Social Recommendation Using Deep Auto-encoder and Confidence Aware Sentiment Analysis

Lamia Berkani[1,2](✉), Abdelhakim Ghiles Hamiti[2], and Yasmine Zemmouri[2]

[1] Laboratory for Research in Artificial Intelligence (LRIA), USTHB University, 16111 Bab Ezzouar, Algiers, Algeria
lberkani@usthb.dz

[2] Department of Artificial Intelligence and Data Sciences, Faculty of Informatics, USTHB University, 16111 Bab Ezzouar, Algiers, Algeria

Abstract. The development of online social networks has attracted increasing interest in social recommendation. On the other hand, recommender systems based on deep learning and sentiment analysis techniques are currently widely used to solve the problem of data sparsity. However, only a few attempts have been made in social-based recommender systems. This article focuses on this issue and proposes a novel hybrid approach named CASA-SR (Confidence Aware Sentiment Analysis-based Deep Social Recommendation). Our approach exploits sentiment analysis by detecting fake reviews and combines predictions generated by collaborative and content-based filtering. A neural architecture has been adopted using an auto-encoder and a multilayer perceptron neural network. Moreover, our approach integrates social information, including users' trust (credibility and similarity degrees). Experimental results conducted on different datasets showed significant improvements in recommendation performance according to the state-of-the-art work.

Keywords: Social recommendation · hybrid sentiment analysis · fake reviews · deep learning · auto-encoder · MLP

1 Introduction

The development of online social media has attracted increasing interest in social recommendation. Previous work demonstrated that the integration of social information, as auxiliary information, can enhance the performance of traditional recommender systems. Several works have integrated social information with collaborative filtering-based methods [1–3].

On the other hand, deep learning techniques have been recently applied in recommender systems to solve the cold start and data sparsity problems and further enhance the recommendation accuracy and performance. Current models mainly use deep neural networks to learn user preferences on items for recommendations. However, only few initiatives have been conducted in social-based recommendation field [4–7].

M. Mosbah et al. (Eds.): MEDI 2023, LNCS 14396, pp. 142–155, 2024.
https://doi.org/10.1007/978-3-031-49333-1_11

To overcome the rating data sparseness, users' comments are being used for rating prediction. These reviews can express users' overall satisfaction on the items through their preferred, non-preferred or neutral opinions. Several research works are applying sentiment analysis (SA) in recommender systems [8]. Some works are based on hybrid deep-learning models. For instance Dang et al. [9] integrated a hybrid SA model into the collaborative filtering, by combining the CNN and LSTM models in different orders. This approach was proposed in the context of social networks, but did not take social information into account in the recommendation process. Berkani and Boudjenah [5], integrated a hybrid SA model to a deep neural network model including social information with friendship and trust features. However this work didn't take into consideration fake reviews. Recommendation systems are vulnerable to intrusions (due to a lack of security) or to the sharing of fake information (many malicious users add misleading information). Thus, recommendation systems based on SA can be manipulated or disrupted by the presence of fake reviews. Recently, some studies have proposed approaches for detecting fake reviews [10, 11]. However, to the best of our knowledge, no work proposed in a social context has combined a hybrid SA model with fake reviews detection along with social information formalization.

In this article, we propose a novel social-based recommender model using a confidence aware hybrid sentiment model to improve the user-item rating matrix by predicting missing ratings and correcting inconsistent values. By exploiting the updated matrix, our system generates predictions using a hybrid recommendation algorithm which combines social information with collaborative and content-based filtering algorithms using an auto-encoder and an MLP network, respectively. Extensive experiments conducted on two datasets demonstrated the effectiveness of our model compared to the state-of-the-art approaches and baselines.

The remainder of this article is organized as follows: Sect. 2 presents some related work. Section 3 and Sect. 4 present respectively the conception of our approach with the associated experiments. Section 5 highlights the most important contributions of this work and proposes some future perspectives.

2 Related Work

The development of online social media has favored the expansion of social recommendation, where several researchers are interested in proposing approaches that integrate social information. Different research works included social trust in recommender systems [2, 3].

With the latest achievements and the great potential for learning effective representations, DL models are being exploited recently in recommender systems. He et al. [12] proposed the widely used Neural Collaborative Filtering algorithm (NCF) using a Generalized Matrix Factorization (GMF) and an MLP to model the linear and nonlinear relationship between users and items. Berkani et al. [13, 14] proposed the Neural Hybrid Filtering model (NHF) based on GMF and Hybrid MLP. To predict ratings in recommender systems, Rama et al. [15] proposed a discriminative model that integrates features from auto-encoders with embeddings in a deep neural network.

On the other hand, the flexibility and accessibility of social networks has enabled millions of people to subscribe and post their comments on these platforms, expressing their preferences and/or feelings about items. Given that people often choose comments rather than numerical ratings, researchers tried to get around the problem of rating data sparsity by leveraging textual reviews. These works are using sentiment analysis (SA) to predict their current preferences for given items [16–18]. Some of the proposed approaches are based on hybrid DL models [5, 9]. This combination has significantly improved the recommender system's performance.

However, despite much attention paid to DL and SA in recommender systems, the state of the art shows that only few works have used DL in social-based recommender systems. For instance, Bathla et al. [7] proposed AutoTrustRec, a recommender system with direct and indirect trust and DL using auto-encoder. However, this work did not exploit the users' reviews for improving the recommendation performance. Berkani et al. [4] developed the SNHF model, incorporating social information in the NHF. However this work has only considered friendship and trust degree that has been calculated with MoleTrust algorithm [19].

The review of related work demonstrates the significant results achieved by the application of DL techniques and sentiment models in recommender systems. Proposed work in a social context has considered only a few features to model social information, including friendship and trust (direct and/or indirect). While other important factors can be considered such as the credibility and the influence of users in social networks. Moreover, as reviews often contain fake good or fake bad information, some studies focused on fake reviews detection. Li et al. [10] exploited the interactivity of review information and used the confidence matrix to measure the relationship between rating outliers and misleading reviews. Birim et al. [11] focused on detecting fake reviews through topic modelling. Similarly, we will propose in this article a novel approach using DL and confident aware hybrid sentiment analysis by detecting fake reviews. We will also focus on modeling other features of social information.

3 Our Approach

We have proposed an approach called CASA-SR (Confidence Aware Sentiment Analysis-based Deep Social Recommendation) including four main modules, as illustrated in Fig. 1:

1. Hybrid sentiment analysis module with fake reviews detection (enabling the construction of a confidence matrix). This module updates the rating matrix.
2. Collaborative filtering based on neural architecture using an auto-encoder. This module uses the updated rating matrix considering user by user (line by line), and generates an output prediction.
3. Content-based filtering: This module uses an MLP network to generate a prediction based on user and item features.
4. Hybrid recommendation: a final neural layer combines users' social information with the results generated from the collaborative and content-based recommendations.

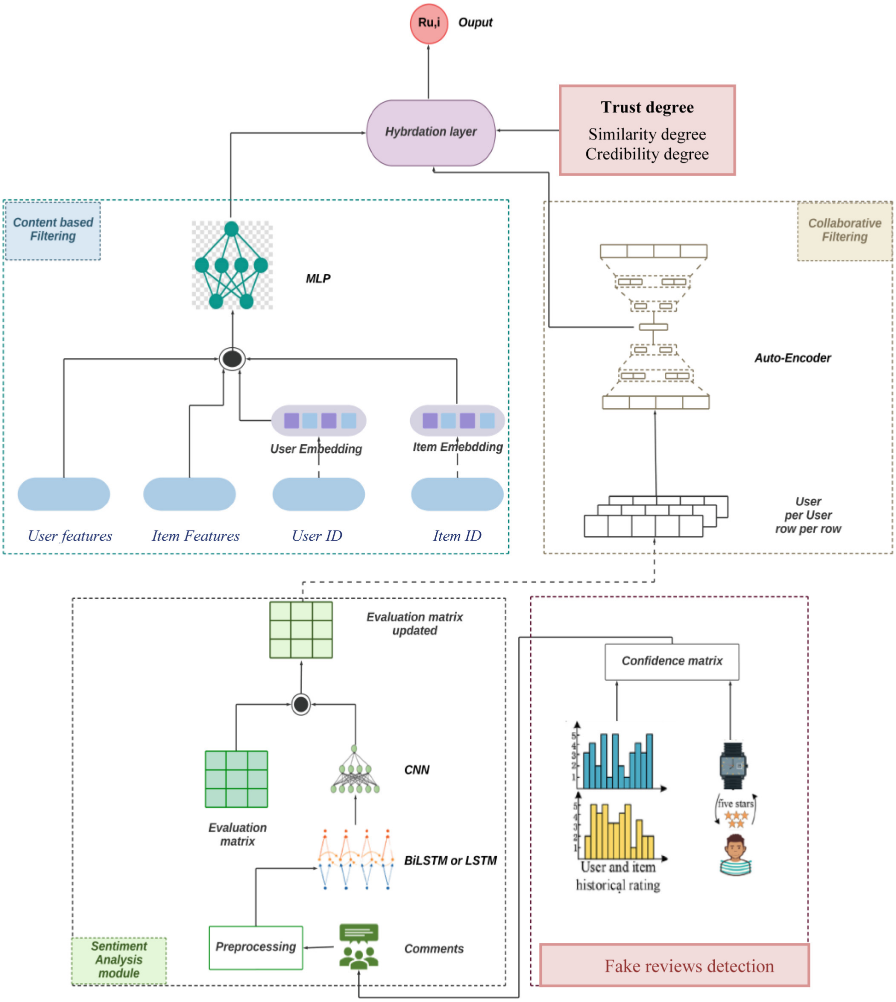

Fig. 1. Overall architecture of the CASA-SR approach.

3.1 Confidence Aware Sentiment Analysis

This module converts comments into a numerical score (from 1 to 5) using a sentiment analysis process based on a combination of an LSTM / Bi-LSTM recurrent neural network and a CNN convolutional neural network, then applies normalization to obtain values between 1 and 5. Figure 2 illustrates this process:

After extraction of the textual data (comments associated with each user's opinion about an item), a set of pre-processing operations is carried out, including: tokenization, used to fragment the comment into sub-words; cleaning (i.e. remove stop words, taking care to retain adjectives and remove all punctuation marks); encoding and padding to homogenize the lengths of the resulting sequences; and generation of embeddings using the BERT model (Bidirectional Encoder Representations from Transformers) [20].

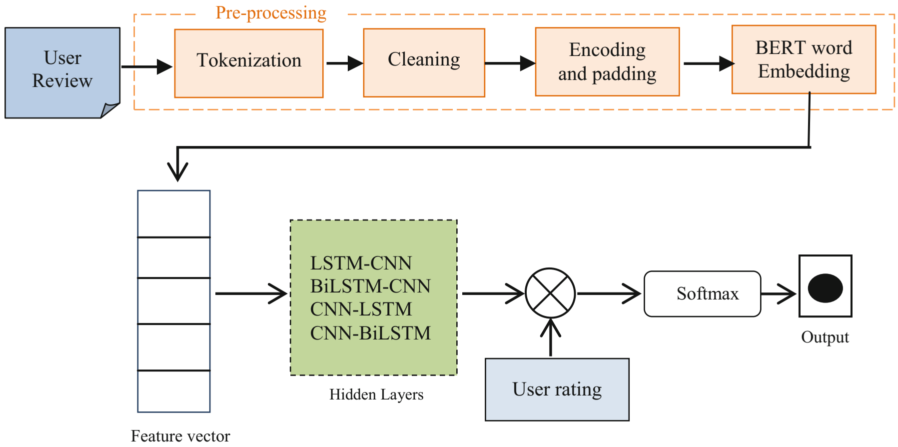

Fig. 2. Structure of the hybrid sentiment analysis module

BERT encodes words taking into account the global context using Attentional Transformers, which allows it to understand the complex semantic relationships between words. This model generates sequence vectors representing the tokens of comments, with the aim of matching words having similar meanings with similar vectors. We considered different combination variants of LSTM (Bi-LSTM) with CNN: LTSM-CNN; CNN-LSTM; Bi-LSTM-CNN; CNN-Bi-LSTM. The generated sentiment-based rating is then combined with the user's initial rating, obtaining the final rating, according to the following formula [21].

$$score_f = \alpha * score_s + (1 - \alpha) * score_R \tag{1}$$

where: score-f is the final score; score-s is the score resulting from sentiment analysis; score-r is the original rating; and α represents the balancing factor between the two values.

In addition to the processing carried out on user comments, we construct a confidence matrix (Q) as in [10]. This matrix can be seen as a regularization that adjusts the reviews. For convenience, the values of the matrix are between 0 and 1. The value of each element of the matrix is calculated by the following confidence degree function [10]:

$$Q_{ij} = F(R_{ij}, \beta) \begin{cases} e^{-Relu\left(\left|\sqrt{\sum_{p \neq j, R_{ip \leq 3}} R_{ip} - R_{ij}}\right| - \beta\right) - Relu\left(\left|\sqrt{\sum_{q \neq i, R_{ip \leq 3}} R_{ip} - R_{ij}}\right| - \beta\right)}, & R_{ij} \leq 3 \\ e^{e^{-Relu\left(\left|\sqrt{\sum_{p \neq j, R_{ip > 3}} R_{ip} - R_{ij}}\right| - \beta\right) - Relu\left(\left|\sqrt{\sum_{q \neq i, R_{ip > 3}} R_{ip} - R_{ij}}\right| - \beta\right)}}, & R_{ij} > 3 \end{cases} \tag{2}$$

where: β: is the deviation rate; Q: is the confidence matrix; R: is the rating matrix;

The process of calculating confidence values shows that when a user assigns a high (or low) rating to an item, this rating deviates considerably from the average of the user's previous ratings and the average of the item's previous ratings. Therefore, the corresponding comments are given a low weight, and a high probability of being considered as false comments.

3.2 Collaborative Filtering

This module takes as input the ratings matrix generated from the sentiment analysis module and processes it via an auto-encoder (AE) neural network. The objective of the AE is to learn a latent representation of the rating matrix by taking this matrix as input line by line and compressing it into a lower-dimensional latent representation through several layers of neurons, one smaller than the previous one. The size of a given layer is the size of the previous layer divided by 2.

Next, the decoder considers this latent representation and transforms it into a reconstruction of the input matrix line via the same number of layers as the encoder, but in this case, one larger than the previous one. This means that the size of a layer is the size of the previous layer multiplied by 2. During the training process, the auto-encoder attempts to minimize the difference between the rows of the original evaluation matrix and their reconstructions. Once the AE has been trained, the latent representation is used to generate personalized item recommendations.

3.3 Content-Based Filtering

For content-based filtering, we add the descriptive item and user information, then concatenate them with the embeddings of their respective identifiers. These vectors are then given as input to a multilayer MLP neural network structure, which will generate the prediction as output.

We have distributed the number of neurons per layer in decreasing order. This number is inversely proportional to the number of nodes per layer. We also set the number of nodes for the last layer, called 'predictive factors'.

3.4 Social Information Modeling

Social information between users will be modeled by considering the concept of trust. The degree of trust between two users will be used to select the closest people to a given user. It is necessary to filter the user's trusted contacts according to their proximity to the user (it is more likely that a user will seek advice from a person who is closer to him/her than from another person) as well as according to their credibility.

The trust degree between two users will take into consideration the degree of similarity between them and the degree of credibility of the second user:

$$D_{Trust}(u_1, u_2) = \alpha.D_{Similarity}(u_1, u_2) + (1 - \alpha).D_{Credibility}(u_2) \tag{3}$$

where: α is the importance weight between the degree of similarity and credibility.

Similarity Degree. We consider that two users u1 and u2 are similar if they interact with the same items. Similar assessment selection implies that users trust each other. To calculate this degree, we use the cosine similarity measure [5]:

$$D_{Similarity}(u1, u2) = \frac{\sum r_{u1,\mathrm{i}} . r_{u2,\mathrm{i}}}{\sqrt{\sum r_{u1,\mathrm{i}}{}^2} . \sqrt{\sum r_{u2,\mathrm{i}}{}^2}} \tag{4}$$

where: $r_{u1,\mathrm{i}}$: is the evaluation of user u_1 on item i; and $r_{u2,\mathrm{i}}$: is the evaluation of user u_2 on item i;

Credibility Degree. For the calculation of the degree of credibility we have considered the following formula which takes into account the calculation of the fake reviews rate, the competence degree and the participation degree:

$$D_{Credibility}(u) = \delta.D_{Fake}(u) + \lambda.dD_{Competence}(u) + \gamma.D_{Participation}(u) \tag{5}$$

where: γ, λ and δ: are weights expressing a priority, with: $\gamma + \lambda = 1$ and δ is negative.

Fake Reviews Rate. This rate is calculated according to the number of fake reviews posted by a given user *u*, based on the total number of fake reviews. The greater the number of fake reviews posted by a user, the less credible that user is.

$$D_{Fake}(u) = \frac{NbFakeReview(u)}{NbFakeReviewTotal} \tag{6}$$

Competence Degree. We consider a user to be competent if he/she has rated the items "correctly" compared to their average ratings, where the average rating of an item is calculated based on the ratings of all users of the system, according to the following formula [22]:

$$D_{Competence}\left(u, I_j\right) = \frac{\left|r_{u,j} - avg\left(I_j\right)\right|}{k} \tag{7}$$

where: I_j: is the item number j; $r_{u,j}$: represents the evaluation of the user u on the item I_j; and $avg\left(I_j\right)$: is the average evaluation of the item I_j relative to all users of the system.

The degree of competence when considering all the items is calculated as follows [22]:

$$D_{Competence}(u) = \frac{1}{n}.\sum\nolimits_{j=1}^{n} D_{Competence}(u, I_j) \tag{8}$$

where: n represents the number of items.

If a user's opinion is far from the average of other users' opinions, then he/she will lose out in credibility.

Participation Rate. This rate is calculated on the basis of the number of evaluations performed by a user u, according to all the evaluations in the system.

$$D_{Participation}(u) = \frac{\alpha.NbReview(u) + \beta.NbRating(u)}{NbTotalReview + NbTotalRating}$$

where β is the importance degree between Reviews and Ratings.

3.5 Hybrid Recommandation

The hybrid recommendation module combines the results obtained from the collaborative and content-based filtering modules, including social information. The last active layer of the MLP multi-layer neural network, related to the content-based module, will be concatenated to the common layer between the encoder and the decoder of the auto-encoder, related to the collaborative module, as well as to the value obtained from the average preference of the user's trusted persons. This concatenation will be fed as input to an active neural layer of 'SoftMax' function, providing a vector of normalized probabilities representing the probability distribution over the different classes (ranging from 1 to 5). The class with the highest probability will be the predicted score.

4 Experiments

We present in this section the experiments performed on two different datasets. We first performed some preliminary evaluations to set the parameter values. Then we evaluated the SA module by comparing the different hybrid models. Next, we evaluated the hybrid recommendation algorithm, by verifying the contribution of SA, social information and fake reviews detection. Finally, we compared our model with existing related works.

4.1 Datasets

For the training and evaluation of our method, we used two datasets from the Yelp[1] social network. We extracted two data samples related to the "Restaurant" and "Shopping" categories. Table 1 shows the corresponding statistics:

Table 1. Dataset statistics

Dataset	#Users	#Items	#Ratings	#Reviews	Density
Yelp-Shopping	2,935	9,637	61,967	61,967	0.2%
Yelp-Restaurant	4,346	15,588	391,924	391,924	0.57%

For the training of the sentiment module, we used the 'IMDB' dataset comprising 50,000 comments, based on the Internet Movie Database (IMDB). Each comment is associated with a sentiment label, which can be either "positive" or "negative". This dataset is balanced in terms of the number of positive and negative comments.

4.2 Evaluation Metrics

We used the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) evaluation metrics. MAE and RMSE have been used as they are the most popular predictive metrics to measure the closeness of predictions relative to real scores:

$$MAE = \frac{\sum_{u,i \in \Omega} |r_{u,i} - p_{u,i}|}{|\Omega|} \tag{9}$$

$$RMSE = \sqrt{\frac{\sum_{u,i \in \Omega} (r_{u,i} - p_{u,i})^2}{|\Omega|}} \tag{10}$$

where:

Ω: set of test assessments and $|\Omega|$ indicates the cardinality of the set Ω;
$r_{u,i}$: is the rating given by the user u on the item i; and.
$p_{u,i}$: is the rating prediction of the user u on the item i.

[1] https://www.yelp.com.

4.3 Baselines

We have compared our model with the following related works:

- **PMF:** the probabilistic matrix factorization approach, widely used in CF.
- **SVD++:** the matrix factorization technique that exploits the concept of "Singular value decomposition" to improve the performance of the CF algorithm.
- **SocialMF:** an approach that enriches the PMF model by integrating social information [23].
- **DeepCoNN:** Deep Cooperative Neural Networks, a DL-based recommendation technique that exploits the reviews to generate recommendations [24].

4.4 Experimental Parameters

We used the Python language version 3.8.5, exploiting several libraries (e.g. Tensorflow, Keras, Scikit-learn, Pandas, NumPy). To train our models, we used the Google Colab platform, offering the following features: 2 GB RAM; 2 virtual cores CPU and a 12 GB GPU. For data distribution, 80% of the dataset was reserved for training of our models, the remaining 20% for the test. Inspired by He et al. [12], we trained the different models separately, then globally, in order to evaluate the results of the different architectures. The parameters considered during the training were: the number of iterations (epochs), the batch size, the optimization function and the cost function. We used the ADAM optimization function to adjust the weights and attributes of the neural architectures during the training stage. Moreover, we evaluated the AE and MLP models by varying the following parameters:

- AE: variation of the latent dimension of the AE core (LDim = 32, 64) and the number of neuronal layers (#Layers = 1–5),
- MLP: variation of embedding size (ES = 16, 32, 64), number of neuronal layers (#Layers = 1–5),

We obtained the following best values, which will be considered in the rest of our experiments: for AE, the LDim was equal to 64 with a single layer. For MLP, the best performance was obtained with 3 layers and an ES equal to 32.

4.5 Results and Analysis

Evaluation of Hybrid SA Models. By varying the hyper-parameters, we were able to set the following best values: number of epochs for training fixed at 5, with a batch_size of 32, the 'PMSProp' optimization function was chosen as it gave better performance than 'ADAM' and 'SGD'.

Then, we evaluated the four combinations of SA models, LSTM-CNN; BiLSTM-CNN; CNN-LSTM and CNN-BiLSTM, varying the number of LSTM / BiLSTM recurrent units (#RU = 20, 60, 12, 200) and the number of convolutional layers of CNN (#CL = 1, 3, 5).

We can see from Table 2 that each architecture offers better performance with different empirical parameters. The best results were obtained with the CNN-LSTM combination, with 120 recurrent units and 5 convolutional layers. Moreover, this combination is

Table 2. Evaluation summary of the different SA combinations

Models	MAE	RMSE	#CL	#RU	Time-Train (S)
LSTM-CNN	0.1570	0.3073	1	20	6687.19
BiLSTM-CNN	0.1543	0.3201	2	20	11795.25
CNN-LSTM	**0.1318**	**0.3028**	5	120	**2076.71**
CNN-BiLSTM	0.1401	0.3036	1	200	2191.02

characterized by the shortest training time. Accordingly, we'll consider the CNN-LSTM combination in the rest of our evaluation.

Contribution of SA on the Different Models We evaluated the contribution of SA on the different models: the collaborative filtering prediction model (AE), the content-based filtering prediction model (MLP) and the hybrid model (AE-MLP). All the results obtained have shown the contribution of integrating the SA in terms of MAE and RMSE evaluation metrics. Table 3 illustrates the MAE evaluations of the hybrid model with and without SA using the Yelp-Restaurant dataset. The best performance is obtained with LDIM = 64, 1 layer for the AE, 5 layers for the MLP and an embedding size equal to 16.

Table 3. MAE performance of AE-MLP hybrid model with and without SA - Yelp Restaurants

LDim (AE)	#Layers (AE)	ES (MLP)	#Layers (MLP)	AE-MLP-SA	AE-MLP
64	1	16	5	**0.27875**	**0.28090**
		32	4	0.27921	0.28195
		64	2	0.27926	0.28118
	2	16	5	0.27984	0.28083
		32	4	0.28077	0.28095
		64	2	0.28214	0.28106
32	1	16	5	0.27921	0.28156
		32	4	0.28077	0.28209
		64	2	0.28075	0.28137
	2	16	5	0.28044	0.28153
		32	4	0.28107	0.28152
		64	2	0.28143	0.28188

Contribution of the Social Information In order to evaluate the contribution of social information on recommendation performance, we have considered in this evaluation the following weights: $\alpha = 0.6$ (favoring similarity rate rather than degree of credibility); δ

$= -0.5$, $\lambda = 0.4$ and $\gamma = 0.6$ (giving more priority to participation rate instead of degree of competence). We evaluated the contribution of social information on the different AE, MLP and Hybrid models, with and without SA. We present the evaluations carried out with the Hybrid model. We considered the different variants, namely: the Hybrid model combining the AE and MLP architectures without SA and social information (AE-MLP); the Hybrid model combining the AE and MLP architectures with SA and without social information (AE-MLP-SA); the Hybrid model combining the AE and MLP architectures with social information but without SA (AE-MLP-Social); and the Hybrid model combining the AE and MLP architectures with the integration of SA and SA social information (AE-MLP-Social-SA).

We can see that SA improved the performance of the hybrid AE-MLP model and that the integration of social information yielded the best performance on the AE-MLP model. Nevertheless, we noted a slight degradation in performance when we included the sentiment model and social information simultaneously. This degradation is due to the presence of fake reviews. Table 4 shows in detail the values obtained with the different parameters and architectures.

Table 4. Evaluation of the hybrid model with and without SA and social information

LDim (AE)	#Layers (AE)	ES (MLP)	#Layers (MLP)	AE-MLP	AE-MLP-SA	AE-MLP-Social	AE-MLP-Social-SA
64	1	16	5	0.28090	0.27875	0.27648	0.27875
		32	4	0.28195	0.27921	0.27625	0.27921
		64	2	0.28118	0.27926	**0.27511**	0.27926
	2	16	5	0.28083	0.27984	0.27816	0.27984
		32	4	0.28095	0.28077	0.27626	0.28077
		64	2	0.28106	0.28214	0.27666	0.28214
32	1	16	5	0.28156	0.27921	0.27825	0.27921
		32	4	0.28209	0.28077	0.27767	0.28077
		64	2	0.28137	0.28075	0.27758	0.28075
	2	16	5	0.28153	0.28044	0.27692	0.28044
		32	4	0.28152	0.28107	0.27846	0.28107
		64	2	0.28188	0.28143	0.27751	0.28143

Contribution of Fake Reviews Detection In order to evaluate the contribution of fake reviews detection, we compared our model with and without fake reviews. The results show that removing fake reviews has slightly improved the performance (see Table 5). The slight difference is due to the few number of fake reviews in these datasets.

Comparison with Related Work. We compared our model with state-of-the-art work. We tried to select a variety of models, choosing two matrix factorization models, a DL and sentiment analysis-based model and a social-based model. We can see from

Table 5. Evaluation of fake reviews detection

Models	Yelp-Restaurant		Yelp-Shopping	
	MAE	RMSE	MAE	RMSE
CASA-SR with Fake reviews	0.2872	0.3734	0.2941	0.3783
CASA-SR without Fake reviews	**0.2767**	**0.3727**	**0.2831**	**0.3778**

this table that our CASA-SR hybrid model outperformed the other models in terms of MAE and RMSE metrics. Table 6 illustrates the results obtained using both datasets Yelp-Restaurant and Yelp-Shopping.

Table 6. Performance comparison with related work

Models	Yelp-Restaurant		Yelp-Shopping	
	MAE	RMSE	MAE	RMSE
PMF	1.2238	1.4585	0.9529	1.0895
SVD++	0.7181	0.9295	0.7914	1.0235
DeepCoNN	0.3900	0.4800	0.6000	0.7500
SocialMF	0.6000	0.7800	0.7200	0.8900
CASA-SR	**0.2767**	**0.3727**	**0.2831**	**0.3778**

The comparison results show that the DL and SA-based models outperformed the standard matrix factorization models. Similarly, the models based on social information outperformed the PMF and SVD++ models. We note that the different variants of our hybrid model outperformed the state-of-the-art models, with better performance obtained with the CASA-SR model, which combines the AE and MLP architectures with social information and the confidence-aware SA module. Indeed, the results showed an improvement in performance when we eliminated the fake reviews.

5 Conclusion

In this article, we have proposed a novel social recommendation model that combines both AE and MLP models with the integration of social information. An improvement of the evaluation matrix has been performed using a hybrid confidence-aware SA model (that detects and removes fake reviews). Evaluation results on two datasets from the Yelp database demonstrated the contribution of SA and social information on the different models (AE, MLP and the hybrid AE-MLP model). The detection of fake reviews has further enhanced these performances. Furthermore, the different variants of our model outperformed the state-of-the-art models.

Future perspectives include further experiments with higher density datasets. It would be interesting to explore other DL techniques such as knowledge graph DL architectures and use the attention mechanism to improve the performance of the sentiment model. On the other hand, as the complexity of our recommendation algorithm can be significant, it would be interesting to reduce the response times.

References

1. Sun, Z., et al.: Recommender systems based on social networks. J. Syst. Softw. **99**, 109–119 (2015)
2. Guo, G., Zhang, J., Yorke-Smith, N.: TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 123–129 (2015)
3. Yang, B., Lei, Y., Liu, J., Li, W.: Social collaborative filtering by trust. IEEE Trans. Pattern Anal. Mach. Intell. **39**(8), 1633–1647 (2017)
4. Berkani, L., Laga, D., Aissat, A.: Social neural hybrid recommendation with deep representation learning. In: Attiogbé, C., Yahia, S.B. (eds.) Model and Data Engineering, 10th International Conference MEDI 2021, pp. 127–140. Tallinn (2021)
5. Berkani, L., Boudjenah, N.: S-SNHF: sentiment based social neural hybrid filtering. Adv. Data-Driven Eng. **52**(3), 297–325 (2023)
6. Nisha, C.C., Mohan, A.: A social recommender system using deep architecture and network embedding. Appl. Intell. **49**, 1937–1953 (2019)
7. Bathla, G., Aggarwal, H., Rani, R.: AutoTrustRec: recommender system with social trust and deep learning using AutoEncoder. Multimedia Tools Appl. **79**, 20845–20860 (2020). https://doi.org/10.1007/s11042-020-08932-4
8. Wankhade, M., Sekhara Rao, A.-C., Kulkarni, C.: A survey on sentiment analysis methods, applications, and challenges. Artif. Intell. Rev. (2022). https://doi.org/10.1007/s10462-022-10144-1
9. Dang, C.N., Moreno-García, M.N., De la Prieta, F.: An approach to integrating sentiment analysis into recommender systems. Sensors **21**, 5666 (2021). https://doi.org/10.3390/s21165666
10. Li, D., et al.: CARM: confidence-aware recommender model via review representation learning and historical rating behavior in the online platforms. Neurocomputing **455**, 283–296 (2021)
11. Birim, S.O., Kazancoglu, I., Mangla, S.K., Kahraman, A., Kumar, S., Kazancoglu, Y.: Detecting fake reviews through topic modeling. J. Bus. Res. **149**, 884–900 (2022)
12. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182 (2017)
13. Berkani, L., Kerboua, I., Zeghoud, S.: Recommandation Hybride basée sur l'Apprentissage Profond. Actes de la conférence EDA 2020, Revue des Nouvelles Technologies de l'Information, RNTI B.16, pp. 69–76 (2020). ISBN: 979-10-96289-13-4
14. Berkani, L., Zeghoud, S. Kerboua, I.: Chapter 19 - Neural hybrid recommendation based on GMF and hybrid MLP. In: Pandey, R., Khatri, S.K., Singh, N.K., Verma, P. (eds.) Artificial Intelligence and Machine Learning for EDGE Computing, Academic Press, pp.287–303 (2022). ISBN 9780128240540. https://doi.org/10.1016/B978-0-12-824054-0.00030-7
15. Rama, K., Kumar, P., Bhasker, B.: Deep autoencoders for feature learning with embeddings for recommendations: a novel recommender system solution. Neural Comput. Appl. **33**, 14167–14177 (2021). https://doi.org/10.1007/s00521-021-06065-9

16. Diao, Q., Qiu, M., Wu, C.Y., Smola, A.J., Jiang, J., Wang, C.: Jointly modeling aspects, ratings and sentiments for movie recommendation. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.193–202 (2014)
17. Lu, Y., Dong, R., Smyth, B.: Co-evolutionary recommendation model: mutual learning between ratings and reviews. In: Proceedings of the World Wide Web conference, pp. 773–782 (2018)
18. Osman N.A., Mohd Noah, S.A., Darwich, M., Mohd, M.: Integrating contextual sentiment analysis in collaborative recommender systems. PLoS ONE **16**(3), e0248695 (2021). https://doi.org/10.1371/journal.pone.0248695
19. Avesani, P., Massa, P., Tiella, R.: Moleskiing it: a trust-aware recommender system for ski mountaineering. Int. J. Infonomics **20**, 1–10 (2005)
20. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv (2018). arXiv:preprint/04805
21. Jiang, L., Liu, L., Yao, J., Shi, L.: A hybrid recommendation model in social media based on deep emotion analysis and multi-source view fusion. J. Cloud Comput. Adv. Syst. Appl. **9**, 57 (2020)
22. Berkani, L., Belkacem, S., Ouafi, M., Guessoum, A.: Recommendation of users in social networks: a semantic and social based classification approach. Expert. Syst. **38**(2), e12634 (2021). https://doi.org/10.1111/exsy.12634
23. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems, pp.135–142 (2010). https://doi.org/10.1145/1864708.1864736
24. Zheng, L., Noroozi, V., Yu, P.S.: Joint deep modeling of users and items using reviews for rec. arXiv:1701.04783 (2017). https://doi.org/10.48550/arXiv.1701.04783