semantic analysis and aspect-based latent factor models [23]–[26]. In this paper, we perform a customer review mining and extract a set of product characteristics that users mentioned in the reviews and will use the Latent Dirichlet Allocation (LDA) method to finalize the set of characteristics. We then use the set of attributes to construct the users-attributes matrix. This matrix, however, is very sparse as each user mentions only a few attributes in the review. Sparsity is a well-known challenge in developing recommender systems. Many papers propose various solutions to deal with the sparsity problem. To deal with this problem, we use a deep neural network that plays an autoencoder role which helps to learn more abstract and latent attributes. Having users-attributes and users-items matrix, we use an MF model to predict ratings and provide recommendations.

The rest of this paper is organized as follows. Section II presents the literature survey and related works. Section III discusses the problem description and the proposed approach. Section IV provides information about the dataset and how we conducted data preprocessing and analyzes the outcomes of the experimental analysis. Finally, we conclude current research, its limitations, and future research directions.

II. RELATED WORK

Dealing with text as unstructured data is challenging. Natural Language Processing (NLP) is a branch of computer science and artificial intelligence (AI) concerned with processing and analyzing natural language data. Deep learning for NLP is one of the approaches that is improving the capability of the computer to understand human language [27]. There are a few studies that try to incorporate customer written reviews in generating recommendations.

Some researches on the integration of customer reviews in recommendation systems are under the category of aspect-based or aspect-aware recommender systems. As an aspect-based recommender system, [24] proposed a model called aspect-based latent factor model which integrates ratings and review texts via latent factor model. The purpose of this research is predicting ratings by using aspects of information from users' and items' information. They constructed user-review and used it directly in the proposed model to provide rating predictions and recommendation lists. Besides, their model accomplishes a cross-domain task by transferring word embedding.

In another aspect-based recommender system paper, [25] proposed an aspect-aware MF model that effectively combines reviews and ratings for rating predictions. It learns the latent topics from reviews and ratings without having the constraint of a one-to-one mapping between latent factors and latent topics. Also, the model estimates aspect ratings and assign weights to the aspects. They performed experimental results on many real-world datasets and showed the performance of their models in accurately predicting the ratings.

Some aspect-based recommender systems utilize semantic analysis on reviews. For example, [26] proposed a sentiment utility logistic model that uses sentiment analysis of user reviews where it predicts the sentiment that the user has about the item and then identifies the most valuable aspects of the user's possible experience with that item. For example, the system suggests a user going to a specific restaurant (as the primary recommendation), and also it recommends an aspect of that restaurant like the time to go to a restaurant (breakfast, lunch, or dinner) as a valuable aspect to the user (the secondary recommendation). The experimental results demonstrated the better experience of those users who followed the recommendations.

In the context of analyzing reviews, [28] analyzed customer reviews to find out what makes a review helpful to other customers. They analyzed 1,587 reviews from Amazon.com and indicated that extremity, depth of review, product type affect the perceived helpfulness of the review. While this research does not incorporate the reviews in making recommendations, it provides information that is potentially useful in developing recommender systems.

Reference [29] proposed a new recommender system that integrates opinion mining and recommendations. They proposed a new feature and opinion extraction method based on the characteristics of online reviews which can address the problem of data sparseness. They used the part-of-speech tagging approach based on association rule mining for each review. They performed their empirical study on online restaurant customer reviews written in Chinese and illustrated the performance of the proposed methods.

Reference [15] considered the review texts using topic modeling techniques and align the topic with rating dimensions to enhance the prediction accuracy. They proposed a unified model combining content-based and collaborative filtering, which can deal with the cold-start problem. They applied the proposed framework to 27 classes of real-case datasets and showed the significant improvement of the recommendations comparing to the baselines methods.

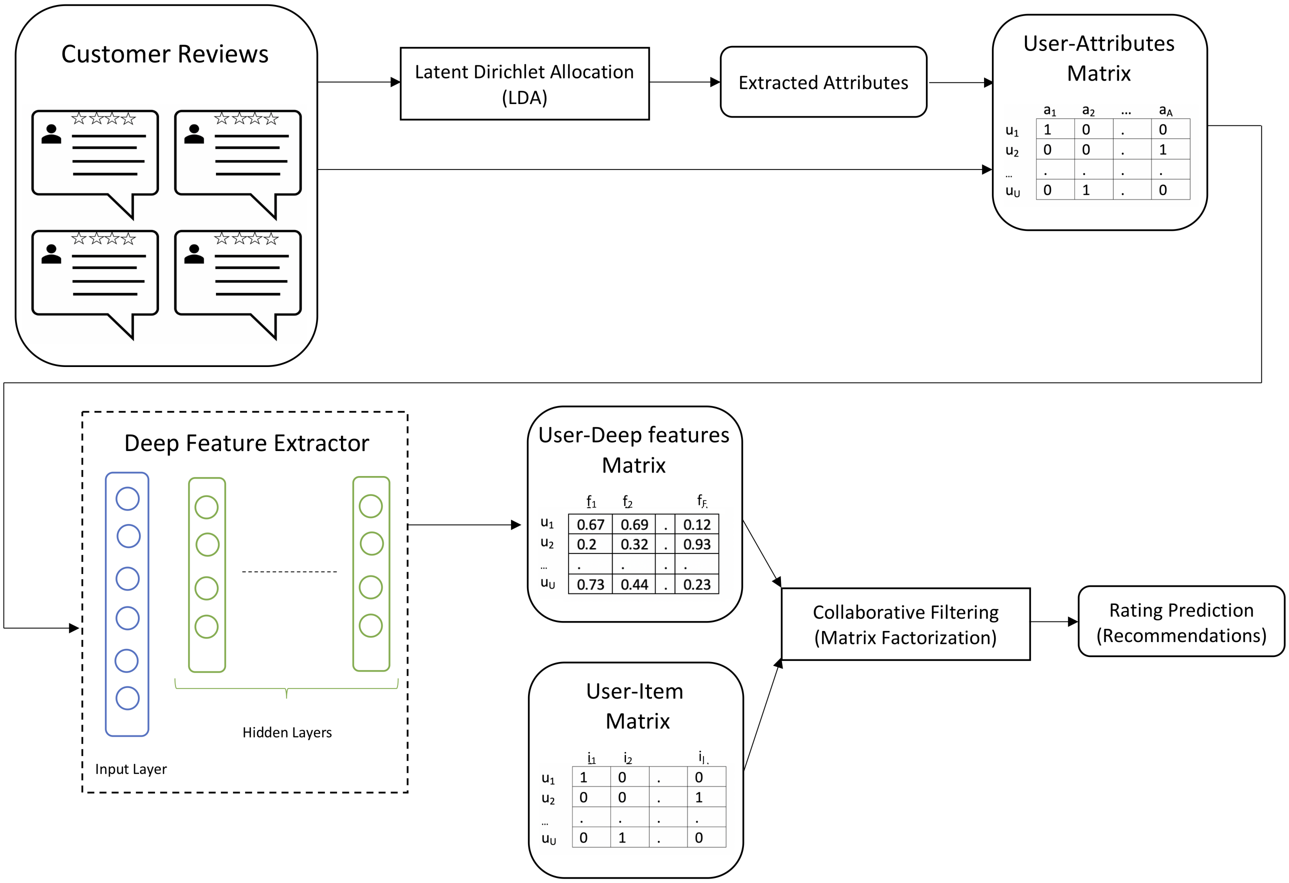


FIGURE 1. General framework of the proposed approach of using customer reviews in recommendation systems.

Reference [16] tried to incorporate the implicit tastes of each user in order to predict ratings as the text review justifies a user's rating. They used latent review topics extracted from topic models as highly interpretable textual labels for latent rating dimensions. Also, they accurately predicted product ratings using the information extracted from the reviews, which can improve the recommendations for those that have too few ratings. Moreover, their discovered topics are useful in facilitating tasks such as automated genre discovery. In a similar study, [17] exploit textual review information along with ratings to model user preferences and item attributes in a shared topic space. They used an MF model for generating recommendations and used 26 real-case datasets to evaluate the performance of their model.

As presented above, none of the abovementioned studies used a deep neural network autoencoder to deal with the sparsity in the user-attributes matrix extracted from the reviews. To the best of the authors' knowledge, this is the first study that extracts deep features from extracted latent topics from the textual user reviews to develop a recommender system. In the next section, we present the proposed approach.

III. METHODOLOGY

In this section, we provide the proposed methodology for incorporating customer written reviews in developing recommender systems. Figure 1 depicts the general framework for transforming customer written reviews into a dense users-attributes matrix and predicting ratings using this matrix and users-items matrix. As described before, the idea of how to use customer written reviews is investigating what attributes of the product category are mentioned in the customer's review. In doing so, we need to match the review with a set of predefined product attributes. As Fig 1 demonstrates, we use Latent Dirichlet Allocation (LDA) to analyze the reviews on a product category and retrieve a dictionary of attributes. Afterward, we can construct the users-attributes matrix, which indicates what attributes the user has pointed out in his or her reviews in a binary format. The major challenge with this matrix is a well-known problem called sparsity. Besides, there are other problems, including ambiguity and redundancy, regarding the extracted attributes in the matrix. To deal with this problem, we propose a deep neural network approach to transform this sparse matrix into a dense matrix presenting a set of deep features extracted from the users-attributes matrix and construct the users-deep features matrix. We use this matrix and users-items matrix to predict ratings and generate recommendations via Matrix Factorization (MF) as a powerful and efficient collaborative filtering method. In the following subsections, we present and describe DLA, deep neural network model, and the MF method used in this research.

A. LATENT DIRICHLET ALLOCATION

The basic latent factor model predicts ratings for a user and item using user and item biases, K-dimensional user and item factors including the item's properties and the user's preferences minimizing the Mean Squared Error (MSE). There are a variety of methods for optimizing MSE for this problem, such as alternating least-squares and gradient-based methods [30]. While latent factor models try to uncover hidden dimensions in review ratings, LDA aims to uncover hidden dimensions in the written part of the review. Introduced by Blei *et al.* [31]. LDA is a generative statistical model for topic modeling in the natural language processing (NLP) context. Topic modeling is the task of describing a collection of documents by identifying a set of topics. In LDA, we model each item of the collection as a finite mixture over an underlying set of topics as a three-level hierarchical Bayesian model. We also model each topic as an infinite mixture over an underlying set of topic probabilities which provides an explicit representation of a document [23]. In order to describe LDA, a set of documents $d \in D$ and LDA associate each document with a K-dimensional stochastic vector as a topic distribution θ_d. This association encodes the fraction of words in a document that discusses the topic k with the probability of $\theta_{d,k}$. LDA associate a word distribution, ϕ_k to each topic to encode the probability of a word used for that topic. LDA assumes a Dirichlet distribution for the topic (θ_d). As a result of applying LDA, we have word distribution of each topic and topic distribution for each document. Having the word distribution and topic assignment of the words, we can calculate the likelihood of a corpus T as

$$p(T|\theta, \phi, z) = \prod_{d \in T} \prod_{j=1}^{N_d} \theta_{z_{d,j}, w_{d,j}} \tag{1}$$

where z is topic assignments updated via sampling. This likelihood is a product of the probability of the topic being the document and the word being the topic [16].

LDA results in a vast number of words from the reviews. Inspired by [16], we filter the extracted words using frequent itemsets using association rules to prune the set of words LDA provides. Association rule mining uses two metrics, including support and confidence where support is a measure that shows if the itemset appears in the dataset frequently, and confidence shows how often a rule can be found.

B. DEEP NEURAL NETWORKS

Sparsity is a significant problem in the recommender systems, which significantly reduces the performance of the rating prediction. The problem of sparsity is sometimes called gray sheep problem, which is peculiar to similarity-based collaborative recommendation systems. The problem arises from the fact that users-attributes interaction will occur for a tiny percentage of all possible interactions because the user only mentions a tiny portion of all the attributes in the written review [7] that makes some users not similar enough to others to discover their preferences. Hence, the system cannot retrieve proper recommendations. In this regard, many investigators have focused on dealing with this problem to provide a solution that mitigates the effect of sparsity [32], [33]. Here, we propose a deep neural network approach to deal with the sparsity in the users-attributes matrix and transform it into a dense matrix. Here, we describe the details of the proposed deep neural networks to process the attributes extracted using LDA.

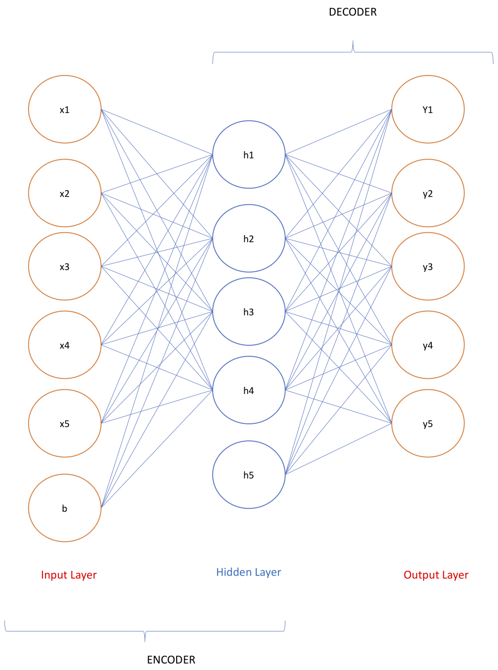


FIGURE 2. **Demonstration of a simple autoencoder.**

The reason for using sparse coding is to learn more interpretable features for machine learning applications [34] and it helps at representing the input matrix as a weighted linear combination of a small number of basis vectors. The resulted matrix is capable of capturing high-level patterns that exist in the input layer. For instance, [35] developed a sparse autoencoder as the result of combined sparse coding with the autoencoder. They implemented their idea by penalizing the deviation between the expected hidden representation and present average activation. In more relevant research, [8] developed an autoencoder using deep neural networks for tag-aware recommender systems. Through experimental results, they demonstrated the usefulness of the sparse autoencoders for the recommendation algorithms.

Inspired by [8], an autoencoder constitutes an input layer, a hidden layer, and an output layer. We can divide the autoencoder itself into an encoder and decoder. The encoder is the input layer and output layer, while the decoder is the hidden layer and the output layer. Figure 2 illustrates the purpose of the autoencoder, which is reconstructing the input data in the output layer with the same dimensionality. In other words, this follows an unsupervised learning framework.

Letting $x_1, x_2, \ldots, x_m$ be an unlabeled dataset, we can obtain the nonlinear representation of the input data using activation function [36]. Using a sigmoid activation for an unlabeled dataset x_i, the representation is

$$h(x_i; W, b) = \sigma(Wx_i + b) \tag{2}$$

where W denotes weight matrix, σ is the sigmoid activation function, and b is the bias term. This representation is also called the hyperbolic tangent function. On the other side, the decoder reconstructs the input into the output layer by minimizing the error between the input and the output layers. The minimizing term is defined in Equation (4.3).

$$min \sum_{i=1}^{m} ||\sigma(W^T h(x_i; W, b) + c) - x_i||^2 \tag{3}$$

where m denotes the number of examples. Since the minimization is a convex function, we can obtain the optimal value. We can calculate the sparsity penalty term by Kullback-Leibler divergence between a preferred activation ratio in the hidden layer and the desired hidden representations [37] using Equation (4.4).

$$P = \sum_{j=1}^{n} D_{KL}(\rho||\hat{\rho}_j) \tag{4}$$

in which ρ is a reset average activation that is set to be close to zero in practice, and n is the number of hidden units. Also, $D_{KL}(\rho||\hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho)\log \frac{1-\rho}{1-\hat{\rho}_j}$.

Combining the objective function and penalty term introduced above, we obtain the final objective function for the autoencoder using Equation (4.5).

$$min \sum_{i=1}^{m} ||\sigma(W^T h(x_i; W, b) + c) - x_i||^2 + \beta P \tag{5}$$

where β is a hyperparameter to change the weight of the penalty term.

In order to transform this architecture into an autoencoder using a *deep* neural network, we need to use more layers as hidden layers where the output of each layer is the input for the next layer. In other words, the procedure and calculations explained above will be followed for more than one time to the result of each implementation. The input data will train the first hidden layer, and the output layer of the first hidden layer will serve as the input of the second hidden layer. We iterate these steps based on the number of hidden layers considered for the autoencoder. We use this deep neural network, which serves as the sparse autoencoder, to extract deep features from the set of retrieved attributes of a product category.

Another advantage of using this approach is that the number of features in the final output layer can be less than the number of attributes from the users-attributes matrix. Having a lower dimensionality can speed-up the learning process when the predictive model is dealing with a large dataset. Besides, the model can potentially reduce the deficiencies caused by ambiguity and redundancy in the set of attributes. These characteristics of the deep neural network significantly enhance the quality of the recommendation list for the users.

C. COLLABORATIVE FILTERING

The last step in generating recommendations is using the MF model to predict ratings. First, we update the user profile based on the users-deep features matrix obtained from the deep neural network. Conventional collaborative filtering models only use user-item or user-attributes matrix to generate a recommendation list. In order to use both users-items and users-deep features information, we employ the approach developed by Ricci *et al.* [38]. In this method, we should find the target user neighborhood N_u based on the similarity between the target user and other users using Equation (4.6).

$$sim_{u,v} = \frac{\hat{X}_u, \hat{X}_v}{||\hat{X}_u||||\hat{X}_v||} \tag{6}$$

Having the similarity matrix between users, we can predict the rating of the target user using a weighted average of ratings from the neighbor users using Equation (4.7).

$$S_{u,i} = \sum_{v \in N_u} (\pi_{UI} Y)_{v,i} \tag{7}$$

The final and easy step is to sort the predicted ratings for items and generate the list of the recommendations according to the size of the list, n. Please note that using this approach, we are exploiting the ternary relation between users-attributes-items [8].

IV. EXPERIMENTAL RESULTS

A. DATASET

In this research, we use the Amazon Review dataset [39]. This dataset contains 142.8 million reviews on the Amazon products between May 1996 and July 2014 along with users profile and item metadata. The dataset includes the ID of the reviewer, the ID of the product (ASIN), name of the reviewer, helpfulness rating of the review, text of the review, rating of the product, a summary of the review, and time of the review. It also has the name of the product, price in US dollars, related products, sales rank information, brand name, and the list of the categories of the product. Table 1 presents the statistics of the Amazon Reviews dataset separated by each product category [15]. Also, Figure 3 demonstrates the sparsity of the reviews in the dataset where the percentage for a product category indicates the percentage of the users with no more than three ratings [17]. This sparsity can reduce the performance of recommender systems drastically. On average, there is roughly an average of 120 words in each review.

As the dataset preprocessing, there are many users without having any written review. We removed these observations from the dataset. For processing the data and implementing the proposed methodology, we used Python 3. As the cleaning up step, we removed punctuations and stop-words

TABLE 1. General statistics of the amazon review dataset.

Dataset	Users($x10^3$)	Items($x10^3$)	Reviews($x10^3$)	Words($x10^3$)	Words per Review	Reviews per Item
Arts	24	4	28	2,007	71.73	6.64
Jewelry	41	19	59	3,101	52.9	3.12
Industrial Scientific	30	23	137	6,920	50.5	6.06
Watches	62	10	68	5,437	79.53	6.62
Cell Phones and Accessories	68	7	79	7,568	95.88	10.61
Musical Instruments	67	14	85	7,442	87.14	6.02
Software	68	11	95	11,013	115.82	8.46
Gourmet Foods	113	23	155	10,543	68.18	6.59
Office Products	110	14	138	11,206	81.16	9.71
Automotive	133	48	189	13,250	70.21	3.97
Patio	167	20	206	17,291	83.83	10.56
Pet Supplies	160	18	217	18,684	86.03	12.39
Beauty	168	29	252	17,890	70.97	8.69
Shoes	74	48	390	23,604	60.54	8.05
Kindle Store	116	4	161	21,533	133.92	36.78
Clothing and Accessories	129	66	582	34,267	58.89	8.77
Health	312	40	429	33,277	77.61	10.84
Toys and Games	291	54	436	35,034	80.35	8.13
Tools and Home Improvement	284	51	409	34,591	84.47	8.03
Sports and Outdoors	329	68	511	38,899	76.12	7.48
Video Games	229	21	464	55,532	119.77	22.05
Home and Kitchen	645	79	992	81,923	82.6	12.55
Amazon Instant Video	313	22	718	88,958	123.96	32.32
Electronics	811	82	1,242	124,065	99.91	15.13
Music	1,135	557	6,396	774,791	121.13	11.49
Movies and TV	1,224	213	7,850	997,262	127.04	36.88
Books	2,589	929	12,886	1,613,604	125.22	13.87
All Categories	**6,644**	**2,441**	**34,687**	**4,053,796**	**116.87**	**14.21**

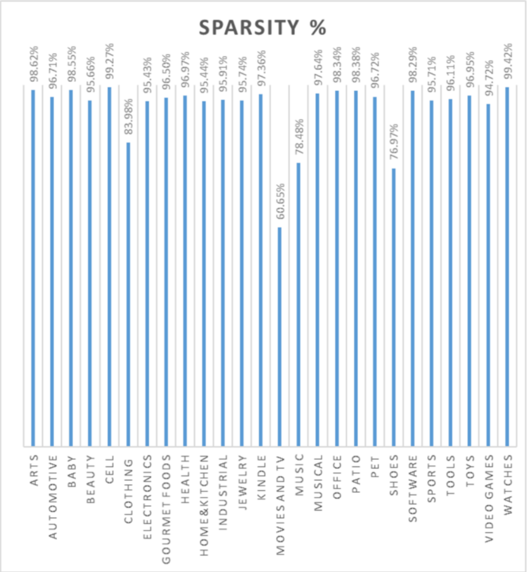


FIGURE 3. Sparsity of the Amazon Reviews Dataset.

using NTLK stopwords. Words that have appeared in the review corpus of a product category only once are most likely irrelevant; thus, we eliminated these words as well. Using the rest of tokens, we construct our preliminary dictionary of attributes. Note that we create a separate dictionary for each product category.

For the training part, we selected 80% of the dataset for training, 10% for validation, and 10% for testing, randomly. Furthermore, we selected 25 topics and 40 words for each topic when we applied LDA to each category review corpus. Then, we used the association rule mining technique to extract frequent itemsets from unique words obtained after the LDA step. Finally, we matched the reviews of each user with the set of extracted words and constructed the users-attributes matrix. For the rest of the parameters required to apply the deep neural network feature extractor and matrix factorization method, the hyperparameters are as follows.

- The number of hidden layers is 2/3.
- The number of neurons in the first layer is 1000.
- The number of neurons in the second layer is 800.
- Average activation is 0.2.

In order to obtain these values, we changed one hyperparameter in a reasonable range to find a value that provides the best performance while we fix other hyperparameters only on one of the product categories. For the number of hidden layers, both two and three hidden layers show high performance.

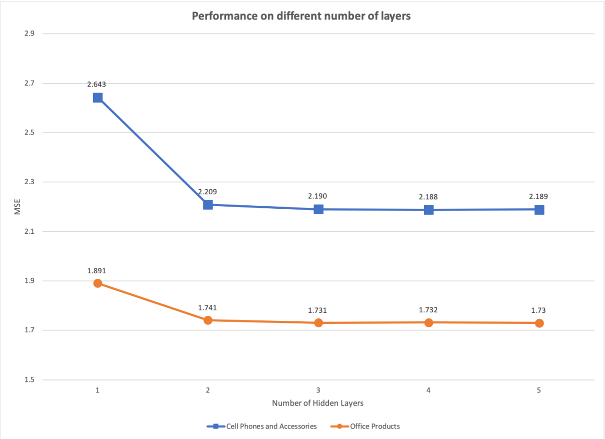


FIGURE 4. Analysis of MSE based on the number of hidden layers.

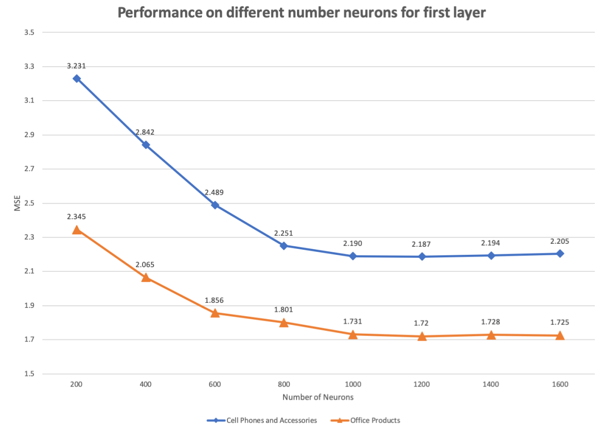


FIGURE 5. Analysis of MSE based on the number of neurons in the first hidden layers.

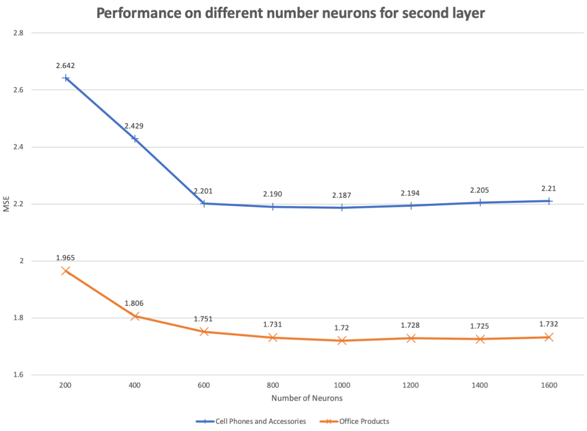


FIGURE 6. Analysis of MSE based on the number of neurons in the second hidden layers.

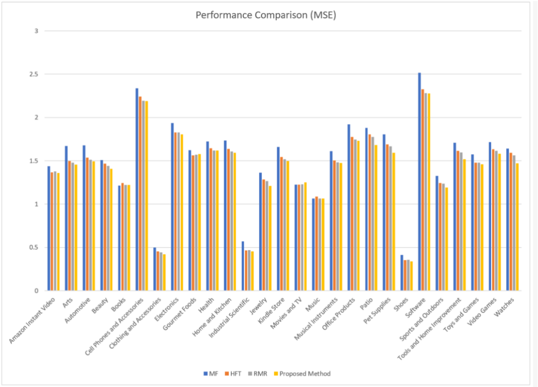


FIGURE 7. Comparing the MSE from the proposed method and the baselines.

During the performance evaluation, we performed both on a product category to find the best results. Figures 4, 5, and 6 demonstrate the MSE on two product datasets used to tune the hyperparameters. As you can see, MSE is not improving after using three hidden layers. MSE stops improving significantly at 1000 and 800 neurons in the first and second hidden layers, respectively.

B. BASELINE METHODS

We compare the performance of our model with three other state-of-the-art models, including MF, the Hidden Topics and Factors (HTF), and the Ratings Meet Review (RMR). The following is the explanation of these models.

- **MF** is the standard and widely used matrix factorization model. We consider the model proposed and described in [40]. This model uses the ratings of the user in generating recommendations, and the written part of the customer's feedback is not incorporated.
- **HTF** is a model proposed by [16] that incorporates the review text with the stochastic topic distribution modeling which can be applied either on users or items. It also employs matrix factorization to deal with the ratings.
- **RMR** is a hybrid model constituted of content-based filtering and collaborative filtering suggested by [15]. This model tries to exploit the information from reviews and improve the recommendation list accuracy across various classes of datasets. They tried to address the cold-start problem with collapsed Gibbs sampler for learning the model parameters.

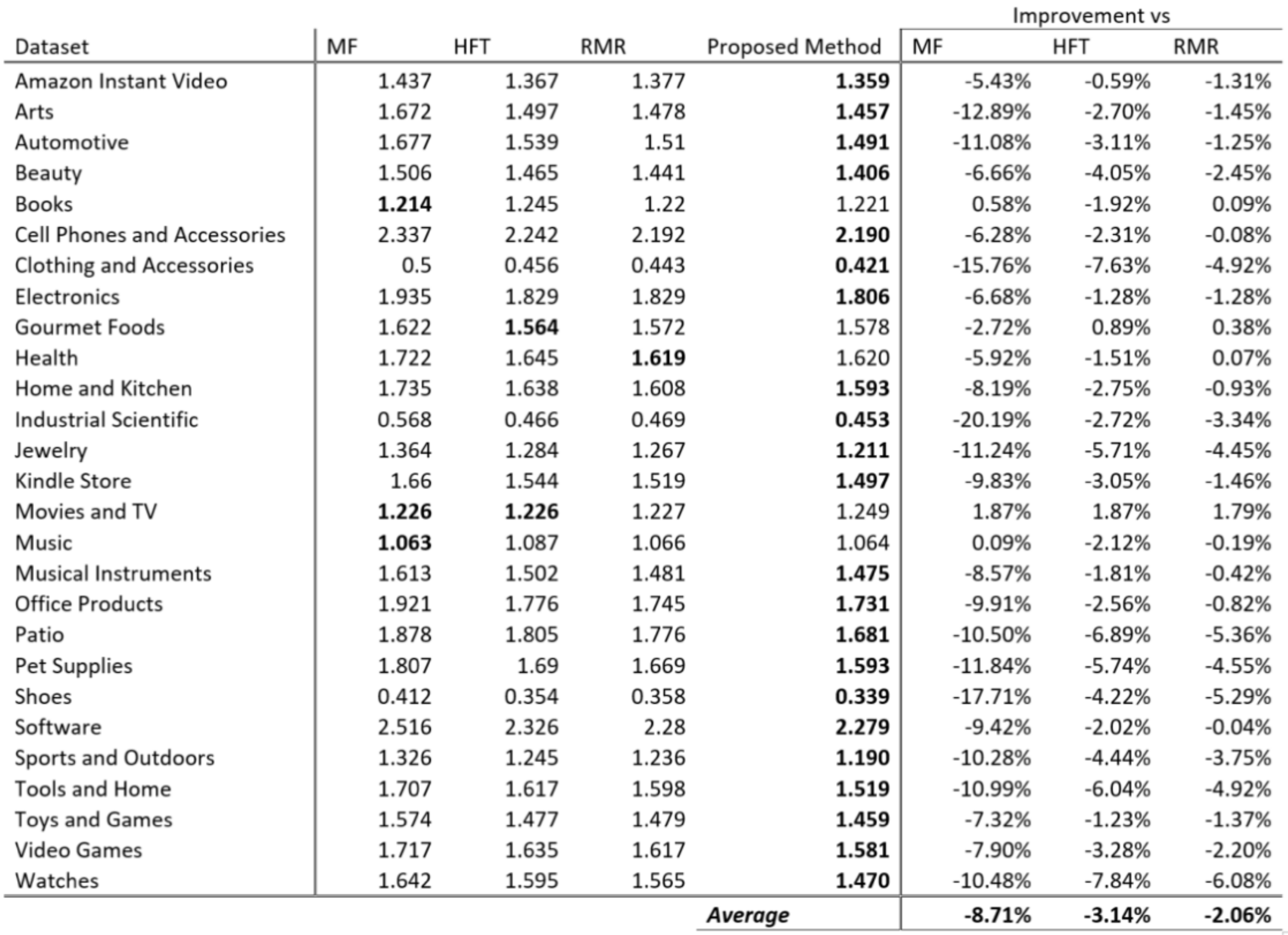
For each product category, we report the performance of each model against our model. We consider Mean Squared Error (MSE) for evaluation of these models against the proposed approach.

C. EVALUATION

We applied the proposed deep feature extractor method to all the product categories datasets and obtained the best MSE for our model and compared these results with our baselines. Figure 7 and Table 2 demonstrate the results.

As you can see in these figures, the proposed method performs better for most of the product categories. Comparing to the MF model, our method is capable of predicting ratings with an average of 8.71% improvement, in some cases up to 20.19%. In three cases, MF shows better performance,

TABLE 2. MSE of the proposed method vs baselines and the percentage of the improvement.



					Improvement vs		
Dataset	MF	HFT	RMR	Proposed Method	MF	HFT	RMR
Amazon Instant Video	1.437	1.367	1.377	**1.359**	-5.43%	-0.59%	-1.31%
Arts	1.672	1.497	1.478	**1.457**	-12.89%	-2.70%	-1.45%
Automotive	1.677	1.539	1.51	**1.491**	-11.08%	-3.11%	-1.25%
Beauty	1.506	1.465	1.441	**1.406**	-6.66%	-4.05%	-2.45%
Books	**1.214**	1.245	1.22	1.221	0.58%	-1.92%	0.09%
Cell Phones and Accessories	2.337	2.242	2.192	**2.190**	-6.28%	-2.31%	-0.08%
Clothing and Accessories	0.5	0.456	0.443	**0.421**	-15.76%	-7.63%	-4.92%
Electronics	1.935	1.829	1.829	**1.806**	-6.68%	-1.28%	-1.28%
Gourmet Foods	1.622	**1.564**	1.572	1.578	-2.72%	0.89%	0.38%
Health	1.722	1.645	**1.619**	1.620	-5.92%	-1.51%	0.07%
Home and Kitchen	1.735	1.638	1.608	**1.593**	-8.19%	-2.75%	-0.93%
Industrial Scientific	0.568	0.466	0.469	**0.453**	-20.19%	-2.72%	-3.34%
Jewelry	1.364	1.284	1.267	**1.211**	-11.24%	-5.71%	-4.45%
Kindle Store	1.66	1.544	1.519	**1.497**	-9.83%	-3.05%	-1.46%
Movies and TV	**1.226**	**1.226**	1.227	1.249	1.87%	1.87%	1.79%
Music	**1.063**	1.087	1.066	1.064	0.09%	-2.12%	-0.19%
Musical Instruments	1.613	1.502	1.481	**1.475**	-8.57%	-1.81%	-0.42%
Office Products	1.921	1.776	1.745	**1.731**	-9.91%	-2.56%	-0.82%
Patio	1.878	1.805	1.776	**1.681**	-10.50%	-6.89%	-5.36%
Pet Supplies	1.807	1.69	1.669	**1.593**	-11.84%	-5.74%	-4.55%
Shoes	0.412	0.354	0.358	**0.339**	-17.71%	-4.22%	-5.29%
Software	2.516	2.326	2.28	**2.279**	-9.42%	-2.02%	-0.04%
Sports and Outdoors	1.326	1.245	1.236	**1.190**	-10.28%	-4.44%	-3.75%
Tools and Home	1.707	1.617	1.598	**1.519**	-10.99%	-6.04%	-4.92%
Toys and Games	1.574	1.477	1.479	**1.459**	-7.32%	-1.23%	-1.37%
Video Games	1.717	1.635	1.617	**1.581**	-7.90%	-3.28%	-2.20%
Watches	1.642	1.595	1.565	**1.470**	-10.48%	-7.84%	-6.08%
				Average	**-8.71%**	**-3.14%**	**-2.06%**

including Books, Movies and TV, and Music. For Books and Music categories, the model is off only by less than 1 percent, which implies that the performance of the model is close to the best performance for these two categories. A similar situation is happening between the HFT model and the proposed approach. Our deep neural network model beats the HFT model predictions for most of the cases. On average, our model improves the predictions by 3.14%. For the only two cases that our model is not performing better, the performance is close enough. In the worst case, which is the Movies and TV product category, MF and HFT models performed only 1.87% better than our model. Finally, our model outperforms the RMR method by 2.06% on average. Figs 8, 9, and 10 illustrate the improvements made by our model compared to MF, HFT, and RMR baseline models, respectively.

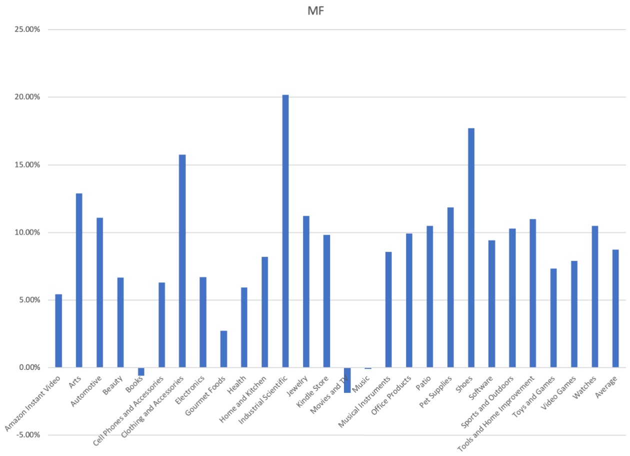


FIGURE 8. Proposed model improvement compared to MF.

We investigated the product categories that our model is less accurate comparing the baselines. We suggest that the reason for this minor inaccuracy is the existence of more non-technical terms than technical attributes in the users-attributes matrix. For example, in the category of Industrial Scientific, we have a significant improvement between the MF model and the proposed method equal to 20.19%. Moreover, the performance improvement is higher for other categories that customers talk more about product attributes such as Tools and Clothing. The superiority of the proposed

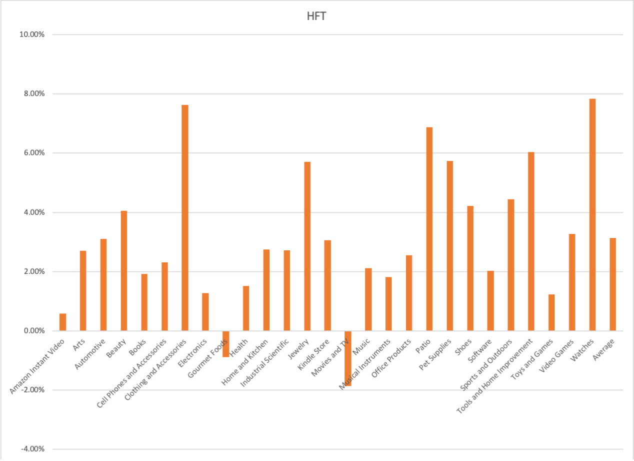


FIGURE 9. **Proposed model improvement compared to HFT.**

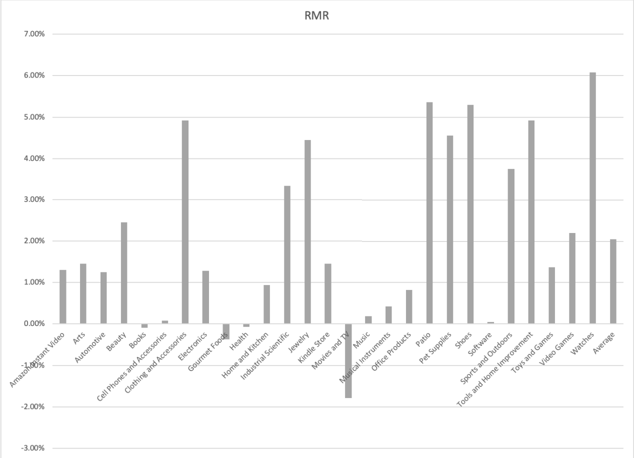


FIGURE 10. **Proposed model improvement compared to RMR.**

model is the fact that the deep neural feature extractor retrieves the deep features and models the extracted words in a way that makes the users-attributes more informative, hence, extracting non-trivial relation between users based on the reviews they write. Our model can benefit e-commerce businesses through increasing revenue and customer satisfaction as recommendation plays a crucial role in real systems.

V. CONCLUSION

In this paper, we proposed a deep neural network approach to incorporate customer reviews in developing recommender systems. In our proposed model, we use Latent Dirichlet Allocation to extract attributes related to each product category. Then, we used association rule mining to use frequent terms in the dataset. Having the set of extracted attributes, we constructed a users-attributes matrix. This matrix suffers from a sparsity. To deal with this challenge, we proposed a deep neural network solution that transforms the sparse users-attributes matrix into a dense users-deep features matrix, as an unsupervised learning tool. Finally, we used matrix factorization to predict ratings. We evaluated the performance of our model using the Amazon Review dataset, which is the largest dataset for customer reviews categorized for each product category. We also compared the MSE of our model with three baseline models from the literature, including MF, HFT, and RMR models. Our model outperforms these state-of-the-art models for most datasets.

For the future research directions, we are going to apply a deep neural network as the predicting model instead of the deep neural encoder and the matrix factorization method to improve the predictive power of our approach. Besides, we will investigate the application of other natural language processing tools for the construction of users-attributes matrix and compare their performance with current research.

REFERENCES

[1] M. Darabi and N. Tabrizi, "An ontology-based framework to model user interests," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Dec. 2016, pp. 398–403.

[2] H. Yamaba, M. Tanoue, K. Takatsuka, N. Okazaki, and S. Tomita, "On a serendipity-oriented recommender system based on folksonomy," *Artif. Life Robot.*, vol. 18, nos. 1–2, pp. 89–94, 2013.

[3] Z.-K. Zhang, T. Zhou, and Y.-C. Zhang, "Tag-aware recommender systems: A state-of-the-art survey," *J. Comput. Sci. Technol.*, vol. 26, no. 5, p. 767, 2011.

[4] B. Horsburgh, S. Craw, and S. Massie, "Learning pseudo-tags to augment sparse tagging in hybrid music recommender systems," *Artif. Intell.*, vol. 219, pp. 25–39, Feb. 2015.

[5] E. Baralis, L. Cagliero, T. Cerquitelli, S. Chiusano, P. Garza, D. Antonelli, G. Bruno, and N. A. Mahoto, "Personalized tag recommendation based on generalized rules," *ACM Trans. Intell. Syst. Technol.*, 2019. [Online]. Available: https://iris.polito.it/retrieve/handle/11583/2570938/29390/AntonellietAl.pdf

[6] A. K. Sahu, P. Dwivedi, and V. Kant, "Tags and item features as a bridge for cross-domain recommender systems," *Procedia Comput. Sci.*, vol. 125, pp. 624–631, Jan. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S187705091732848X

[7] H. Kim and H.-J. Kim, "A framework for tag-aware recommender systems," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 4000–4009, 2014.

[8] Y. Zuo, J. Zeng, M. Gong, and L. Jiao, "Tag-aware recommender systems based on deep neural networks," *Neurocomputing*, vol. 204, pp. 51–60, Sep. 2016.

[9] H.-N. Kim, A. Alkhaldi, A. El Saddik, and G.-S. Jo, "Collaborative user modeling with user-generated tags for social recommender systems," *Expert Syst. Appl.*, vol. 38, no. 7, pp. 8488–8496, Jul. 2011.

[10] Y. Shi, M. Larson, and A. Hanjalic, "Mining contextual movie similarity with matrix factorization for context-aware recommendation," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 1, pp. 16:1–16:19, Feb. 2013.

[11] H. Zhang, Z.-J. Zha, Y. Yang, S. Yan, Y. Gao, and T.-S. Chua, "Attribute-augmented semantic hierarchy: Towards a unified framework for content-based image retrieval," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 11, no. 1s, p. 21, 2014.

[12] R. He and J. McAuley, "VBPR: Visual Bayesian personalized ranking from implicit feedback," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1–7.

[13] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 287–296.

[14] X. Wang, X. He, L. Nie, and T.-S. Chua, "Item silk road: Recommending items from information domains to social users," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2017, pp. 185–194.

[15] G. Ling, M. R. Lyu, and I. King, "Ratings meet reviews, a combined approach to recommend," in *Proc. 8th ACM Conf. Recommender Syst.*, 2014, pp. 105–112.

[16] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proc. 7th ACM Conf. Recommender Syst.*, 2013, pp. 165–172.

[17] Y. Tan, M. Zhang, Y. Liu, and S. Ma, "Rating-boosted latent topics: Understanding users and items with ratings and reviews," in *Proc. IJCAI*, vol. 16, 2016, pp. 2640–2646.

[18] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 448–456.
[19] L. Zhang, Y. Zhou, X. Duan, and R. Chen, "A hierarchical multi-input and output bi-GRU model for sentiment analysis on customer reviews," *IOP Conf. Ser., Mater. Sci. Eng.*, 2018, vol. 322, no. 6, Art. no. 062007.
[20] A. Bagheri, M. Saraee, and F. De Jong, "Care more about customers: Unsupervised domain-independent aspect detection for sentiment analysis of customer reviews," *Knowl.-Based Syst.*, vol. 52, pp. 201–213, Nov. 2013.
[21] Q. Sun, J. Niu, Z. Yao, and H. Yan, "Exploring eWOM in online customer reviews: Sentiment analysis at a fine-grained level," *Eng. Appl. Artif. Intell.*, vol. 81, pp. 68–78, May 2019.
[22] P. K. Sari, A. Alamsyah, and S. Wibowo, "Measuring e-Commerce service quality from online customer review using sentiment analysis," *J. Phys., Conf. Ser.*, vol. 971, no. 1, 2018, Art. no. 012053.
[23] P. P. Ładyżyński and P. Grzegorzewski, "A recommender system based on customer reviews mining," in *Proc. Int. Conf. Artif. Intell. Soft Comput.* Cham, Switzerland: Springer, 2014, pp. 512–523.
[24] L. Qiu, S. Gao, W. Cheng, and J. Guo, "Aspect-based latent factor model by integrating ratings and reviews for recommender system," *Knowl.-Based Syst.*, vol. 110, pp. 233–243, Oct. 2016.
[25] Z. Cheng, Y. Ding, L. Zhu, and M. Kankanhalli, "Aspect-aware latent factor model: Rating prediction with ratings and reviews," in *Proc. Conf. World Wide Web*, 2018, pp. 639–648.
[26] K. Bauman, B. Liu, and A. Tuzhilin, "Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 717–725.
[27] E. Fathi and B. M. Shoja, "Deep neural networks for natural language processing," in *Computational Analysis and Understanding of Natural Languages: Principles, Methods and Applications*, vol. 38. Amsterdam, The Netherlands: Elsevier, 2018, p. 229.
[28] M. M. Susan and S. David, "What makes a helpful review? A study of customer reviews on Amazon.com," *MIS Quart.*, vol. 34, no. 1, pp. 185–200, 2010.
[29] H. Liu, J. He, T. Wang, W. Song, and X. Du, "Combining user preferences and user opinions for accurate recommendation," *Electron. Commerce Res. Appl.*, vol. 12, no. 1, pp. 14–23, 2013.
[30] S.-M. Kim and E. Hovy, "Determining the sentiment of opinions," in *Proc. 20th Int. Conf. Comput. Linguistics*, 2004, p. 1367.
[31] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[32] Z. Zhang, D. D. Zeng, A. Abbasi, J. Peng, and X. Zheng, "A random walk model for item recommendation in social tagging systems," *ACM Trans. Manage. Inf. Syst.*, vol. 4, no. 2, p. 8, 2013.
[33] M. A. Ghazanfar and A. Prügel-Bennett, "Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems," *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3261–3275, 2014.
[34] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Adv. neural Inf. Process. Syst.*, 2007, pp. 801–808.
[35] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng, "On optimization methods for deep learning," in *Proc. 28th Int. Conf. Int. Conf. Mach. Learn.*, 2011, pp. 265–272.
[36] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[37] G. E. Hinton, "A practical guide to training restricted boltzmann machines," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 599–619.
[38] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2011, pp. 1–35.
[39] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *Proc. 25th Int. Conf. World Wide Web*, 2016, pp. 507–517.
[40] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264.

BABAK MALEKI SHOJA is currently pursuing the degree majoring in computer science and economics with East Carolina University. His current research interests include recommender systems, folksonomy, and the application of neural networks and deep learning in recommender systems, image recognition and processing, and natural language processing.

NASSEH TABRIZI is currently the Graduate Program Director of the Computer Science Department and the Founder and the Director of the Software Engineering Program, East Carolina University. His research interests include big data analytics, recommender systems, image processing, software engineering, and computer vision.