Enhancing Recommender Systems Using Sentiment Analysis: Addressing Cold Start Issues and Improving Recommendation Quality

Sana Nabil *, Jaber El Bouhdidi , and Mohamed Yassin Chkouri

Information Systems and Software Engineering (SIGL) Laboratory, National School of Applied Sciences of Tetuan (ENSATE), Abdelmalek Essaadi University, Tetuan, Morocco

Email: sana.nabil@etu.uae.ac.ma (S.N.); jaber.elbouhdidi@uae.ac.ma (J.E.B.); mychkouri@uae.ac.ma (M.Y.C.)

*Corresponding author

***Abstract*—Recommender systems are algorithms designed to provide personalized suggestions for products and services that are most relevant to individual users. These systems analyze historical data about user interactions, such as reviews, clicks, likes, and dislikes, to predict and recommend products or services that align with users' interests. By leveraging this data, recommender systems can effectively tailor suggestions to each user, enhancing their overall experience and satisfaction. Recently, sentiment analysis has gained traction as an approach for recommendation systems, offering an alternative to traditional methods such as collaborative filtering, content-based approaches, and hybrid models. By analyzing user sentiments expressed in product reviews, these systems can make recommendations based on the emotional tone of the feedback, leading to more personalized and contextually relevant suggestions. So, to enhance the performance of recommendation systems and address the cold start problem for new users and data scalability issues, we propose a hybrid approach that integrates sentiment analysis with collaborative filtering. This system leverages deep learning algorithms alongside traditional collaborative filtering techniques, such as Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN). We analyzed user reviews and comments using IMDB, Amazon, and Amazon Fine Foods datasets for experiments and results. Our approach recommends the best items and improves overall system accuracy, resolving cold start problems and data scalability. The results demonstrate that integrating sentiment analysis significantly enhances recommendation quality compared to traditional collaborative filtering methods and effectively mitigates the cold start issue**.

***Keywords*—sentiment analysis, deep learning, recommender system, opinion mining, Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), collaborative filtering, factorization matrix, users' ratings, cold start**

I. Introduction

Sentiment analysis has become highly valuable in various domains, including e-commerce, customer relationship management, and social media. By gaining deeper insights into users' opinions, attitudes, and emotions, sentiment analysis can significantly enhance the reliability of recommendation systems. Integrating sentiment analysis into these systems leads to more accurate and personalized recommendations, ultimately improving user satisfaction and engagement.

A recommender system is designed to provide personalized recommendations for products or services, assisting users in decision-making and guiding them toward the most relevant options. Among the myriad of recommender algorithms and techniques, the most well-known are collaborative filtering, content-based filtering, and hybrid approaches. Recently, sentiment analysis of user reviews and opinions has emerged as a crucial foundation for recommender systems [1], enabling them to predict users' future preferences based on their past sentiments. This integration enhances the system's ability to tailor recommendations more accurately to individual users.

Recommender systems continue to face critical challenges, particularly with collaborative filtering methods [1], which struggle with the cold start problem for new users or items. Similarly, content-based recommender systems often lack diversity in their recommendations, negatively impacting overall system accuracy. Addressing these challenges is crucial to enhancing the effectiveness of sentiment-based recommender systems. Incorporating advanced techniques such as deep learning and hybrid models can provide significant improvements.

In recent years, deep learning models like Transformers and Graph Neural Networks (GNNs) have transformed the field of recommendation systems. Transformers, with their attention mechanisms, effectively capture the users' behavior for recommendation, while GNNs model complex relationships within user-item networks, uncovering deeper insights into preferences. However, these methods often require substantial computational resources and large datasets, which can pose practical limitations.

Manuscript received October 22, 2024; revised November 27, 2024; accepted January 7, 2025; published April 27, 2025.

doi: 10.12720/jait.16.4.556-567

This study proposes a hybrid approach that leverages Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, combined with Term Frequency-Inverse Document Frequency (TF-IDF) and GloVe embeddings, to perform sentiment-based recommendations. By integrating collaborative filtering into this framework, the approach seeks to improve recommendation accuracy while addressing the cold start problem, offering a practical and effective solution to these challenges.

In this article, we address three major challenges in recommendation systems. First, we tackle the cold start problem for new users by incorporating sentiment analysis of existing user reviews. This approach allows us to infer preferences for new users and recommend items with the most positive reviews, even when user data is minimal. Second, we address the data scalability challenge. As the number of users and items grows, processing and generating recommendations become increasingly resource-intensive. Sentiment analysis helps mitigate this by filtering and prioritizing data with the most positive reviews, reducing the computational load. Finally, by combining collaborative filtering with sentiment analysis, we enhance the system's ability to predict user preferences more accurately, leading to more precise recommendations.

Sentiment analysis is performed using deep learning techniques, specifically Recurrent Neural Network (RNN) LSTM and CNN algorithms. We begin by preprocessing the data to prepare it for analysis and prediction, then combine the sentiment analysis results with a collaborative filtering algorithm to enhance recommendation accuracy and effectiveness. The data preprocessing stage involves removing missing values, special characters, numbers, and any irrelevant elements that might reduce the effectiveness of the sentiment analysis process. After preprocessing, the data undergoes tokenization, where it is split into individual tokens. This is followed by the extraction of vectors using a word embedding algorithm and TF-IDF technique. Finally, sentiment analysis is performed using CNN and LSTM models. After predicting the sentiment scores, if it's the user's first time accessing the platform, we use these sentiment scores to recommend suitable products based on the analysis of their reviews. For returning users, we combine the sentiment analysis results with a collaborative filtering approach to refine the recommendations. This article is structured as follows: The first section covers related works, where we discuss various studies and approaches relevant to our research. The second section outlines our proposed approach. The third section presents the experiments conducted and their results. Finally, the conclusion summarizes our findings and future directions.

II. Literature Review

Hassan *et al*. [2] developed a recommender system that leverages sentiment analysis to address data sparsity issues using Natural Language Processing (NLP) techniques and machine learning algorithms. Their study evaluated the performance of various algorithms on three datasets: IMDB reviews, Yelp restaurant reviews, and Arabic restaurant reviews from qaym.com. The results showed that Naïve Bayes achieved the best precision of 0.88 on the Yelp dataset, Logistic Regression performed best with a precision of 0.91 on the IMDB dataset, and both Decision Tree and Logistic Regression delivered the highest precision of 0.82 on the Arabic dataset. These variations suggest that dataset-specific factors, such as language, domain, and feature representation, strongly influence model performance.

Osman *et al*. [3] proposed a recommender system model that integrates sentiment analysis with contextual information, combining it with collaborative filtering techniques. The novelty of their approach lies in the fusion of sentiment analysis with collaborative filtering for enhanced recommendation accuracy. Their results indicate that this integrated model outperforms traditional collaborative filtering methods, demonstrating its effectiveness in improving recommendation quality.

Asani *et al*. [4] proposed a method for extracting user food preferences from online restaurant comments using NLP techniques. Their approach involves processing text to identify food names, employing semantic similarity for conceptual clustering of these names, and utilizing the Wu–Palmer method for improved precision in clustering results. Sentiment analysis is then applied to determine whether user opinions about each food are positive or negative. The method culminates in a recommender system that suggests nearby restaurants aligned with the user's food preferences.

Hung [5] developed a recommender system that integrates sentiment analysis with a hybrid deep learning method combining CNN and LSTM for analyzing customer review sentiments based on word vectors. The system then uses the sentiment analysis results alongside neighbor item ratings through matrix factorization to predict user preferences. Using the Amazon food review dataset, the study found that CNN achieved a precision of 0.8276, LSTM 0.8254, and the CNN-LSTM combination 0.8345. In terms of recommendation performance, the baseline popular method had a result of 1.7364, while the matrix factorization integrated with sentiment analysis achieved a significantly improved result of 1.1536.

Hsu *et al*. [6] proposed a recommendation system called the Sentiment Analysis and Machine Learning Framework (SAMLRF), which utilizes both machine learning and deep learning algorithms for sentiment analysis. They tested four machine learning algorithms (logistic regression, support vector machine, decision tree, and gradient boosting decision tree) along with two deep learning algorithms (simple recurrent neural networks and long short-term memory). Their results indicated that the decision tree and gradient boosting decision tree algorithms provided the best performance, achieving an impressive accuracy of 0.99 on a data of 6000 articles. However, small datasets may limit the generalizability of these results to more diverse or larger-scale datasets. This highlights the importance of validating models across multiple datasets to ensure robust and scalable performance. Despite this limitation, the study presents the potential of combining machine learning and deep learning techniques for sentiment-based recommendations and serves as a valuable benchmark for hybrid systems.

Dang *et al*. [7] proposed a recommendation system integrates sentiment analysis with Collaborative Filtering (CF). Using two datasets Amazon Fine Foods Reviews and Amazon Movie Reviews, they demonstrated that combining deep learning-based sentiment analysis with collaborative filtering significantly enhances the system's performance. Their results show high performance, with accuracy and F-score exceeding 80% and AUC surpassing 84% for sentiment analysis. For the combined approach, the metrics with SVD++ include a Mean Absolute Error (MAE) of 0.577, a Root Mean Square Error (RMSE) of 0.8577, and a Normalized Mean Absolute Error (NMAE) of 0.1443, all with a regularization parameter B = 0.3.

Table I illustrates the Summary of the previous discussed related works with the advantages and disadvantages of each work.

TABLE I. SUMMARY OF THE PREVIOUS DISCUSSED RELATED WORKS WITH THE ADVANTAGES AND DISADVANTAGES

Authors	Methodology	Advantages	Disadvantages
Hassan and Abdulwahhab [2].	Leveraged sentiment analysis with NLP and machine learning to address data sparsity across datasets.	High precision across datasets; dataset-specific evaluation.	Performance varies significantly depending on dataset structure and feature representation.
Osman *et al*. [3].	Integrated sentiment analysis with contextual information and collaborative filtering.	Improved recommendation accuracy over traditional collaborative filtering.	Limited analysis of scalability and cold-start issues.
Asani *et al*. [4].	Used NLP to extract user food preferences and applied sentiment analysis for clustering results.	Effective in clustering and determining user preferences for personalized recommendations.	Focused on a specific domain (food/restaurants); may not generalize well to other datasets.
Hung [5].	Combined CNN, LSTM, and matrix factorization for hybrid sentiment analysis and recommendation.	Achieved high precision and improved recommendation quality using hybrid deep learning methods.	Higher error rates in collaborative filtering metrics (MAE, RMSE) compared to other hybrid models.
Hsu and Liao [6].	Developed SAMLRF using machine learning and deep learning for sentiment analysis.	Achieved 0.99 accuracy, demonstrating exceptional performance in sentiment analysis.	Small dataset may limit the generalizability of these results to more diverse or larger-scale datasets.
Dang *et al*. [7].	Integrated sentiment analysis with collaborative filtering using deep learning and matrix factorization.	Enhanced performance metrics (accuracy, F-score, AUC); effective use of sentiment analysis.	Scalability and computational efficiency remain challenges.
Karabila *et al*. [8].	Combined Bi-LSTM sentiment analysis with collaborative filtering techniques.	High sentiment classification accuracy; effective integration of collaborative filtering.	Higher error rates in collaborative filtering metrics (MAE, RMSE) compared to other hybrid models.
Dang *et al*. [9].	Hybrid deep-learning-based sentiment analysis with collaborative filtering	Leverages genre attributes for better recommendations SVD++ achieves superior performance Enhances accuracy and recommendation quality	Scalability in real-world large-scale systems not addressed
Al-Rossais [10].	Deep Neural Network (DNN) for addressing cold start problems	Significantly improves cold start problem Evaluates key metrics like fairness and serendipity High accuracy in recommendation	Focuses only on cold start issues
Putri *et al*. [11].	CNN for book recommendation system	Effectively handles scalability and sparsity issues Best parameter combinations achieved high evaluation metrics	Dataset limited to book recommendations from GitHub Scalability in real-time recommendation not explored
Choi *et al*. [12].	Collaborative filtering with two-stage clustering	Reduces data sparsity by clustering Improves scalability Predicts preference scores effectively	Does not address dynamic changes in user preferences

Karabila *et al*. [8] developed a recommender system that combines sentiment analysis with collaborative filtering. Their approach involved first constructing a sentiment model using Bidirectional Long Short-Term Memory (Bi-LSTM) and then developing a recommendation model based on collaborative filtering techniques. The sentiment analysis component achieved a high accuracy of 93% for sentiment classification. For the collaborative filtering part, the system produced a Mean Absolute Error (MAE) of 1.32 and a Root Mean Square Error (RMSE) of 1.46, demonstrating effective integration of sentiment analysis into the recommendation process.

Dang *et al*. [9] proposed a recommender system that leverages the genre attribute alongside a hybrid deep-learning-based sentiment analysis of reviews to enhance collaborative filtering-based recommendations. Their study utilized the MARD dataset, comprising 263,525 samples, and the Amazon Movie dataset, containing 203,967 samples. They evaluated three CF algorithms: SVD, NMF, and SVD++. The best results in terms of RMSE and MSE were achieved using SVD++ on the MARD dataset, with values of 0.9089 and 0.6733, respectively. On the Amazon Movie dataset, SVD++ also delivered the best performance, with RMSE and MSE values of 1.0689 and 0.7965, respectively.

Nourah A. employs a Deep Neural Network (DNN) algorithm to address the cold start problem in recommender systems. The study evaluates performance across various metrics, including accuracy and the value content of ranked lists, as well as serendipity and fairness—metrics that are increasingly crucial for online platforms offering recommendations. The findings reveal that a multi-layer DNN significantly enhances accuracy metrics for the cold start problem. Using a public dataset of movie reviews, movie features, and user features combined from MovieLens 1M and IMDb, the results demonstrate substantial improvements. For the new user problem, the obtained metrics include an RMSE of 0.887 and an MAE of 0.691, while for the new item problem, the RMSE and MAE are 0.891 and 0.711, respectively [10].

Putri *et al*. [11] propose an approach that leverages the CNN algorithm in a book recommender system to address scalability and sparsity challenges. The study utilizes a book recommendation dataset sourced from GitHub. To evaluate the system's performance, various parameter combinations were tested, including filter size, kernel size, and the number of epochs. The best model was achieved with a filter size of 8, a kernel size of 2, and 50 epochs, yielding MAE, MAPE, and MSE values of 1.361, 23.766, and 3.034, respectively.

Choi *et al*. [12] propose a collaborative filtering technique that addresses the limitations of data sparsity and scalability in traditional collaborative filtering recommendation systems through a two-stage clustering approach. In the first stage, the method utilizes customers' basic information to perform clustering, subsequently predicting individual customers' preference scores based on the preference information of their respective clusters. This approach helps to fill gaps in the customer-item matrix. In the second stage, clustering is used to reduce the data space, effectively improving the system's scalability.

III. THE PROPOSED SOLUTION

While works like those by Hassan *et al*. [2] and Hsu *et al*. [6] demonstrate the effectiveness of individual techniques, they lack a robust mechanism for tackling both the cold-start problem and scalability simultaneously. In contrast, the proposed model tackles the problem of scalability, cold start problem for a new user and improve recommendation accuracy.

The key advantage of the proposed model lies in its integration of sentiment analysis with collaborative filtering techniques, which offers several distinct benefits over existing approaches:

- **Improved Recommendation Accuracy**

By incorporating sentiment analysis, the model captures nuanced user preferences expressed in reviews. After predicting the sentiment of user reviews, the dataset is refined to include only products with more positive reviews than negative ones, effectively reducing its size. The predicted sentiment scores are then used to correct erroneous ratings, ensuring greater accuracy. This refined dataset, combined with corrected ratings, is subsequently used as input for collaborative filtering, leading to improved recommendation results.

- **Addressing the Cold-Start Problem**

Unlike many previous approaches that struggle with new users, the proposed model utilizes Sentiment Analysis (SA) to recommend top-rated items by analyzing and prioritizing their reviews, offering good recommendations to new users effectively.

- **Enhanced Scalability**

After predicting the sentiment of user reviews, we filtered the dataset to retain only products with more positive reviews than negative ones, thereby effectively reducing the dataset.

- **Domain-Generalization Capability**

While previous works focus on specific domains (e.g., movies, food, or restaurants), the proposed model's generalized sentiment analysis approach ensures adaptability across various domains.

IV. MATERIALS AND METHODS

In this work, we propose a two-part recommender system designed to enhance accuracy, improve data scalability, and address the cold start problem. For new users who join the platform for the first time, we utilize a sentiment analysis approach based on existing users' reviews. By employing deep learning algorithms, such as Simple Neural Network (SNN), CNNs, and LSTMs, we provide initial recommendations even without personal user data, effectively mitigating the cold start issue by leveraging sentiment insights from other users.

For existing users, the system combines user ratings with sentiment analysis of reviews. Initially, items are classified based on review sentiments, retaining only those with a higher number of positive reviews compared to negative ones. Ratings are then preprocessed by replacing null, zero, empty, and string ratings with sentiment scores. We utilize collaborative filtering with the Alternating Least Squares (ALS) algorithm to predict ratings, which are further refined through the analysis of review comments using deep learning models.

This integrated approach aims to enhance collaborative filtering results, leading to more accurate and relevant recommendations tailored to user preferences.

Fig. 1 illustrates the general architecture of our proposed system, which consists of two main modules to enhance recommendation accuracy and address the cold start problem. The first module targets new users by providing recommendations based on sentiment analysis of existing users' reviews. This process involves extracting and preprocessing user reviews, feature extraction using an embedding matrix, and sentiment classification into positive and negative categories using deep learning algorithms like CNN and LSTM. Items are then recommended based on the most positive reviews. The second module focuses on existing users by first classifying items to retain those with predominantly positive reviews and correcting erroneous or missing ratings with sentiment scores. Finally, collaborative filtering, specifically using the ALS algorithm, is applied to predict and refine user ratings, thereby enhancing the relevance and precision of recommendations for returning users.

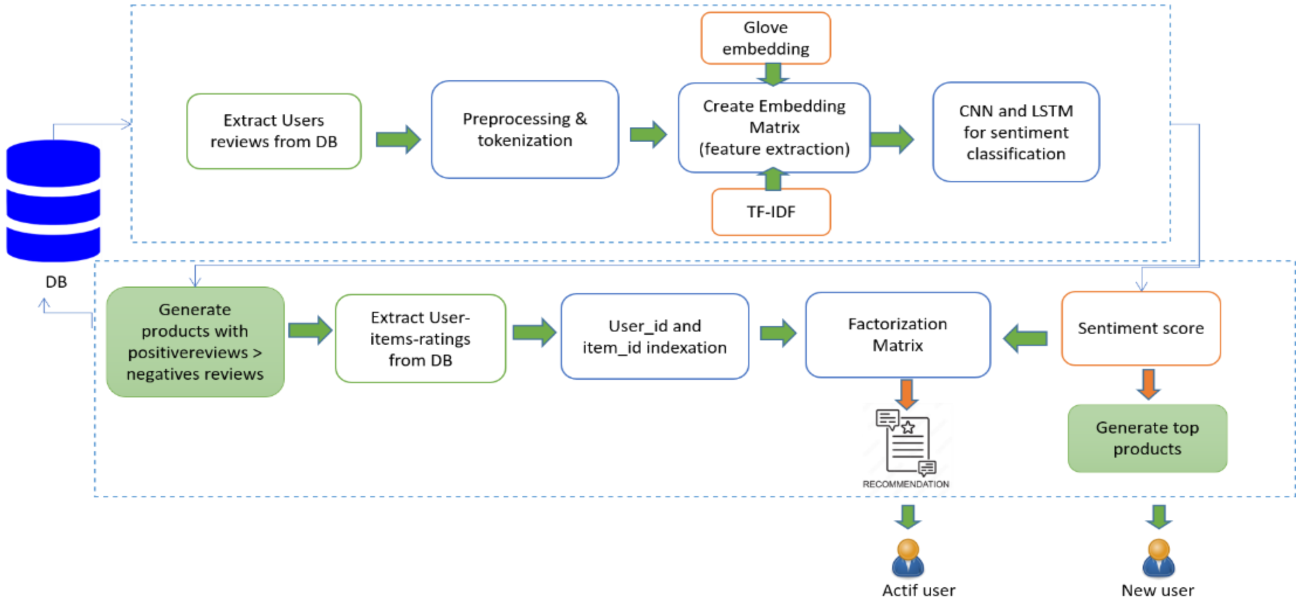

Fig. 1. The general architecture of our recommender system.

A. Sentiment Analysis Approach

Fig. 2 illustrates the architecture of the sentiment analysis model used in our system. The process begins with pre-processing the historical reviews dataset, which involves tokenization and feature extraction using word embedding techniques. The data is then split into 80% for training and 20% for testing. We train the model using deep learning algorithms, specifically CNN and LSTM, to capture the nuances of user sentiment. After evaluating the model's performance, the trained model is applied to analyze and predict sentiment for new reviews. For new reviews, the process mirrors the training procedure: pre-processing, tokenization, word embedding, and finally, sentiment prediction using the trained model.

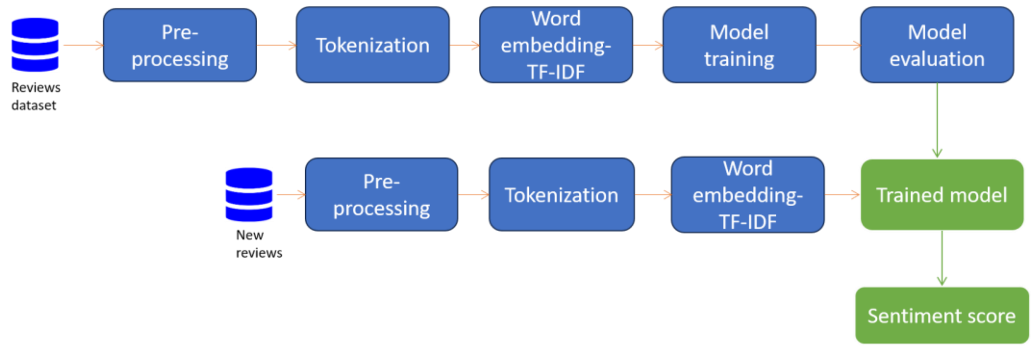

Fig. 2. Sentiment analysis process.

1) Data preprocessing

In this part we remove the special characters, and we prepare the data and adapt it to the next step. The data are preprocessed by removing the missing values, special characters, numbers, anything that is not relevant and that can reduce the effectiveness of the sentiment analysis process.

- Remove html tags
- Remove punctuations and numbers
- Remove single character
- Remove stop words

2) Tokenization

In NLP and machine learning, tokenization involves splitting text into smaller units, or 'tokens', such as words or phrases. This process is essential because it enables machines to parse and interpret human language by breaking down complex text into small components, facilitating further analysis [13].

3) Feature extraction

a) Embedding layer with pre-trained glove embeddings

The embedding layer utilizes pre-trained GloVe word embeddings (glove.6B.100d.txt) to transform the text into dense vectors that represent the semantic relationships between words. These vectors are then fed into the neural network as input features. In neural networks, the embedding layer serves as a mechanism for converting categorical data, such as words, into continuous vector representations. This process maps each word in the vocabulary to a fixed-size vector within a lower-dimensional space, helping the model process text in a way that retains the semantic meaning. GloVe embeddings, derived from the Global Vectors for Word Representation (GloVe) model, are pre-trained on a large corpus and capture word co-occurrence patterns. This allows words that frequently appear together in similar contexts to have closer vectors in the embedding space [14].

b) TF_IDF

TF-IDF is a statistical method used to assess how significant a word is within a specific document compared to a larger collection of documents (corpus). It helps transform textual data into a structured format for machine learning models. The technique assigns higher importance to words that frequently appear within a specific document but are less common across the entire corpus. The 'TF' (Term Frequency) component calculates how often a word appears in a document, while 'IDF' (Inverse Document Frequency) reduces the importance of words that appear frequently across many documents, highlighting words that are unique to specific contexts [15, 16].

4) Deep learning algorithms

A Neural Network is a machine learning method technique that is inspired by and resemble human nervous system and the structure of the brain. It consists of interconnected layers: input, hidden, and output layers. Each layer contains nodes or neurons that pass data through weighted connections to nodes in adjacent layers, mimicking brain function. In this study, we employed three deep learning algorithms—SNN, CNN, and LSTM—due to their high performance and effectiveness in sentiment analysis of textual data [5].

a) CNN

CNNs are a specialized type of Artificial Neural Network (ANN) that apply convolutional operations in one or more of their layers. Originally inspired by the primate visual system and explore spatial invariances in the data [17]. CNN is a regularized type of feed-forward neural network that learns feature engineering by itself via filters (or kernel) optimization. Vanishing gradients and exploding gradients, seen during backpropagation in earlier neural networks, are prevented by using regularized weights over fewer connections [18, 19].

In sentiment analysis, CNNs are effective tools for classifying text by recognizing patterns in word embeddings. CNNs transform text into matrices, where each word is represented as a vector, typically through embeddings like GloVe. Convolutional filters are then applied to these matrices to extract relevant features, which help in detecting sentiment-related patterns. After this, a pooling layer is used to reduce dimensionality while preserving the most important features. The final layers classify the text as either positive, negative, or neutral sentiment. The application of CNNs in sentiment analysis shows strong performance in classifying sentences and achieving high accuracy on datasets like IMDB reviews [20, 21].

b) LSTM

A recurrent neural network is a type of advanced artificial neural network. RNNs can use their internal state (memory) to process sequences of inputs. RNNs have shown great success in many natural language processing tasks. They connect previous information to the present task. RNNs can learn to use the past information [22].

Long-term dependencies can be captured by LSTM networks. They have a memory cell that is capable of long-term information storage [13].

LSTM is a powerful recurrent neural network architecture commonly used in sentiment analysis due to its ability to capture long-term dependencies in textual data. Sentiment analysis involves detecting emotions, opinions, or attitudes in text, and LSTM networks excel in processing sequences like text where the context and order of words are critical for understanding sentiment.

5) Training and evaluation

The model is trained using:

- Optimizer: Adam.
- Loss Function: Binary cross-entropy, suitable for binary classification tasks.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 100)]	0	[]
embedding (Embedding)	(None, 100, 100)	9254700	['input_1[0][0]']
lstm (LSTM)	(None, 100, 128)	117248	['embedding[0][0]']
input_2 (InputLayer)	[(None, 1000)]	0	[]
global_max_pooling1d (GlobalMaxPooling1D)	(None, 128)	0	['lstm[0][0]']
dense (Dense)	(None, 128)	128128	['input_2[0][0]']
concatenate (Concatenate)	(None, 256)	0	['global_max_pooling1d[0][0]', 'dense[0][0]']
dense_1 (Dense)	(None, 128)	32896	['concatenate[0][0]']
dropout (Dropout)	(None, 128)	0	['dense_1[0][0]']
dense_2 (Dense)	(None, 1)	129	['dropout[0][0]']

Fig. 3. Sentiment analysis architecture using LSTM with Glove and TF-IDF.

The architecture (Fig. 3) is designed for sentiment classification and combines GloVe embeddings with LSTM and TF-IDF features. The input text is processed in two parallel pipelines: the first utilizes GloVe embeddings to generate 100-dimensional dense word vectors, which are then fed into an LSTM layer to capture sequential dependencies. The output of the LSTM is summarized using a Global Max Pooling layer to extract the most salient features. The second pipeline employs TF-IDF for feature extraction, with the resulting sparse vectors transformed into dense representations through a fully connected layer. The outputs of both pipelines are concatenated into a unified 256-dimensional vector, which is passed through additional dense layers with dropout regularization to mitigate overfitting. Finally, a single neuron with sigmoid activation is used to classify the sentiment, effectively leveraging both contextual and statistical features for accurate predictions.

This architecture (Fig. 4) is designed for sentiment classification, combining the strengths of GloVe embeddings and TF-IDF to enhance performance. The input text is processed through two parallel pipelines: one uses GloVe embeddings to convert words into dense 128-dimensional vectors, which are then passed through a 1D CNN to capture local patterns and contextual dependencies, followed by a Global Max Pooling layer to summarize key features. The other pipeline uses TF-IDF to generate sparse feature vectors representing word importance, which are then transformed into dense representations through a dense layer. The outputs from both pipelines are concatenated into a 256-dimensional feature vector, integrating contextual and statistical text information. This fused vector is passed through a dense layer with dropout for regularization and a final sigmoid-activated neuron for sentiment prediction, enabling the model to leverage complementary features for improved classification accuracy.

Layer (type)	Output Shape	Param #	Connected to
input_3 (InputLayer)	[(None, 100)]	0	[]
embedding_1 (Embedding)	(None, 100, 128)	128000	['input_3[0][0]']
conv1d (Conv1D)	(None, 96, 128)	82048	['embedding_1[0][0]']
input_4 (InputLayer)	[(None, 1000)]	0	[]
global_max_pooling1d_1 (GlobalMaxPooling1D)	(None, 128)	0	['conv1d[0][0]']
dense_3 (Dense)	(None, 128)	128128	['input_4[0][0]']
concatenate_1 (Concatenate)	(None, 256)	0	['global_max_pooling1d_1[0][0]', 'dense_3[0][0]']
dense_4 (Dense)	(None, 128)	32896	['concatenate_1[0][0]']
dropout_1 (Dropout)	(None, 128)	0	['dense_4[0][0]']
dense_5 (Dense)	(None, 1)	129	['dropout_1[0][0]']

Fig. 4. Sentiment analysis architecture using CNN with Glove and TF-IDF.

B. Collaborative Filtering

Collaborative filtering is commonly used for recommender systems. These techniques aim to fill in the missing entries of a user-item association matrix. We used model-based collaborative filtering, in which users and products are described by a small set of latent factors that can be used to predict missing entries, using Alternating Least Squares by Spark ml.

a) Alternating least squares

ALS is a matrix factorization technique commonly used in collaborative filtering algorithms, particularly for recommendation systems. The aim of ALS is to predict missing values in the user-item interaction matrix (such as user ratings for products) by decomposing it into two lower-dimensional matrices: one representing users U and the other representing items V. The product of these matrices approximates R, which allows predictions of unknown user-item interactions such as ratings. This technique is central to collaborative filtering, particularly for recommendations [23].

ALS is scalable and can handle large-scale datasets, as it can be distributed over multiple computing nodes, making it highly efficient for big data applications. For example, ALS is implemented in Apache Spark's MLlib, where it can efficiently process large recommendation datasets [24].

b) The collaborative filtering combined with sentiment analysis approach

Mistaken ratings can lead to significant errors during the execution phase and negatively affect recommendation outcomes. After predicting the sentiment of user reviews, we filtered the dataset to include only products where the number of negative reviews is less than the positive reviews, effectively reducing the dataset. Following this, the predicted sentiment score was used to correct erroneous ratings. As depicted in Fig. 5, we replaced mistaken ratings with the corresponding sentiment-based score (i.e., if the sentiment is 0, the score is adjusted to 1; if the sentiment is positive, the score is adjusted to 5). Figs. 5 and 6 demonstrate the integration of sentiment analysis with collaborative filtering. Specifically, Fig. 5 shows how erroneous ratings are updated based on sentiment analysis results. Fig. 6 illustrates the use of the updated ratings in the collaborative filtering.

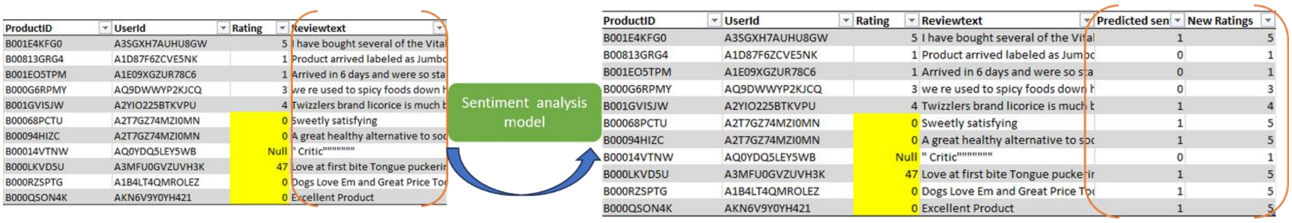

Fig. 5. Sentiment analysis for ratings correction.

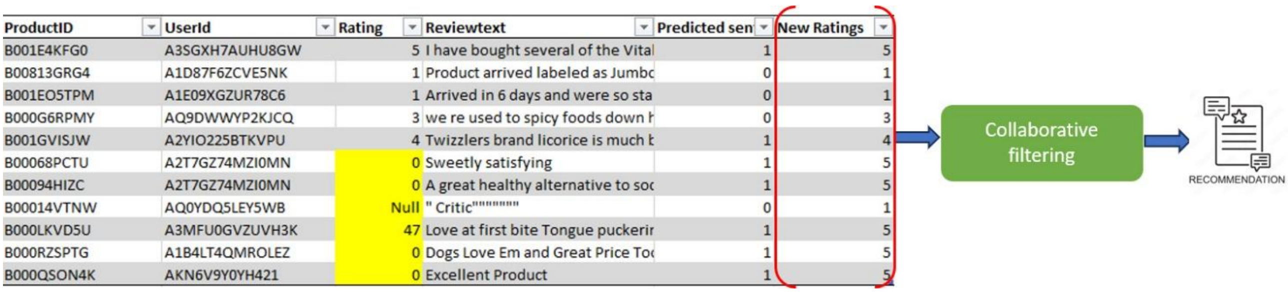

Fig. 6. Collaborative after the update of the ratings.

V. Result and Discussion

A. Results

As mentioned earlier; to address the cold start problem for new users, we utilize SA to recommend the top-reviewed items based on the analysis of their reviews. Figs. 7 and 8 show the results of recommendations generated using sentiment analysis for new users, illustrating the top 5 and top 10 recommendations, respectively. Similarly, Figs. 9 and 10 display the top 5 and top 10 recommendations for existing users on the platform, where sentiment analysis is combined with collaborative filtering.

```
recommendations for the new user
id  ProductId  number_of_positive_comments
0 B007JFMH8M        204
1 B002QWHJOU        111
2 B002QWP89S        108
3 B0026RQTGE        106
4 B005K4Q34S        106
5 B005K4Q1YA        105
6 B005K4Q37A        105
7 B005K4Q4LK        105
8 B002QWP8H0        103
9 B000UBD88A         93
```

Fig. 7. Top 10 recommendations for a new user using SA.

```
recommendations for the new user
id  ProductId  number_of_positive_comments
0 B007JFMH8M        204
1 B002QWHJOU        111
2 B002QWP89S        108
3 B0026RQTGE        106
4 B005K4Q34S        106
```

Fig. 8. Top 5 recommendations for a new user using SA.

```
Top 10 recommendations for user 269:
---------------------------------
Rating(269,6361,5.0)
Rating(269,7802,5.0)
Rating(269,6687,5.0)
Rating(269,8719,5.0)
Rating(269,11718,5.0)
Rating(269,15665,5.0)
Rating(269,11392,5.0)
Rating(269,9610,5.0)
Rating(269,6051,5.0)
Rating(269,10422,5.0)
```

Fig. 9. Top 10 recommendations for a new user using SA and CF.

```
Recommendations: (MovieId => Rating)
---------------------------------
MovieID: 6361 => Rating: 5.0
MovieID: 7802 => Rating: 5.0
MovieID: 6687 => Rating: 5.0
MovieID: 8719 => Rating: 5.0
MovieID: 11718 => Rating: 5.0
```

Fig. 10. Top 5 recommendations for a new user using SA and CF.

B. Datasets

In this work we used the IMDB and Amazon datasets to train and test the sentiment analysis part. Table II demonstrates each dataset with their attributes and description.

TABLE II. Datasets

Datasets	Attributes	Description
IMDB Dataset	Review	Users' comments from IMDB
	Sentiment	Sentiment class (positive or negative)
Amazon Dataset	Label	Sentiment Class Label1 for negative Label2 for Positive
	Text	Reviews from Amazon Database
Amazon Fine Foods Dataset	User_id	Unique identifier for the user
	Product_id	Unique identifier for the product
	Score	Rating between 1 and 5
	Review text	Text of the review

- **IMDB dataset**

In this work, we used the IMDB and Amazon datasets to train and test the sentiment analysis part. Table II demonstrates each dataset with its attributes and description.

- **IMDB dataset**

IMDB dataset contains 50K movie reviews. This is a dataset for binary sentiment classification positive or negative. In our experiments, we used 80% of reviews for training and 20% for testing [25].

- **Amazon dataset**

This dataset consists of a few million Amazon customer reviews. It consists of two attributes Label contains the sentiment class for sentiment analysis in this case, the classes are__label__1correspond to negative reviews and __label__2 correspond to negative reviews, and the attribute text corresponds to reviews from the Amazon dataset [26].

- **Amazon fine foods**

This dataset consists of reviews of fine foods from Amazon. The data contains 500,000 reviews from Oct 1999 to Oct 2012. Reviews include 568,454 reviews by 256,059 users on 74,258 products [27].

C. Experiments

In our effort to enhance sentiment analysis results and apply it as a solution to the cold start problem, as well as to improve the performance of collaborative filtering in recommender systems, we utilized three well-known deep learning algorithms: the SNN, CNN, and LSTM. For our experiments, we evaluated sentiment analysis performance using metrics such as Accuracy, Precision, Recall, F1 Score, and Test Error.

We employed two datasets: the IMDB dataset, with 50,000 reviews, and the Amazon dataset, also consisting of 50,000 reviews. In the case of collaborative filtering combined with sentiment analysis, we used 568,454 reviews from the Fine Foods dataset. To assess the performance of the collaborative filtering model with sentiment analysis, we measured Accuracy, Precision, Test Error, Loss, Recall, and F1 Score.

Table III reports the results obtained on the IMDB dataset while testing various approaches used for sentiment analysis. Using the GloVe embeddings and TF-IDF techniques for data vectorization, and applying the SNN, we achieved an accuracy of 0.74 using GloVe embeddings, 0.88 with TF-IDF, and 0.88 when combining both GloVe embeddings and TF-IDF. When using the CNN algorithm, the accuracy results were 0.84 for GloVe embeddings, 0.88 for TF-IDF, and 0.88 when combining both techniques. Similarly, for the LSTM algorithm, we obtained 0.85, 0.87, and 0.88 for GloVe, TF-IDF, and both, respectively. The CNN-LSTM hybrid model yielded accuracies of 0.84 using GloVe embeddings and 0.88 with TF-IDF.

From this analysis, we conclude that the best results on the IMDB dataset were achieved using the algorithms with TF-IDF, especially when combined with GloVe embeddings.

TABLE III. IMDB RESULTS

Feature Extraction Techniques	Algorithm	Accuracy	Precision	Test error	Loss	Recall	F1
Glove embeddings	SNN	0.74	0.76	0.25	0.53	0.71	0.73
	CNN	0.84	0.88	0.16	0.44	0.76	0.82
	LSTM	0.85	0.85	0.14	0.33	0.85	0.85
	CNN→LSTM	0.84	0.84	0.15	0.36	0.85	0.84
TF-IDF	SNN	0.88	0.87	0.11	0.38	0.89	0.88
	CNN	0.88	0.88	0.11	0.52	0.90	0.89
	LSTM	0.87	0.88	0.12	0.60	0.86	0.87
	CNN→LSTM	0.88	0.86	0.11	0.37	0.90	0.88
TF-IDF+Glove	SNN	0.88	0.87	0.11	0.37	0.89	0.88
	CNN	0.88	0.86	0.11	0.10	0.91	0.88
	LSTM	0.88	0.85	0.11	0.27	0.92	0.88

Table IV presents the results obtained using the Amazon dataset with various algorithms such as SNN, CNN, and LSTM, and applying both GloVe embeddings and TF-IDF for data vectorization. Using the SNN algorithm, we achieved an accuracy of 0.76 with GloVe embeddings, 0.81 with TF-IDF, and 0.81 when combining both GloVe and TF-IDF. For the CNN algorithm, the accuracy was 0.81 using GloVe embeddings, 0.80 with TF-IDF, and 0.81 with both methods combined. With LSTM, the results were 0.83 for GloVe, 0.80 for TF-IDF, and 0.82 for the combined approach. Lastly, the hybrid CNN-LSTM model gave accuracies of 0.81 using both GloVe and TF-IDF.

From this analysis, we conclude that the best results on the Amazon dataset were achieved with the LSTM algorithm, particularly when using TF-IDF and GloVe embeddings.

TABLE IV. AMAZON RESULTS

Feature Extraction Techniques	Algorithm	Accuracy	Precision	Test error	Loss	Recall	F1
Glove embeddings	SNN	0.76	0.78	0.23	0.49	0.73	0.75
	CNN	0.81	0.81	0.18	0.49	0.80	0.81
	LSTM	0.83	0.87	0.16	0.36	0.77	0.81
	CNN→LSTM	0.81	0.81	0.18	0.42	0.83	0.82
TF-IDF	SNN	0.81	0.82	0.18	0.51	0.80	0.81
	CNN	0.80	0.80	0.20	0.63	0.81	0.80
	LSTM	0.80	0.79	0.19	0.65	0.81	0.80
	CNN→LSTM	0.81	0.82	0.18	0.48	0.82	0.81
TF-IDF+Glove	SNN	0.81	0.81	0.18	0.48	0.81	0.81
	CNN	0.81	0.85	0.18	0.19	0.77	0.81
	LSTM	0.82	0.82	0.17	0.39	0.84	0.83

Figs. 11–13 present a comparative analysis of the algorithms SNN, CNN, and LSTM using different techniques such as TF-IDF and GloVe embeddings on the IMDB dataset. The comparison is based on several evaluation metrics: Accuracy, Precision, Test Error, Loss, Recall, and F1 Score. These metrics provide a comprehensive view of each algorithm's performance and effectiveness in sentiment analysis tasks. Figs. 14–16 compare the results on Amazon Dataset.

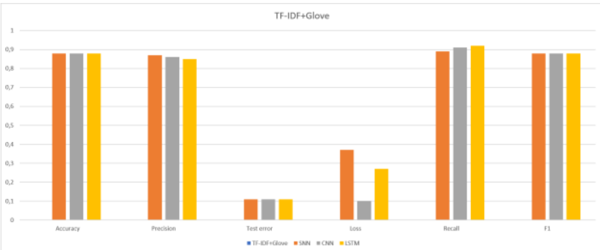

Fig. 12. TF-IDF+Glove using SNN, CNN and LSTM on IMDB.

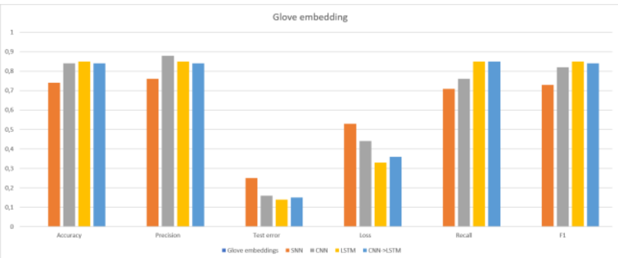

Fig. 11. Glove embedding using SNN, CNN and LSTM on IMDB.

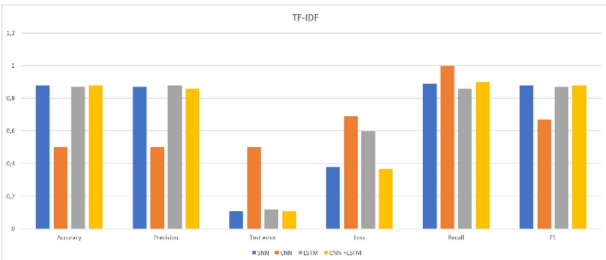

Fig. 13. TF-IDF using SNN, CNN and LSTM on IMDB.

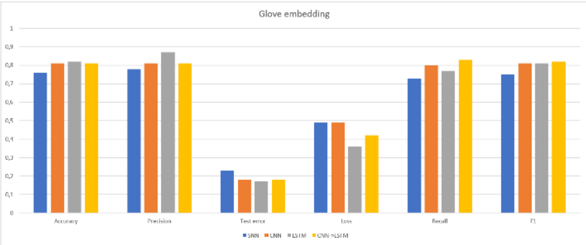

Fig. 14. Glove embedding using SNN, CNN and LSTM on Amazon.

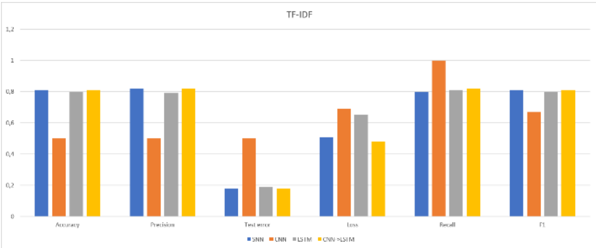

Fig. 15. TF-IDF using SNN, CNN and LSTM on Amazon.

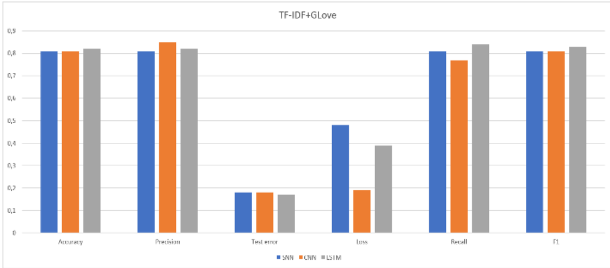

Fig. 16. TF-IDF using SNN, CNN and LSTM on Amazon.

Tables V and VI summarize the results of combining sentiment analysis techniques, specifically LSTM and CNN, with GloVe and TF-IDF embeddings alongside the Alternating Least Squares (ALS) algorithm.

Figs. 17–19 illustrate also a comparison of sentiment analysis methods integrated with the ALS algorithm and their respective impacts on the performance of the recommendation system.

TABLE V. THE RESULTS OF COMBINING ALS WITH SENTIMENT ANALYSIS

Evaluation metrics	ALS	ALS+LSTM_glove	ALS+CNN_TF-IDF_glove	ALS+TF-IDF_CNN
RMSE	0.58	0.56	0.56	0.57
MSE	0.33	0.31	0.31	0.32
MAE	0.15	0.14	0.14	0.14
R-Squared	0.8	0.77	0.8	0.8

TABLE VI. THE RESULTS OF COMBINING ALS WITH SENTIMENT ANALYSIS

Evaluation metrics	ALS+LSTM_TF-IDF_glove	ALS+CNN_glove	ALS+CNN_TF-IDF
RMSE	0.58	0.59	0.55
MSE	0.34	0.34	0.31
MAE	0.14	0.15	0.14
R-Squared	0.8	0.8	0.8

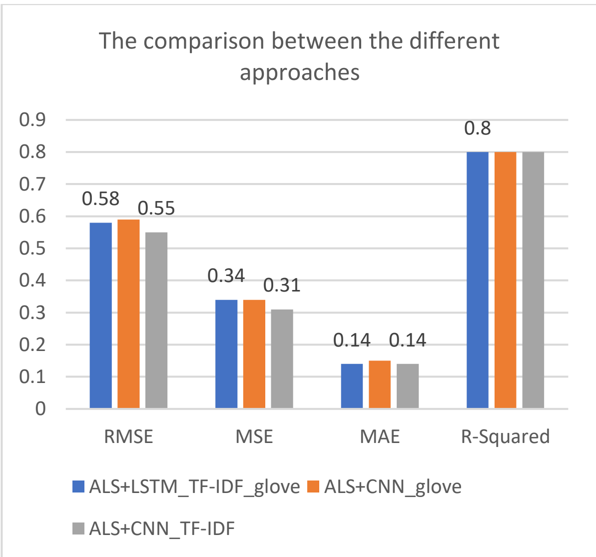

Fig. 17. The comparison between the different SA approaches with CF.

The analysis indicates a notable improvement in performance when sentiment analysis is applied. Specifically, the RMSE value decreased from 0.58 to 0.55 when using the combination of ALS with CNN and TF-IDF embeddings. This suggests that integrating sentiment analysis not only refines the recommendations but also enhances the overall predictive accuracy of the system.

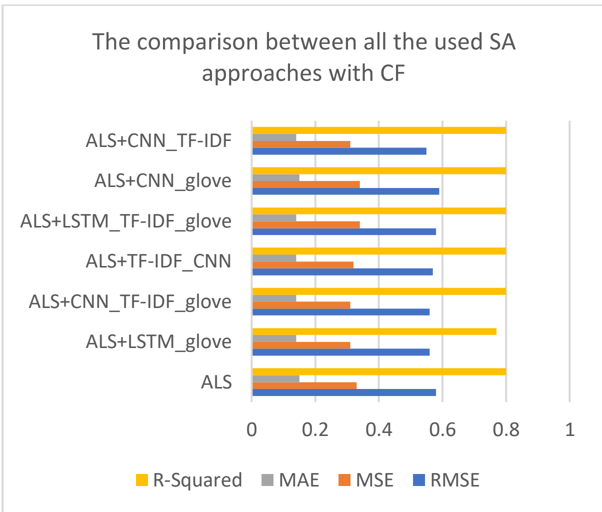

Fig. 18. The comparison between the different SA approaches with.

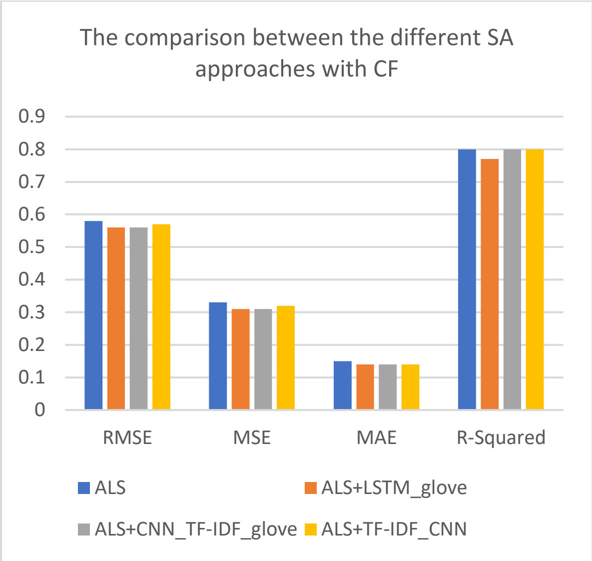

Fig. 19. The comparison between the different SA approaches with CF.

VI. Conclusion

In this article, we proposed a recommender system that leverages sentiment analysis to enhance recommendation outcomes and address the cold start problem faced by new users. Our approach combines TF-IDF and GloVe embeddings with three deep learning algorithms—SNN, CNN, and LSTM networks—to improve sentiment analysis predictions. The system architecture consists of two key modules: the first provides recommendations based solely on sentiment analysis during a user's initial interaction, while the second integrates sentiment analysis with collaborative filtering using the ALS algorithm.

The proposed approach was evaluated using the IMDB, Amazon, and Amazon Fine Foods datasets, demonstrating that the combination of TF-IDF and GloVe embeddings significantly enhances sentiment analysis performance. Additionally, the integration of sentiment analysis with the ALS algorithm improved overall recommendation results, effectively addressing the cold start problem by generating quality recommendations based on sentiment predictions. And solve scalability problem.

While our approach shows strong performance, it leaves room for exploring emerging deep learning models such as Transformers and GNNs. These models could capture complex dependencies and graph-based relationships, potentially further improving recommendation accuracy. Future work will focus on applying the proposed method to other recommendation algorithms and methodologies, as well as investigating the computational trade-offs and potential benefits of advanced alternatives like BERT, Word2Vec, and other state-of-the-art techniques. This exploration will aim to balance model complexity, training time, and performance to push the boundaries of sentiment-based recommender systems.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

SN led the problem identification, solution development, concept design, data collection, application, results analysis, resource acquisition, and investigation; the supervision and review were conducted by MYC and JEB; all authors had approved the final version.

Acknowledgment

The work is done through the support of the CNRST in part merit scholarship of research.

References

[1] S. Nabil, J. Elbouhdidi, and M. Y. Chkouri, "Recommendation system based on data analysis-application on tweets sentiment analysis," in *Proc. 2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, IEEE, 2018, pp. 155–160. doi: 10.1109/CIST.2018.8596418

[2] A. K. A. Hassan and A. B. A. Abdulwahhab, "Reviews sentiment analysis for collaborative recommender system," *Kurdistan Journal for Applied Research*, vol. 2, no. 3, 2017.

[3] N. A. Osman, S. A. M. Noah, M. Darwich *et al.*, "Integrating contextual sentiment analysis in collaborative recommender systems," *PLoS One*, vol. 16, no. 3, e0248695, 2021. doi: 10.1371/journal.pone.0248695

[4] E. Asani, H. V. Nejad, and J. Sadri, "Restaurant recommender system based on sentiment analysis," *Machine Learning with Applications*, vol. 6, 2021.

[5] B. T. Hung, "Integrating sentiment analysis in recommender systems," in *Reliability and Statistical Computing*, H. Pham, Eds. Springer, Cham. 2020, pp. 127–137. doi: 10.1007/978-3-030-43412-0_8

[6] C. Hsu and A. H. Liao, "Sentiment-based chatbot using machine learning for recommendation system," *Springer Soft Computing*, 2022.

[7] C. N. Dang, M. N. Moreno-García, and F. D. la Prieta, "An approach to integrating sentiment analysis into recommender systems," *Sensors*, vol. 21, no. 16, p. 5666, 2021. doi: 10.3390/s21165666

[8] I. Karabila, N. Darraz, A. El-Ansari *et al.*, "Enhancing collaborative filtering-based recommender system using sentiment analysis," *Future Internet*, vol. 15, no. 7, 235, 2023. doi: 10.3390/fi15070235

[9] C. N. Dang, M. N. Moreno-García, and F. De la Prieta, "Using hybrid deep learning models of sentiment analysis and item genres in recommender systems for streaming services," *Electronics*, vol. 10, 2459, 2021. doi: https://doi.org/10.3390/electronics10202459

[10] N. A. Al-Rossais, "Improving cold start stereotype-based recommendation using deep learning," *IEEE Access*, vol. 11, pp. 145781–145791, 2023.

[11] A. Putri, Z. K. A. Baizal, and D. Rischasdy, "Book recommender system using convolutional neural network," in *Proc. 2022 International Conference on Advanced Creative Networks and Intelligent Systems (ICACNIS)*, 2022, pp. 1–6.

[12] J. W. Choi, S. K. Yun, J. B. Kim, "Improvement of data sparsity and scalability problems in collaborative filtering based recommendation systems," in *Proc. International Conference on Applied Computing and Information Technology*, 2020. doi: https://doi.org/10.1007/978-3-030-25217-5_2

[13] A. A. Awan. (Nov, 2024). What is Tokenization? Types, Use Cases, Implementation. *DataCamp*. [Online]. Available: https://www.datacamp.com/blog/what-is-tokenization

[14] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. International Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[15] S. Qaiser and R. Ali, "Text mining: use of TF-IDF to examine the relevance of words to documents," *Int. J. Comput. Appl.*, vol. 181, no. 1, pp. 25–29, 2018. doi: 10.5120/ijca2018917395

[16] H. Xingwei, "Sorting big data by revealed preference with application to college ranking," *J. Big Data*, 2020.

[17] G. Guney, B. O. Yigin, N. Guven *et al.*, "An overview of deep learning algorithms and their applications in neuropsychiatry,"

Clinical Psychopharmacology and Neuroscience, vol. 19, no. 2, pp. 206–219, 2021. doi: 10.9758/cpn.2021.19.2.206

[18] D. Durstewitz, G. Koppe, and A. Meyer-Lindenberg, "Deep neural networks in psychiatry," *Mol. Psychiatry*, vol. 24, no. 11, pp. 1583–1598, 2019. doi: 10.1038/s41380-019-0365-9

[19] GeeksforGeeks. (June, 2024). Deep learning. Introduction to long short-term memory. [Online]. Available: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/

[20] Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint, arXiv:1408.588, 2014.

[21] Y. Zhang and B. C. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," arXiv preprint, arXiv:1510.03820, 2017.

[22] R. Sharedalal, "Amazon fine food reviews—Design and implementation of an automated classification system," Master thesis, University the Graduate School College of Engineering, The Pennsylvania State, USA, 2019.

[23] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009. doi: 10.1109/MC.2009.263

[24] Apache Spark MLlib documentation. Collaborative Filtering with ALS. [Online]. Available: https://spark.apache.org/docs/latest/ml-collaborative-filtering.html

[25] Kaggle. (March, 2019). IMDB dataset of 50K movie reviews. [Online]. Available: https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

[26] Kaggle. (Nov., 2019). Amazon reviews dataset. [Online]. Available: https://www.kaggle.com/datasets/bittlingmayer/amazonreviews

[27] Kaggle. (May, 2017). Amazon fine food reviews dataset. [Online]. Available: https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews