International Journal of Mathematical, Engineering and Management Sciences
Vol. 10, No. 6, 2223-2247, 2025
https://doi.org/10.33889/IJMEMS.2025.10.6.103

Ram Arti Publishers

SAFPRS: Novel Framework of Sentiment Analysis for Lifestyle Product Recommendation System

Ritu Rajal
Department of Computer Science and Engineering,
Faculty of Engineering and Technology, Gurukula Kangri (Deemed to be University), Haridwar, Uttarakhand, India.
Corresponding author: 20631002@gkv.ac.in

Nishant Kumar
Department of Computer Science and Engineering,
Faculty of Engineering and Technology, Gurukula Kangri (Deemed to be University), Haridwar, Uttarakhand, India.
E-mail: nishant@gkv.ac.in

Sanjeev Kumar
Department of Master of Computer Applications,
G. L. Bajaj Institute of Technology and Management, Greater Noida, Uttar Pradesh, India.
E-mail: sanjeevnmc83@gmail.com

(Received on April 14, 2025; Revised on July 2, 2025; Accepted September 8, 2025)

Abstract
Sentiment analysis is a significant tool for evaluating opinions in e-commerce applications since it is used to categorize customer feedback according to positive, neutral, and negative sentiments. These feelings are very useful in helping future customers to make the right decisions when shopping. This paper introduces a novel framework that employs two distinct deep learning-based approaches: For this purpose, we divide the general sentiment analysis into two categories which include the Sentiment Analysis with Enhanced Features (SAEF) and the Sentiment Analysis with Filtered Features (SAFF). Both models classify product reviews from which the sentiment polarity (positive, negative, or neutral) is derived and used for a recommendation system. These models work under the given framework and when the same dataset is used, separate customized recommendations for specific products are produced, emphasizing the fact that the framework outperforms the traditional methods. A key novelty of the proposed approach is that both models exhibit over 97% accuracy and this clearly shows how the approach improves on the recommendations and benefits the users greatly.

Keywords- Sentiment analysis, Deep learning, Long short-term memory, Collaborative filtering, Product recommendation, Product reviews.

1. Introduction

Electronic commerce plays an important role in the field of information technology that has become a cornerstone of modern life in recent years and is reforming how consumers interact with products and services. E-commerce offers several benefits (Taher, 2021) to consumers, such as time savings, access to a wide range of products and information, convenience of 24/7 shopping and easy price comparisons. Still, there are certain limitations like lack of personal interaction, the inability to physically inspect products before purchasing, risks of damage during transportation, and potential delays in delivery. Despite these challenges, more people are turning to e-commerce (Pandey et al., 2023) for their daily needs during the COVID-19 pandemic, with significantly enhanced adoption of online shopping. To address the limitations of e-commerce, customer feedback in the form of reviews, ratings and images has become steadily more important. Reviews may be their positive and negative emotions that can be referred as sentiments (Sharma et al., 2022). These sentiments can be integrated with recommender system (RS) (Liu et al., 2021; Mu, 2018) to enhance the product recommendations. The purpose of RS is to produce pertinent recommendations for a group of users for some items or products which might suit their curiosity. Recommendations of books on Amazon, or movies on Netflix, are instantiations of the RS. RS are of two types: Content based RS and Collaborative RS. Recommender systems typically recommend to the user items that he/she has liked in the past, or items he/she is currently examining. Later, RS accumulates ratings or recommendations (Khapre et al., 2023) on objects and then compares users based on ratings and provides fresh recommendations based on inter-user similarities.

Sentiment analysis (SA) plays an important role in enhancing RS by incorporating user feedback into the recommendation process by extracting emotions from text data. There are many approaches (Dang et al., 2020) exist for sentiment analysis like lexicon-based approach, machine learning (ML) (Ongsulee, 2017) approach and hybrid approach shown in **Figure 1**. This paper specifically focuses on a methodology belonging to a technical area known as deep learning (DL) (Mu, 2018), that forms a subcategory of the larger field of ML and that has had significant success in areas like speech and image recognition, Artificial Neural Networks (ANN) and Natural Language Processing (NLP).

Automated feature extraction and hyperparameter tuning are significant advantages of deep learning techniques over traditional ML methods. DL models achieve higher accuracy and performance in SA through a multilayer approach in the depth of NN, thereby supporting integration for RS. Many solutions were reported in previously conducted studies. For instance, to conduct topic-based SA, DNN along with the document embedding approaches has been used to explore the sentiment distribution in a document with respect to the topics of interest (Seilsepour et al., 2023). Yet, as has already been pointed out, categorising complex or ambiguous sentiments can be difficult for these models. Also, the different DL approaches have been implemented to categorise sentiments in product reviews in an attempt to enhance efficiency and effectiveness (Kuang et al., 2023) while at same time may raise the cost of computation and complicate the system.

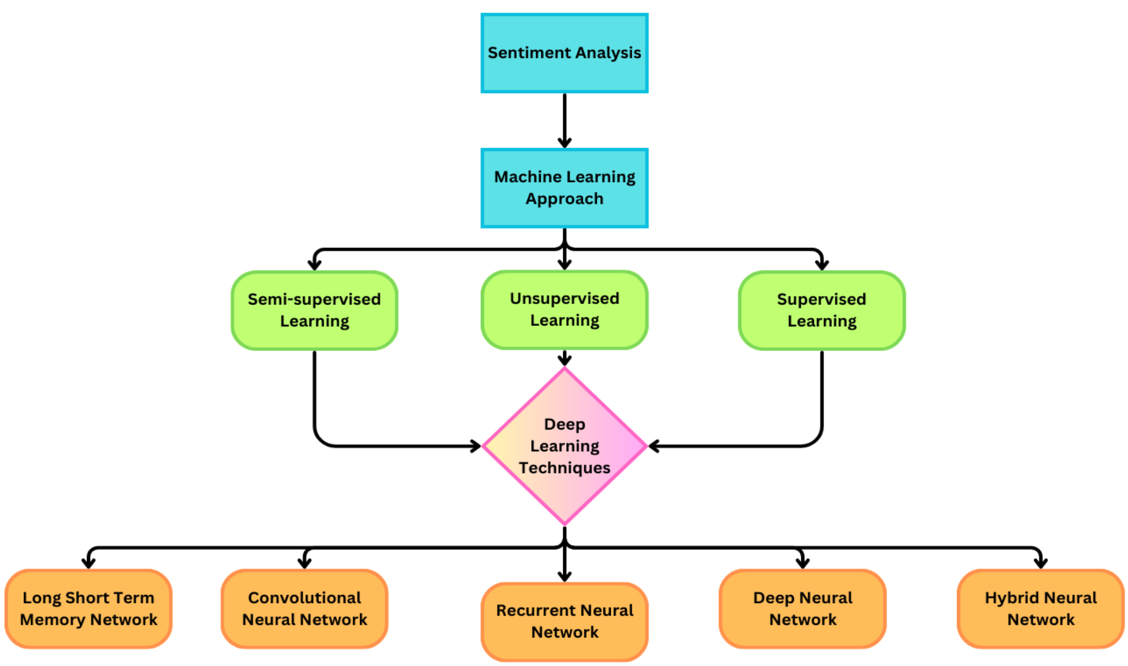

Figure 1. Taxonomy of sentiment analysis.

Some of the methods used for sentiment analysis include Hamiltonian DNNs (Ajmeera & Kamakshi, 2024) and combination of models such as BERT, BiGRU and Softmax layer (Liu et al., 2020), which effectively capture sentiments in product reviews. Various approaches integrate SA with DL-based RS to improve recommendation accuracy and better understand user preferences (Dang et al., 2021; Elahi et al., 2023; Osman et al., 2021; Thomas & Jeba, 2024). After analyzing previous research and comparing different models, Long Short-Term Memory (LSTM) networks (Dhola & Saradva, 2021) have been selected as the most effective for sentiment analysis in the suggested recommender system. The proposed framework consists of two distinct models that integrate sentiment analysis with a collaborative filtering recommendation system.

1.1 Novelty and Research Contributions of SAFPRS

In this paper, the authors propose a new hybrid method, which incorporated dual deep learning models with collaborative filtering to achieve a sentiment-based lifestyle recommendation of products. This proposed architecture incorporates two domain-specific sentiment analysis models namely SAEF and SAFF where each is trained to learn different levels of sentiment data. SAEF pays attention to linguistic and contextual sentiment indicators extracted the feature in the form of customer review and improves the identification of finer emotional expressions and inaccuracies. By contrast, SAFF addresses attribute-specific emotions through analyzing fixed product parameters (size, quality, color of the products, etc.). The two models outputs are then wisely combined and merged with collaborative filtering to produce individual-oriented top-n product suggestions based on the item and user relationships. The framework offers a combination of fine-grained sentiment analysis and collaborative filtering in an effort to not only increase the accuracy of classification but also make the recommendations more relevant. Contextuality in collaborative filtering combining with this dual-path sentiment extraction will provide a very unique and effective method of building sentiment-aware recommending systems in the lifestyle product domain.

The proposed model addresses critical gaps in the existing literature, particularly in handling sentiment variability and improving recommendation personalization, while the demonstrated performance gains suggest substantial potential for real-world deployment in commercial platforms.

The rest of this paper is structured as follows: Introduction is provided in Section 1. Section 2 overviews the work with regard to sentiment analysis. The methodology used in our proposed model is explained in detail in section 3. Section 4 explained the evaluation metrices used. The results are highlighted in Section 5 of the paper, and Section 6 provided the conclusion of the paper.

2. Related Work

Sentiment analysis (SA), or opinion mining (OM), has become increasingly important in enhancing the effectiveness of recommender systems (RS), particularly for lifestyle products. However, despite the advances in this field, several challenges remain. This discussion synthesizes relevant research contributions that inform our proposed framework for sentiment analysis in a lifestyle product recommendation system.

The main challenge in SA constitutes extracting useful sentiments from different data sources particularly when we deal with imbalanced datasets such as those found in social network data. This issue is addressed by the study Kuang et al. (2023) by introducing hybrid deep learning methods for SA in product reviews. By supporting a Recurrent Neural Network (RNN) for sentiment classification and using a resampling technique to balance the Amazon dataset across four product categories, the study showed improved performance. The hybrid model approach gives us insights for our framework, where balancing data in lifestyle product reviews is critical for accurate sentiment analysis.

Accuracy in sentiment analysis remains a crucial focus. The Embedding Topic Sentiment Analysis using Deep Neural Networks (ETSANet) (Seilsepour et al., 2023) represents an important advancement in this area by enhancing accuracy through the identification of semantically related documents to specific topics. The introduction of a Semantic Topic Vector and a hybrid CNN-GRU model allowed for attaining deeper semantic aspects, which could be especially beneficial in the context of lifestyle products where user preferences are often meticulous.

To improve product recommendations, another promising direction is to integrate sentiment analysis with collaborative filtering (CF). A study, Thomas & Jeba (2024) proposed a framework that combined an LSTM model for sentiment analysis with CF techniques for e-commerce platforms. The approach substantially improves recommendation accuracy and consumer satisfaction by integrating SA with CF. For lifestyle product recommendations, this integration is relevant for our framework which often requires a personalized approach that considers both sentiment and CF.

By incorporating the Adaptive architectures with advanced feature extraction methods used in DL, a study Dang et al. (2021) utilized BERT for text transformation, combined with CNN and LSTM models for sentiment classification. DL models integrate with sentiment analysis can significantly enhance RS performance by applying this approach to Amazon datasets like food and movie reviews. Similarly, our framework aims to leverage such adaptive architectures to better understand user sentiments toward lifestyle products.

To improve user satisfaction, further revolutions in SA focused on enhancing sentiment classification by including a Hamiltonian Deep Neural Network (DNN)-based approach (Ajmeera & Kamakshi, 2024). The method included adaptive self-organizing filter followed by feature extraction and classification with the help of Hamiltonian DNN. This approach provides important information on how to improve SA in lifestyle product reviews given challenges like the variability in sequence length and complex review dynamics.

Another field of interest is the study of the correlation between user's attitude and his/her ratings. In a hybrid recommender system (Elahi et al., 2023), this relationship was studied in detail to identify how other aspects of user preferences could be revealed with the help of sentiment analysis which are often hidden in a rating based approach.is could expose various aspects of user preferences not captured by traditional rating-based systems. This perception is rather valuable especially for lifestyle products, in which recipients' attitude may significantly vary depending on personal experiences.

Addressing challenges like cold-start and data sparsity, contextual sentiment analysis has proven to be a valuable tool in collaborative RS. A study, Osman et al. (2021) suggested a sentiment-based model that used contextual information to improve recommendation quality across multiple domains. The results exhibited that this model surpassed traditional CF methods, underscoring the importance of incorporating domain-specific sentiments in our lifestyle product recommendation framework.

A research work, Liu et al. (2020) that proposed a BERT-BiGRU-Softmax for sentiment classification pointed out the relevance of SA in e-commerce. The inspected model achieved up to 95.5% of accuracy on the large dataset of product reviews compared to other DL models such as RNN and BiLSTM. It is regrettable that the current work could not undertake a more precise categorisation of lifestyle products, but in any future research, it is recommended to employ more sophisticated neural network architecture that yielded higher accuracy for sentiment classification within our framework.

In the field of SA, other learning techniques using supervised ML have also been reviewed. Another (Sultan, 2022) work used Naive Bayes, Logistic Regression, Support Vector Machine (SVM) and Ensemble Classification on the corpus of Amazon product reviews to arrive at Ensemble Classification with the highest accuracy. These research findings imply that ensemble methods may also provide reliable performance in SA for lifestyle products.

There is another challenge solved using a Bidirectional LSTM network (Rehman et al., 2019) i.e. analyzing long term dependencies in text especially in IMDB and Amazon movie reviews dataset. It was proved during the experiment that the application of Hybrid CNN-LSTM makes it possible to obtain great precision, the rate of the given options and accuracy which can be useful in treating large and complex lifestyle product reviews.

For the purpose of the current study, automated sentiment classification was established by models comprising of BERT, LSTM, and Bidirectional LSTM (Kumar et al., 2023). The research acknowledged a higher accuracy level in BERT than other models when it comes to precision, recall, F1-score, and proves that BERT can be used for precision SA in lifestyle products recommendation.

Thus, for the purpose of maintaining model robustness, overfitting is a major issue that must be resolved in sentiment analysis. To overcome this, a Bi-LSTM model (Abdillah et al., 2020) with dropout parameters and activity regulation was proposed for this purpose, i.e., used on a database of song lyrics. Several future works were recommended and the study proposed the use of other datasets and layers such as the attention, convolution and pooling, which can also extend to this theory to enhance the efficacy.

The author, Gao et al. (2025) presented a hybrid movie recommendation system that used a DL methods and graph-based methods to precisely predict user interests. It integrates a CNN to analyze the movie contents and PageRank algorithm to assess the significance of movies through the user browsing habits. The system gives 7.15% precision improvement and 5.19% recall improvement compared to the system based on 215 user data in 508 IMDb movie pages.

The recommendation system, Krishna et al. (2025) involved multi-layer attention-based encoder-decoder temporal CNN to make a prediction of sentiments and Modified Conditional Generative Adversarial Network (GAN) to solve class imbalance. The feature extraction is carried out by means of a log-term frequency-based modified inverse class frequency (LTFMICF) algorithm and the hyperparameters of the model are optimized by Ocotillo Optimization Algorithm. The model is more accurate, more precise, and has better recall as well as F1-score and AUC compared to those state-of-the-art systems.

A novel recommendation model called as DADRL (Gheewala et al., 2025) that overcomes the shortcomings of conventional reviews-based systems. It employed semi-supervised latent Dirichlet allocation model to learn sentiment and aspect-level topics and BERT as the method of generating contextualized embeddings. The dynamic modulator-based Long Short-Term Memory in DADRL simulates the sentiment changes in review sequences, and attentional fusion mechanism makes topical and global representations coherent. The new architecture of the model manages to retain contextual richness and capture a variety of user sentiments making it better at predicting rating tasks.

The author, Awati et al. (2025) presented a hybrid system, HODL that combined both top-N item recommendations with SA and DL. Its architecture is referred to as HOCLA_CBiGRU and is an innovative system used to identify user sentiments and item ranking improvement. It begins with a series of strenuous pre-processing procedures, where semantically contextual features are to be extracted through LTFMICF and GloVe embeddings. A modified mud ring algorithm is used as well in the system to optimize the hyperparameter. This model performed better than a number of current methods, and when used to recommend the top-10 items had a RMSE of 0.69.

The new sentence-level SA framework proposed (Atlas et al., 2025) in the research paper offered a new framework that employed the NLP strategies along with the DL networks and is aimed at enhancing the online product review mining. The model is based on scraping the web page, preprocessing of the data, and feature extraction, a Bidirectional Gated Recurrent Unit (BiGRU) network. A hybrid RNN-based LSTM classifier is used to classify sentiments as either positive, negative or other. The model also does better than conventional lexicon-based ones in representations of sarcastic and context-sensitive language. It can be used practically in market research, brand monitoring or in recommendation systems.

Despite the above advances, there are still some challenges in sentiment analysis, especially in equating a particular rating to its equivalent review. For example, anomalies between high ratings and unhealthy reviews can pose a problem in recommendations which might have been based almost entirely on ratings. This research gap leads to the provision of our proposed framework, namely, a novel framework of sentiment analysis for lifestyle product recommendation system which explores the feature extraction technique through LSTM model for sentiment analysis and the system incorporation of collaborative filtering recommender system. This approach aims to give better and more accurate lifestyle product recommendation information compared with the weaknesses highlighted in several studies.

3. Methodology

An innovative solution based on developing a deep learning algorithm & implementing two approaches including SAEF as well as SAFF as successful components for the development of an efficient recommendation system that was constructed to be functional as well as scalable for managers. This framework is subdivided into three individual phases.

The first step carried out was data extraction and data cleansing. Customer reviews were selected from a featured data set of Amazon fashion along with tokenization, filtering out stop words and normalization of reviews data.

The second stage identified is where sentiment analysis models were created. Two distinct algorithms were designed within the framework: SAEF and SAFF. SAEF extracts discriminating features from the context and sensibility of sentiments from customer's reports, misclassification of sentiments as positive, negative or neutral. On the other hand, the features utilized in SAFF include quality, size, and color, which are input parameters that are fixed from the dataset so as to work the sentiment classification. Such algorithms allow understanding customers' overall impressions referring to particular characteristics of the product offered.

In the third stage, the outputs of the sentiment analysis models were integrated with Collaborative Filtering (CF) to enhance recommendation accuracy. CF was applied to build both item-based and user-based recommendation systems. The recommendation engine utilizes sentiment polarity from the SAEF and SAFF algorithms to recommend the top-n products that are most likely to align with user preferences. By leveraging both sentiment analysis and collaborative filtering, the framework offers a more personalized recommendation system. The overall framework, as shown in **Figure 2**, provides an advanced mechanism for identifying products based on user sentiments, demonstrating a significant improvement in recommendation accuracy, as outlined in Elahi et al. (2023) and **Table 1** denotes the notations used in proposed framework.

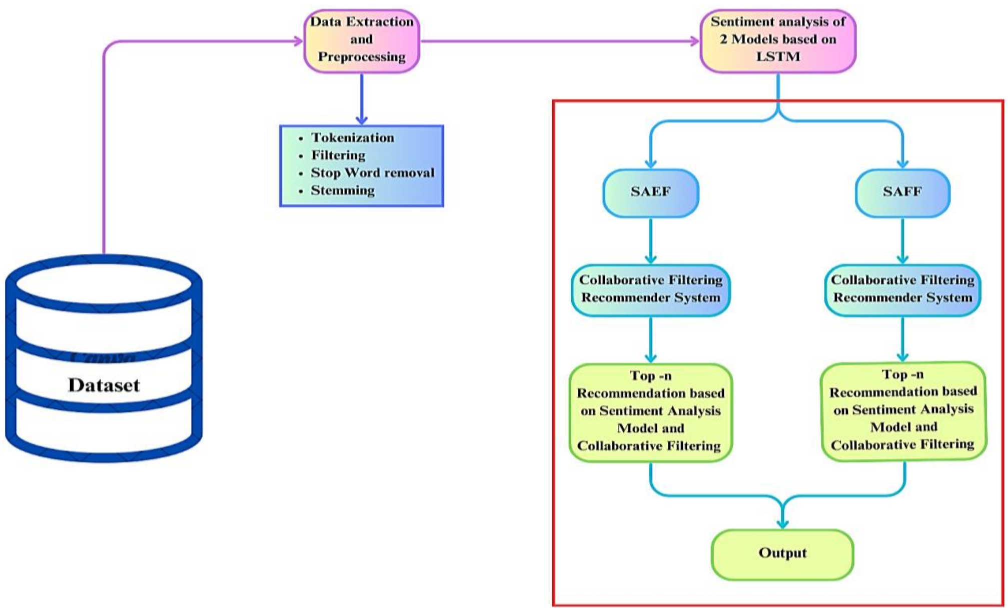

Figure 2. Proposed novel framework.

Table 1. Notations used in proposed framework.

Notation	Meaning
$X \in R^{n \times t}$	Input data where *n* is the number of reviews and *t* is the maximum sequence length.
$x_{i,j}$	The *j*-th word (token) in the *i*-th review.
f_{embed}	Embedding function that maps each token to a 32-dimensional vector.
$v_{i,j} \in R^{32}$	32-dimensional vector representing the embedding of the *j*-th word in the *i*-th review.
$V \in R^{n \times t \times 32}$	The matrix after embedding, representing the entire dataset.
$h_T \in R^{32}$	The hidden state of the LSTM layer at time step *t*.
$f_{LSTM(V_t,h_{t-1})}$	LSTM recurrence function that computes the hidden state based on the current input and the previous hidden state.
y	The output of the dense layer, representing the probability of a sentiment class.
$W_d \in R^{1 \times 32}$	Weight matrix of the dense layer.
$b_d \in R$	Bias term of the dense layer.
$\sigma(x) = 1/(1 + e^{-x})$	Sigmoid activation function.
$y = \sigma(W_d h_T + b_d)$	Formula for computing the output of the dense layer, where h_T is the final LSTM hidden state.
$r_{u,i}$	Rating given by user *u* to item *i*, $r_{u,i} \in [1,5]$.
D_{train}	Training set, a subset of *R*, used to fit the SVD model.
D_{test}	Testing set, a subset of *R*, used to evaluate the SVD model.
P_u	Latent feature vector for user *u*.
Q_i	Latent feature vector for item *i*.
$\hat{r}_{u,i}$	Predicted rating for user *u* and item *i* using SVD.
M	Global average rating across all users and items.
b_u	Bias associated with user *u*.
b_i	Bias associated with item *i*.
$S(i_j)$	Average sentiment score for item i_j, derived from sentiment analysis.
$Estimate(i_{target}, i_j)$	Final estimated score for item i_j for the target item i_{target}, combining collaborative filtering and sentiment analysis.
$Top - 10\ \left(Estimate(i_{target}, i_j)\right)$	The top 10 recommended items based on the estimated score.
RMSE	Root mean squared error, used to measure the performance of the recommendation system.
x_t	Word at time step *t* (word index).
X	Input sequence of words ($x_1, x_2, \ldots, x_t$).
e_t	Embedding vector for word x_t , $e_t \in R^d$.

Table 1 continued…

D	Dimensionality of the embedding vectors (here d=32).
E	Matrix of embedded word vectors $E=(e_1,\ldots,e_{100})$.
Σ	Sigmoid activation function $\sigma(z)=1/1+e^{-z}$.
W_i, U_i	Weight matrices for the input gate.
W_f, U_f	Weight matrices for the forget gate.
W_o, U_o	Weight matrices for the output gate.
W_c, U_c	Weight matrices for the cell state.
b_i, b_f, b_a, b_c	Bias terms for the gates and cell state.
h_{100}	Final hidden state of the LSTM after processing the entire sequence, $h_{100} \in R^{32}$.
W_d	Weight matrix of the dense output layer $W_d \in R^{1\times 32}$.
y	Sentiment score $y = \sigma(W_d h_{100} + b_d)$.
$x \to E \to h_{100} \to y$	Flow of data through the embedding, LSTM, and dense output layers.

3.1 Dataset

Amazon is one of the largest online markets with the largest number of reviews available for any product. The Amazon fashion review dataset used as an example was obtained via Amazon (Amazon review data (ucsd.edu)). The dataset that is being offered includes 28 different Amazon Standard Identification Numbers (ASIN) and 3045 reviews covering a broad spectrum of categories. **Table 2** tabulates the dataset's attribute description. These attributes are essential in representing lifestyle interests' trends in preferences as the users tend to comment on these features elements such as fit, color attractiveness, comfort, and usage occasion (e.g., lounge wear, work, outdoor), which corresponds to lifestyle choices. The designated attributes are the only ones that the SA model can analyse. **Figure 3** shows the sample data that the SA model was based on.

Table 2. Amazon fashion dataset attribute.

Attribute Title	Description
reviewerID	Reviewer identifier
Asin	Product identifier
reviewerName	Name of the reviewer
reviewText	Text of the review
overall	Rating of the product
summary	Summary of the review
unixReviewTime	Time of the review (UNIX time)
reviewTime	Time of the review (raw)
vote	Review helpful vote

	overall	verified	reviewTime	reviewerID	asin	style	reviewerName	reviewText	summary	unixReviewTime	vote
0	5	True	09 4, 2015	ALJ66O1Y6SLHA	B000K2PJ4K	{'Size:': ' Big Boys', 'Color:': ' Blue/Pink'}	Tonya B.	Great product and price!	Five Stars	1441324800	NaN
1	3	True	05 6, 2015	A3W11493KS6Z2L	B000K2PJ4K	{'Size:': ' Little Boys', 'Color:': ' White/Bl...	NaeNae	Waaay too small. Will use for futur children!	Oops!	1430870400	NaN
2	5	True	05 6, 2015	A3W11493KS6Z2L	B000K2PJ4K	{'Size:': ' Little Boys', 'Color:': ' Blue/Ora...	NaeNae	Stays vibrant after many washes	Great	1430870400	NaN
3	5	True	05 6, 2015	A3W11493KS6Z2L	B000K2PJ4K	{'Size:': ' Little Boys', 'Color:': ' Blue (37...	NaeNae	Stays vibrant after many washes	Good	1430870400	NaN
4	5	True	05 6, 2015	A3W11493KS6Z2L	B000K2PJ4K	{'Size:': ' Little Boys', 'Color:': ' Blue/Pink'}	NaeNae	My son really likes the pink. Ones which I was...	Great	1430870400	NaN
5	3	True	05 6, 2015	A3W11493KS6Z2L	B000K2PJ4K	{'Size:': ' Little Boys', 'Color:': ' Light Bl...	NaeNae	Waaay too small. Will use for future child.	Oops!	1430870400	NaN
6	2	True	01 25, 2018	A3HX4X3TIABWOV	B000KPIHQ4	{'Size Name:': ' Men's 6-6.5, Women's 8-8.5', ...	Denise A. Conte	Relieved my Plantar Fascitis for 3 Days. Then ...	These were recommended by my Podiatrist	1516838400	NaN
7	2	True	01 5, 2017	AW8UBYMNJ894V	B000KPIHQ4	{'Size Name:': ' Men's 8-8.5, Women's 10-10.5'...	Cognizant Consumer	This is my 6th pair and they are the best thin...	Not the same as all my other pairs.	1483574400	NaN
8	5	True	10 17, 2016	A265UZVOZWTTXQ	B000KPIHQ4	NaN	William_Jasper	We have used these inserts for years. They pr...	Great inserts	1476662400	NaN
9	5	True	08 22, 2016	AW8UBYMNJ894V	B000KPIHQ4	NaN	Cognizant Consumer	Pinnacle seems to have more cushioning so my h...	Personal favorite	1471824000	NaN

Figure 3. Sample dataset.

3.2 Data Pre-processing

Data pre-processing (Kuang et al., 2023; Mahadevaswamy & Swathi, 2022) is the process of transforming raw data via a number of methodical steps in order to make it ready for additional analysis. This critical stage seeks to increase the accuracy of the results, optimise the quality of the data, and expedite the entire analytical process. Raw data gathered from multiple sources frequently has noise, missing numbers, inconsistent information, and irrelevant data. The effectiveness of statistical studies and deep learning models can be severely hampered by these flaws. These problems are addressed by efficient data pre-processing methods such as reduction, normalisation, transformation, and cleaning of the data shown in **Figure 4**. Text data gets transformed through data processing procedures that prepare the data for activities such as sentiment analysis and text categorization to be efficient.

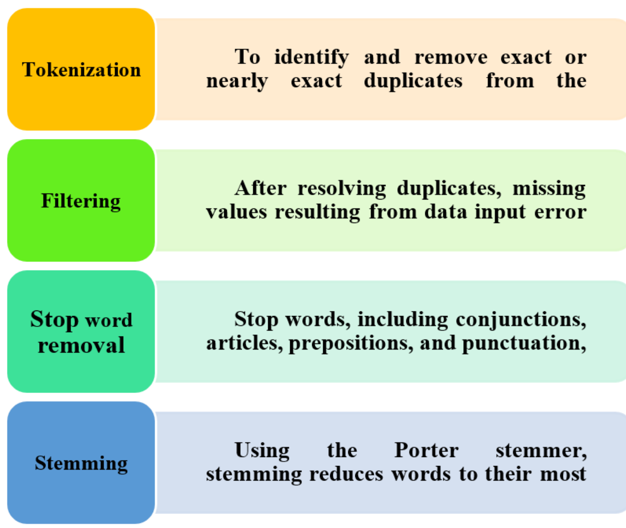

Figure 4. Data processing procedure.

3.3 Long Short-Term Memory (LSTM) for SA

After preprocessing the data, moved on to constructing the LSTM (Thomas & Jeba, 2024) network. The model was constructed layer by layer in a sequential fashion using the Sequential API provided by Keras. But before going to explain the different layers in our LSTM model, let us discussed the LSTM shown on **Figure 5** (Thomas & Jeba, 2024).

LSTM is a development of Recurrent Neural Networks (RNN). Neurons are concurrently interconnected in a directed cyclic manner in the case of RNN. The operation is done in a sequential mode in the RNN model because in the case of capturing a number of words or an input for a sequence, it necessitates the machine to have an internal memory. Since the output of the RNN model is determined by the preceding node inputs and the collected information, it has to perform the same operation for each element. In further operations, Equation (1) represents the terminology of the traditional RNN model, where h_t, f_w, h_t-1 and x_t represent the new state at time t, function with parameter w, previous state and input vector at time t, respectively.

$$h_t = f_w(h_t - 1, x_t) \tag{1}$$

We modified Equation (1) to (2) which is used to allocate weights

$h_t = tanh(W_h h_1 - W_{xh} x_t)$ (2)

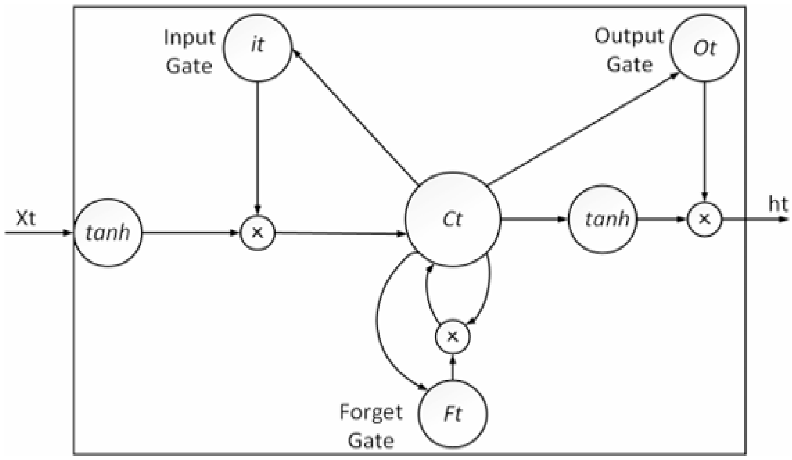

Figure 5. LSTM network.

where, tanh, W_h and x_t represent activation function, hidden weights and input vector respectively. The single hidden state of a traditional RNN makes difficult to learn long term dependencies for the network. This problem is addressed by LSTM by introducing a container that can hold information for long period that is referred as memory cell. LSTM networks are compatible for tasks such as language modelling, recommender systems and speech recognition due to capability of learning long-term dependencies in sequential data.

The input gate, the forget gate and the output gate are the three gates which is controlled by memory cell. What information is to be added, removed and to take the output from memory cell is decided by these gates. Input gate controls the information to be added to the memory cell of the computer. Located in the memory cell, the forget gate determines the information which needs to be erased. Depending of the output gate, it supplies the information to take output from the memory cell. Hence LSTM network allows to select, stored or ignore information passed through the network and in doing so long term dependencies are allowed by them.

It also has three gates for passing and storing information into each of the nodes called the LSTM. A brief overview of LSTM gates and cells is described in Equations (3) to (6) are described below,

$Input\ Gate\ In_t = \sigma(W_{in.}[hs_t - 1], x_t + b_{in})$ (3)

$Memory\ Cell\ C_t = tan\,h(W_{c.}[hs_t - 1], x_t + b_e)$ (4)

$Forget\ Gate\ f_t = \sigma(W_{f.}[hs_t - 1], x_t + b_f)$ (5)

$Output\ Gate\ f_o = \sigma(W_{o.}[hs_t - 1], x_t + b_o)$ (6)

In the above equations b is used for bias vector, w represents weight and x_t, represents input vector at time *t*, whereas in, *f* and *o* represent input, forget and output gates and c_t represents cell memory.

So based on that network our proposed framework uses 3 layers of LSTM model i.e. embedding layer, LSTM layer and dense layer. The first layer i.e. embedding layer utilised to convert the input words-which are represented as integers into dense vectors with predetermined sizes. The second layer is the LSTM layer which can recognise long-term dependencies consists of 32 LSTM units for capturing a distinct element of the input sequences throughout time intervals. The third layer is a dense (fully connected) layer that produces an output value using binary classification problems. For applying this problem, a probability value between 0 and 1 is produced by the sigmoid activation function. The summary of the proposed model is contained in **Figure 6**.

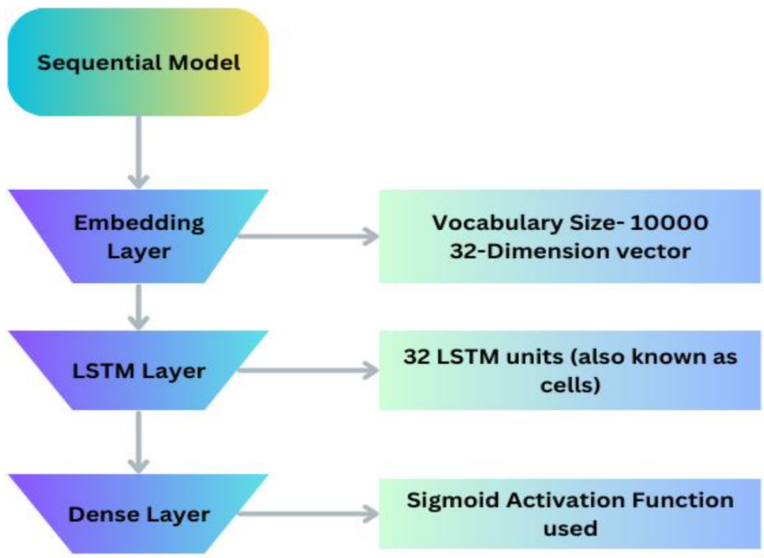

Figure 6. LSTM for proposed framework.

3.4 Collaborative Product Recommendation System

Recommendation Systems apply Information Retrieval (IR) techniques to select some information relevant to a given user by taking into account their preferences, relevant data and reviews sentiment. Filtering system is utilized to improve the recommendations in e-commerce platforms. Currently, CF is the most popular method for creating recommendation systems, predicting user behaviour from user-item evaluations. CF (Papadakis et al., 2022) is mainly of two types i.e. Memory based and model based. In the former category, each time a new recommendation is required, specific computations are made using a database of past users' choices. In the latter case, the database is kept once again, but initially it is employed to create a description model, which is subsequently utilised to provide suggestions for an active user. In our model, a model-based CF system is used by performing a Singular Value Decomposition (SVD) (Nilashi et al., 2018) on a sparse matrix. In this research, amazon fashion dataset (Sultan, 2022) is used for sentiment analysis of reviews. In the proposed model, SA is combined with a collaborative filtering (CF) system for providing the top n product recommendation system. A CF-RS is used to develop user based and item-based RS. In user-based CF similar users are identified based on their past behaviour with items which are reviewed or purchased and in item-based CF similar items are recommended to the user based on their purchasing items or the items which are referred to other users if they liked a specific item. A widely used approach that creates individualised recommendations by examining the choices and behaviours of a user base is called collaborative filtering. It is based on such an assumption that those who have similar interest and preferences in the past would also act similarly in the future. The CF technique utilises past user-item interactions to identify patterns and similarities between individuals and items. The characteristics or attributes associated with items or users do not need to be explicitly known in order to apply the CF technique. It simply makes use of the visible interactions that occur between users and items. It is especially useful in scenarios when item attributes are not always easily accessible or when consumers' preferences vary over time.

3.5 Proposed SAEF Model based on LSTM

In the Sentiment Analysis with Enhanced Features (SAEF) algorithm, user reviews were first loaded from the Amazon dataset. Tokenization of the textual data and sequence padding to standardise input lengths were part of the stage of preprocessing.

Let *R* be a set of reviews, which may be shown as:
$R = \{r_1, r_2, \dots \dots \dots . r_m\}$

where, *m* is the total number of the review.

The tokenizer is set with a parameter that constrains the size of the vocabulary that it will allow. The tokenizer learns the word index from the reviews:
$T = Tokenization(num\_words = 1000)$
$W = T.fit\_on\_texts(R)$
$W: word \rightarrow index$

where, *W* is the matrix of word to index conversion.

The reviews are converted into sequences of indices based on the learned word index, and each sequence s_i for review r_i is defined as:
$S = T.texts\_to\_sequences(R)$
$S = \{s_1, s_2, \dots \dots \dots \dots . s_m\}$
$s_i = [w_1, w_2, \dots \dots \dots w_k]$

where, *S* is actually a list of sequences, k refers to the number of tokens in review r_i, and where each w_j corresponds to the mapping *W* for the *j*-th word in the review.

Each of these sequences was then taken to an embedding layer of LSTM to convert the sequences into 32 dimensions to give explicit meaning to the types of words used. The LSTM layer was employed to model temporal dependencies within the text, allowing the model to capture contextual information and the overall meaning of the sequence.
$X \in R^{n \times t}$.

where, *X* represents the input data *n* being a number of reviews and *t* being maximum sequence length. The embedding function f_{embed} maps each token $x_{i,j}$ (the *j*-th word of the *i*-th review) to a 32-dimensional vector.
$f_{embed} \coloneqq x_{ij} \rightarrow v_{ij} \in R^{32}$.

after the embedding, the input sequence *X* is transformed into:
$V = \{v_1, v_2, \dots \dots \dots . . v_n\}$ where, $V \in R^{n \times t \times 32}$.

LSTM layer processes the embedded input V sequentially, learning temporal dependencies between the tokens in the sequence. LSTM operation can be denoted as
$h_t = f_{LSTM}(v_t, h_{t-1})$.

where, $h_t \in R^{32}$ is the hidden state at the timestamp t, v_t is the input at timestamp t and f_{LSTM} is the LSTM recurrence function.

The dense layer then transforms the final LSTM output into a single scalar using a linear transformation followed by a sigmoid activation function,
$y = \sigma W_d h_t + b_d$.

where, $W_d \in R^{l\times 32}$is the weight matrix, $b_d \in R$ $is\ the\ bias\ term, \sigma(x) = 1/1 + e^{-x}$ is the sigmoid activation function. The output y is the probability of the review belonging to a certain sentiment class (e.g., positive sentiment if $y > 0.5$).

It generates a probability score between 0 and 1. A review was classified as expressing positive sentiment if the probability score exceeded 0.6, negative sentiment if the score was below 0.4, and neutral sentiment if the score fell between 0.4 and 0.6. This structure enabled the model test to give an accurate polarity to sentiment which in turn allowed it to being used by the product recommendation system.

Following the sentiment prediction for user reviews, the sentiments were categorized into three classes: positive, negative, or neutral. These categorized sentiments were then summed in per product ASIN basis, as demonstrated in the **Figure 7** sentiment distribution graph. For building the Re command line Script for the collaborative filtering recommendation system, the Singular Value Decomposition (SVD) algorithm of the Surprise Python library was used.

Then a user-item interaction matrix R is created where rows include users and columns include number of products/items. The matrix R comprises $R_{u,i}$ where $R_{u,i} \in [1,5]$ which denotes the rating given by user u to item i.

It starts by creating a data set D with the attributes being the users' id, items' id and the ratings these users gave to the items. Opening the Surprise library, a reader object is being initialized with rating from 1 to 5. On the basis of an 80:20 stratified random sampling technique, the dataset D is divided into a training and a testing set.
$D_{train} \subset R$ and $D_{text} \subset R$.

Once the data is prepared, the SVD algorithm factorizes the user-item interaction matrix R into three components:
$R \approx PQ^T$

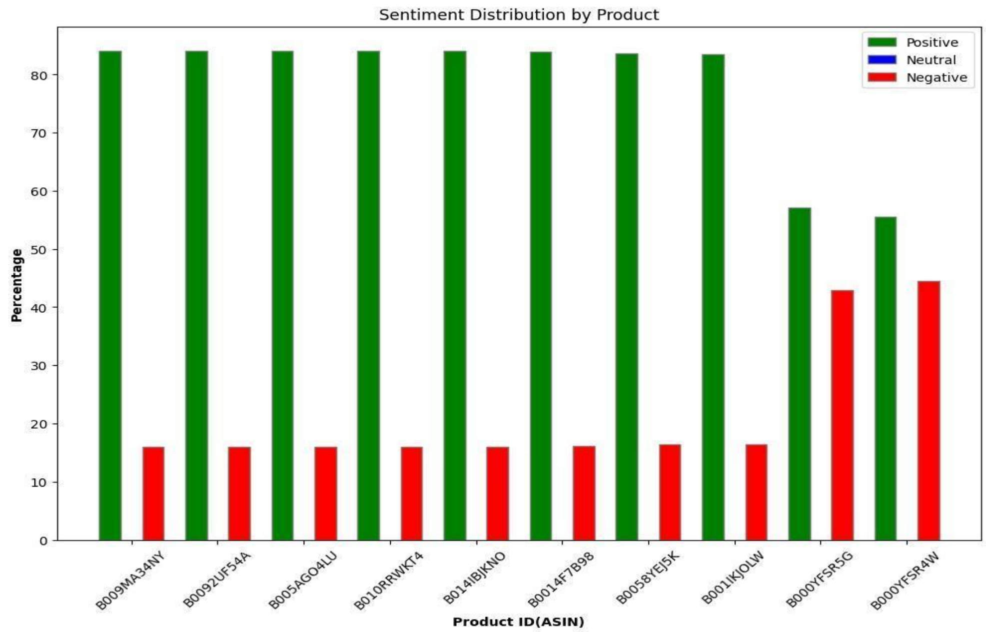

Figure 7. Sentiment distribution graph for SAEF.

where, $P \in R^{m\times k}$ is the user latent matrix, capturing user-specific features in *k*-dimensional space, $Q \in R^{m\times k}$ is the item latent matrix, capturing item-specific features in the same *k*-dimensional space and *k* represents the number of latent factors that define the underlying dimensions of the user-item interaction. For any user u and item *i*, the predicted rating $r^{u,i}$, *i* is computed as follows:

$$r^{u,i} = \mu + b_u + b_i + P_u^T Q_i.$$

where, μ is the overall mean rating of all users' and items' while b_u is specifically a user bias *u* and b_i is an item bias *i*; P_u is latent feature vector of the user *u* and Q_i is the latent feature vector for the corresponding item *i*.

The trained SVD model is then applied to predict missing user-item ratings. After fitting the SVD model on the training set D_{train}, predictions for unseen user-item pairs are obtained. The estimated rating $r^{u,i}$ for a given user-item pair is computed by combining the learned biases and latent feature vectors.

Once the collaborative filtering model is established, the recommendation process is refined by incorporating sentiment scores derived from the LSTM-based sentiment analysis models. The final score for each product i_j in relation to a specific product i_{target} is calculated by adding the predicted rating r_{i_{target},i_j} from the SVD model with the average sentiment score $S(i_j)$ for that item:

$$Estimate\left(i_{target}, i_j\right) = r_{i_{target},i_j} + S(i_j).$$

These combined scores are then used by the system to sort the items and therefore offer the first *n* number of product recommendations for the user. The top 10 items $\{i_1, i_2, \dots \dots \dots i_{10}\}$ are selected by sorting the estimated scores in descending order:

$$\{i_1, i_2, \dots \dots \dots i_{10}\} = Top - 10\left(Estimate\left(i_{target,} i_j\right)\right).$$

Finally, the accuracy and RMSE are presented to conclude the model performance. This is a hybrid approach that integrates SA and weighted CF whereby user sentiment preferences as well as user-item interaction matrices are jointly used to enhance the quality of recommendations.

3.6 Proposed SAFF Model based on LSTM

In the case of the Sentiment Analysis with Filtered Features (SAFF), to employ the process, first, there is a set of product-related features referred to as F. Such aspects include quality, fit, comfort, style, design, durability, material, size, color, price, shipment, packing, customer service, support, breathability and friction, sole, lining, and stitching. The set *F* serves as a predefined list of features to be analyzed across product reviews.

Given a set of reviews $R = \{r_1, r_2 \dots\dots.. r_N\}$, where each $r_i \in R^n$ represents the textual content of the *i*-th review, and a corresponding set of numerical ratings $S = \{s_1, s_2 \dots\dots.. s_N\}$, where each $s_i \in [1,5]$ indicates the rating assigned by the user for the respective review, the first task is to filter out invalid data. Reviews and ratings containing NaN values are removed to ensure the quality of the data, yielding a filtered set of reviews denoted by $R_{filtered}$:

$R_{filtered} = \{r_i \in R | r_i = NaN \text{ } and \text{ } s_i = NaN\}$.

For each valid review in $R_{filtered}$, we generate corresponding sentiment labels based on the rating. A binary sentiment label ℓ_i is assigned for each review r_i as follows:

$$l_i = \begin{cases} 1, & if \ s_i \geq 4 \\ 0, & if \ s_i < 4 \end{cases}$$

Thus, the resulting set of sentiment labels, $L = \{l_i, l_2 \dots\dots. l_N\}$, distinguishes between positive and negative reviews, with a threshold of 4. Next, we process the textual data. Each review in $R_{filtered}$ is tokenized using a vocabulary of size *V*=10,000. The tokenizer maps each review r_i into a sequence of tokens $t_i = \{t_1, t_2 \dots\dots\dots t_n\}$, where each token $t_i \in [1, V]$ corresponds to a word from the vocabulary:

$T = \{t_1, t_2 \dots\dots\dots t_N\}$.

where, t_i represents the tokenized sequence of review r_i.

Because the length of each of the tokenized sequences may be arbitrary, we apply a padding operation to all of the sequences. Each token sequence t_i is padded to a fixed length *L*=100, producing a padded sequence p_i, where:

$p_i = Pad(t_i, L)$.

Thus, the complete set of padded sequences is denoted as $P = \{p_1, p_2 \dots\dots\dots p_N\}$, where, each $p_i \in R^L$ ensures consistency in the input length for subsequent stages of the model.

It is important to note that the threshold for sentiment labeling could be refined to include a neutral class, resulting in a more granular sentiment analysis. For instance, instead of using binary labels, we could classify reviews into three categories: positive, neutral, and negative, as follows:

$$l_i = \begin{cases} 1, & if \ s_i \geq 4 (positive), \\ 0, & if \ s_i = 3 (neutral), \\ -1, & if \ s_i < 3 (negative). \end{cases}$$

This modification makes possible the classification of sentiment into more categories which may be of much use depending on what the study wants to achieve. Last but not least, choosing $L = 100$ is based on the empirical choice but this parameter is just as well could be tuned further. Additionally, leveraging pre-trained embeddings (such as GloVe or Word2Vec) instead of randomly initialized embeddings might significantly improve the model's capacity to capture semantic relationships within the reviews.

Next, we adjust the ratings with the sentiment score, s_i represent the adjusted rating for a given review r_i, based on its sentiment score $\sigma(r_i)$. The adjustment to the rating s_i is defined by the following rule:

$$s_i = \begin{cases} min(s_i + 0.5, 5.0), & if\ \sigma(r_i) > 0.6, \\ max(s_i - 0.5, 1.0), & if\ \sigma(r_i) < 0.4, \\ s_i, & otherwise. \end{cases}$$

Thus, if the sentiment score $\sigma(r_i)$ is greater than 0.6, indicating positive sentiment, the original rating s_i is boosted by 0.5, up to a maximum of 5. If the sentiment score is less than 0.4, indicating negative sentiment, the rating is reduced by 0.5, with a lower bound of 1. Ratings within the neutral sentiment range (0.4 to 0.6) remain unchanged.

The next step involves classifying the sentiment of specific product features mentioned within each review. Given a set of features F for each review $r_i \in R$,we classify the sentiment of each feature $f_j \in F$ as positive, neutral, or negative based on the sentiment score $\sigma(r_i)$. If a feature f_j is present in the review text r_i, the sentiment score for the entire review *σ(ri)* is used to classify the feature's sentiment:

$$Sentiment(f_j, r_i) = \begin{cases} Positive, & if\ \sigma(r_i) > 0.6, \\ Negative, & if\ \sigma(r_i) < 0.4, \\ Neutral, & otherwise. \end{cases}$$

For each review, if a feature f_j is not mentioned, it is labeled as 'Not mentioned'.

To aggregate sentiment scores for each product across multiple reviews, let $I = \{i_1, i_2, \dots \dots \dots \dots \dots i_k)$, denote the set of unique product IDs. For each product i_k, and for each feature $f_j \in F$, the sentiment counts for positive, negative, neutral, and not mentioned are accumulated across all reviews:

$$A(i_k, f_i) = \{Positive{:}\, p_{k,j}, Negative{:}\, n_{k,j}, Neutral{:}\, u_{k,j}, Not\ mentioned{:}\, m_{k,j}\}$$

where, $p_{k,j}$, $n_{k,j}$, $u_{k,j}$ and $m_{k,j}$ represent the counts of positive, negative, neutral, and not mentioned sentiments for feature f_j across all reviews of product i_k. This aggregation helps in understanding the sentiment distribution for each product feature across different reviews.

Finally, the overall sentiment for a product feature is summarized as a distribution of positive, negative, neutral, and not mentioned sentiments, providing insight into which aspects of the product are praised or criticized by customers. This aggregated data can further be used to generate recommendation insights. **Figure 8** demonstrates sentiment distribution graph for SAFF.

To incorporate the sentiment-based rating adjustments into the recommendation system, we construct a user-item interaction matrix M where each entry represents a user's rating for a specific product. This matrix will reflect the adjusted ratings based on the sentiment analysis of the corresponding product reviews.

Given the user IDs $U = \{u_{1,} u_2 \dots \dots \dots . . u_m\}$ and item IDs $I = \{i_1, i_2 \dots \dots \dots . i_n\}$, we first define the original rating matrix:

$R = [r_{ij}]_{m\times n}.$

where, r_{ij} represents the initial rating provided by user u_i for item i_j.

For each item i_j, if a review r_i for that item contains any of the predefined product features F, we adjust the rating r_{ij} using the predicted sentiment score $\sigma(r_i)$ from the review text. This adjustment follows the previously defined rules:

$$\hat{r}_i = \begin{cases} min(r_{ij} + 0.5, 5.0), & if\ \sigma(r_i) > 0.6, \\ max(r_{ij}\ -\ 0.5, 1.0), & if\ \sigma(r_i) < 0.4, \\ r_{ij}, & otherwise. \end{cases}$$

This produces an adjusted rating matrix $R = [r_{ij}]_{m\times n,}$, where the rating reflects the sentiment analysis for the review of each product.

Next, we aggregate the ratings for each user-item pair by taking the mean of the adjusted ratings when multiple reviews are available. Let $R_{ui,ij}$ represent the set of all reviews from user u_i for item i_j. The final rating r~ij for that pair is given by:

$$r_{ij} = \frac{1}{|\mathcal{R}_{u_i,i_j}|} \sum_{r_k \in \mathcal{R}_{u_i,i_j}} r_k,$$

where, r_k is the sentiment-adjusted rating for each individual review r_k.

After aggregating the sentiment-adjusted ratings, the user-item matrix M is constructed as follows:

M=$[\tilde{r}_{ij}]_{m*n}$,

where, each entry represents the average sentiment-adjusted rating for user u_i and item i_j. Any missing entries (i.e., cases where a user has not rated an item) are filled with zeros.

To compute item-based recommendations, we calculate the similarity between items using cosine similarity. The cosine similarity $S(i_k,i_j)$ between two items i_k and i_j is defined as:

$$\mathrm{S}(i_k, i_j) = \frac{M_{:,k}\ .\ M_{:,j}}{\|M_{:,k}\|\|M_{:,j}\|},$$

where, $M_{:,k}$ and $M_{:,j}$ are the columns of the matrix M corresponding to items i_k and i_j, representing the user's ratings for those items. This results in an item-item similarity matrix S, which can be used to recommend similar items for each product in the catalog. Here top 10 similar items are recommended for each product. Finally, the model performance is assessed using root mean squared error (RMSE) and accuracy metrics.

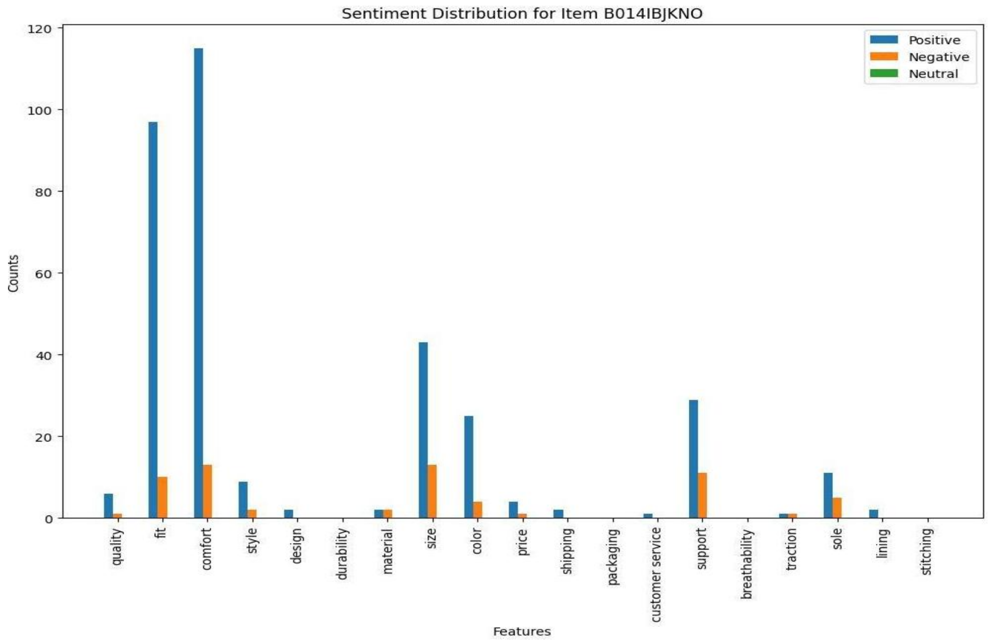

Figure 8. Sentiments distribution graph for SAFF.

4. Evaluation Metrics Used

This performance of our model is assessed using different parameters in other words accuracy of the model is evaluated by finding precision, accuracy, recall, specificity, Confusion matrix, f1-score, AUC (area under the curve), ROC (Receiver Operator Characteristic). The table that is used for description of the performance of classification model on a set of the test data with known true values is called a confusion matrix shown in **Figure 9**.

- **True Positives (TP)**: Scenarios where the prediction made by the model was accurate with regard to the target class that was labelled positive.
- **True Negatives (TN):** That is, only those situations that the model produces an accurate and correct output when the actual and predicted value are negative.
- **False Positives (FP)**: Sometimes the model's output might go wrong and classify when it is actually negative, referred to as Type I error.
- **False Negatives (FN)**: Situation where the model classified as negative when the correct classification should have been positive (called a "Type II error").

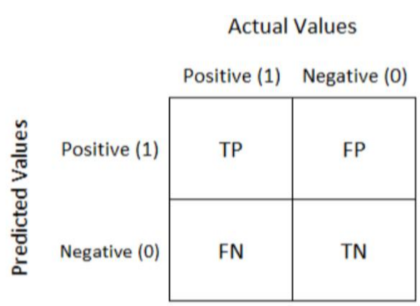

Figure 9. Confusion matrix showing classification outcomes.

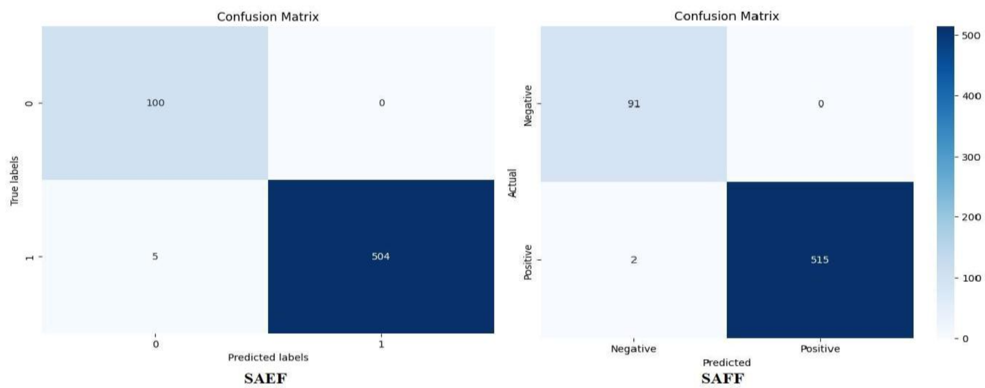

Figure 10. Confusion matrix for proposed models.

Figure 10 shows the confusion matrix for the proposed framework.

The model is trained with approximately 600 reviews from a set of 3045 reviews showing an accuracy of 99.18 % for SAEF and 99.67 % for SAFF. The accuracy of the model is calculated by applying different performance metrics shown in **Table 3**.

Table 3. Performance metrics.

Performance metrics	Formulas
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1-Score	$\frac{2*Precision*Recall}{Precision+Recall}$
Specificity	$\frac{TN}{TN+FP}$
RMSE	$sqrt[(\sum(P_i - O_i)^2)/n]$

Figure 11 demonstrates the results of various performance metrics through the graphical statistics.

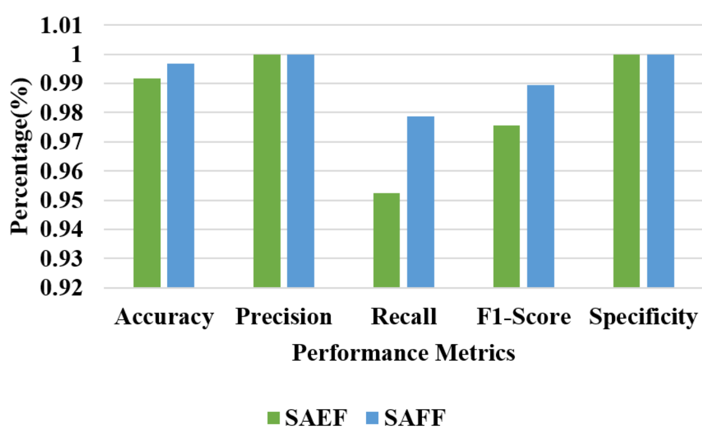

Figure 11. Performance metrics for proposed framework.

As shown in the **Table 4**, the proposed model outperformed the traditional model in terms of performance metrics.

Table 4. Comparison with existing model.

Performance metrics	SAEF	SAFF	Existing model
RMSE	**0.37**	**0.27**	**1.99**
Accuracy	**98.85 %**	**99.34 %**	**98.43 %**
Precision	**100 %**	**100 %**	**99.27 %**
Recall	**97.46 %**	**98.79 %**	**98.12 %**
F1-Score	**96.59 %**	**97.8 %**	**98.69 %**
Specificity	**100 %**	**100 %**	**Missing**

5. Results

After the dataset was ready, the recommended model was run. The obtained dataset may be divided into two independent sets: a test set with 20% of the data and a training set with 80% of the data. The model was developed, trained, and assessed on Google Collaboratory utilising LSTM. TensorFlow and Python were utilised nonstop during the entire procedure. For prediction, the Adam optimisation technique was applied. The hyperparameter utilized in recommendation model was shown in **Table 5**.

Table 5. Hyperparameter.

Hyperparameter	
Number of epochs	**40**
Batch size	**64**
Optimizer	**Adam**
Activation function	**Sigmoid**
Loss function	**BinaryCrossentropy**
Maximum sequence length	**100**

The accuracy and loss plots demonstrate the functioning and convergence of the model throughout training. For both training and validation data, the accuracy plot offers a visual depiction of how the model's accuracy varies during training. As the model gains knowledge from the training instances, the accuracy of the training data often increases with time. The Accuracy graph shown in **Figure 12**.

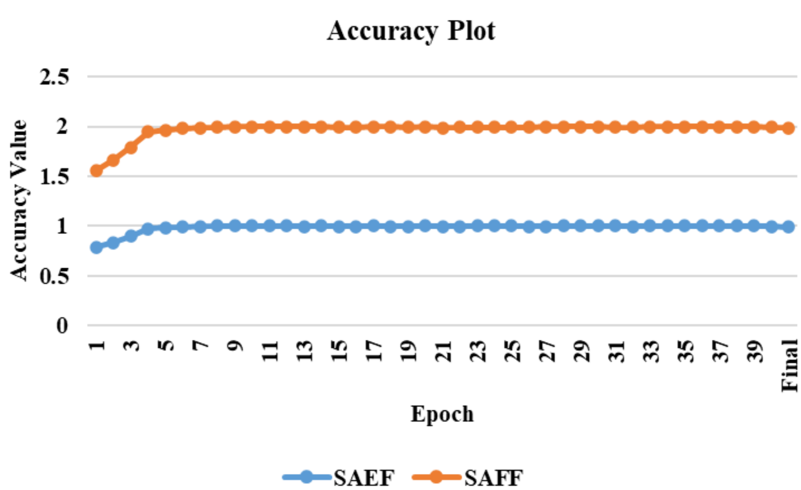

Figure 12. Accuracy plot for proposed models.

This loss is shown in the loss plot in the form of trend of the model's loss for all the training epochs. The degree of agreement between the model's predictions and the actual values is assessed by the loss function. The purpose of training is to ascertain the extent to which error is minimized in order to show where the model is predicting the correct values. This means the model is learning how to fit the training set of data and from the figure it evident that the loss is decreasing across the epochs. The loss graph shown in **Figure 13**.

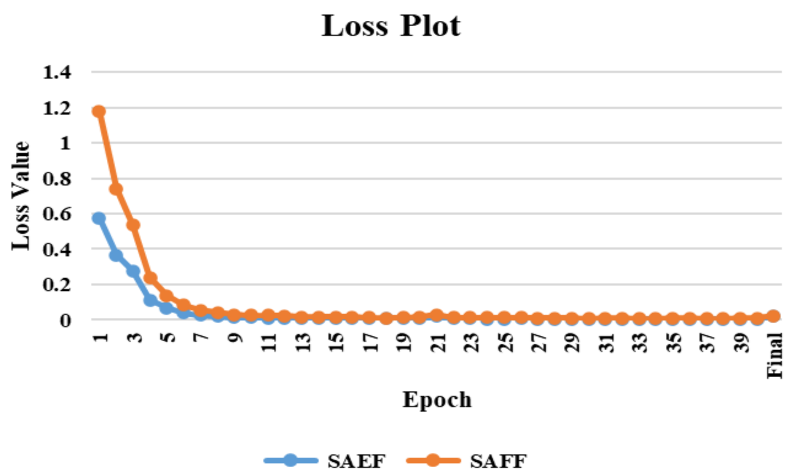

Figure 13. Loss plot for proposed models.

The Sentiment Analysis with Enhanced Features (SAEF) and Sentiment Analysis with Filtered Features (SAFF) algorithms both utilize collaborative filtering to enhance product recommendations, but they take distinct approaches in how they incorporate sentiment analysis. SAEF emphasizes a broader contextual understanding of sentiment, while SAFF focuses on feature-specific sentiment analysis.

SAEF demonstrated strong overall performance with an accuracy of 99.18%, making it highly effective in classifying the overall sentiment within customer reviews. Its precision of 100% shows that it never misclassifies a sentiment once identified, ensuring that every sentiment classification is correct. This contributes to improving recommendations by capturing the nuanced emotional tone in user feedback, such as distinguishing between slightly positive and strongly positive sentiments. However, SAEF's recall of 95.23% indicates that it sometimes misses more subtle sentiments, which may cause it to overlook some user preferences. With its F1-Score reaching a remarkable 97.56%, the strengths concerning the model are optimized and maintained. The specificity of 100% shows that SAEF does not make any error in classifying a neutral sentiment as a false positive and seizes true negative sentiments fully. The technique of integrating sentiment into collaborative filtering assists in making better quality recommendations, where the customers' reviews hold more about their all-around perceptions about the product rather than its attributes. Its RMSE of 0.37 shows how the adjustment of the ratings given by a user to a point where the user's readiness to use the product is increased, improving how the recommendations suit the user's needs.

In contrast, SAFF algorithm, for instance, can target the comfort or durability sentiment associated with the particular product feature and achieve an even greater accuracy of 99.67%. The reason for this increase is decidedly based on feature-oriented sentiment classification - they present a better picture of user preferences. And as was the case with SAEF, the SAFF system does not make any false positive assertions, and the reported precision score is also at perfection - 100%. Its recall is reported at 97.87%, which is greater than that of SAEF and implies greater hits for true positive sentiments concerning specific details for products that customers are most concerned with. This is important when it comes to tweaking feature specific preferences because it provides measures of specific features such as the "fit" or "style" the users appreciate. Here we presented the F1-score of 98.93% which indicates about SAFFs capability of retaining high consistency for the sentiment classification. The specificity of SAFF is also 100%, which means that the algorithm does not include non-relevant sentiment at all. Additionally, its reduced RMSE of 0.27 compared to that of SAEF indicates that the sentiment-aware carried out in the proposed method is closer to the actual user expectation in terms of rating adjustment for certain product features.

Both algorithms incorporate collaborative filtering to produce improved results; nevertheless, the methods give rise to diverse advantages. While SAEF is especially strong at capturing a wide range of general user sentiment, SAFF captures detailed feature-level biases which result in uniquely personal recommendations. Thus, in terms of the SA parameters accuracy, recall and RMSE, SAFF is higher than SAEF, making it useful for fine-tuning recommendations depending on certain features of the products. On the other hand, due to its focus on overall sentiment context, capability of boosting recommendation becomes highly effective for scenarios where user feedback extends to a number of attributes of the product.

These two algorithms when used together within a CF style framework provide a strong foundation for building a RS. By complementing the general positive/negative sweeping features by SAEF, SAFF can capture precise feature-level positive/negative trends, so that the recommendations based on satisfaction and preference for both overall product and particular features can be properly driven.

By integrating the item sentiments with CF, top 10 items are recommended for both the models where SAEF model recommends (B00G8Q7JZ4, B00RLSCLJM, B00ND9047Y, B016XAJLVO, B01595OS62, B00LKWYX2I, B01H7KY678, B00MLYE8PQ, B00GKF5BAS, B005AGO4LU) and SAFF model recommends (B001IKJOLW, B00I0VHS10, B0092UF54A, B0058YEJ5K, B00GKF5BAS, B00ND9047Y, B001LNSY2Q, B01595OS62, B000YFSR4W, B0014HA6VG) for same product with B0014F8TIU item id. After analysing the recommended products of both the models our framework has given recommendations are (B00ND9047Y, B01595OS62, B00GKF5BAS).

6. Conclusion

This research proposes an enhanced product recommendation system by integrating Sentiment Analysis (SA) with Collaborative Filtering (CF), using Long Short-Term Memory (LSTM) networks. The system not only considers user actions but also interprets the emotional tone of reviews, resulting in more personalized and accurate recommendations. The two models developed, Sentiment Analysis with Enhanced Features (SAEF) and Sentiment Analysis with Filtered Features (SAFF), demonstrated strong performance in sentiment classification, contributing to improved recommendation accuracy.

The integration of SA with CF led to significant improvements in metrics such as Root Mean Squared Error (RMSE) and accuracy, outperforming traditional models shown in **Table 4**. SAEF excelled at capturing overall sentiment, while SAFF focused on feature-specific sentiment analysis, enhancing the personalization of product suggestions.

Future work could include real-time data integration and scaling the system for larger datasets using GPUs. This approach shows great potential in refining product recommendation systems by aligning suggestions more closely with user preferences and sentiments, thereby improving user satisfaction and engagement in e-commerce platforms.

Conflict of Interest
The authors declare that they have no conflict of interest.

Acknowledgments
This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors would like to thank the editor and anonymous reviewers for their comments that help improve the quality of this work.

AI Disclosure
The author(s) declare that no assistance is taken from generative AI to write this article.

References

Abdillah, J., Ibnu, A., & Yanuar, F.A.W. (2020). Emotion classification of song lyrics using bidirectional LSTM method with GloVe word representation weighting. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *4*(4), 723-729. https://doi.org/10.29207/resti.v4i4.2156.

Ajmeera, N., & Kamakshi, P. (2024). Hamiltonian deep neural network fostered sentiment analysis approach on product reviews. *Signal, Image and Video Processing, 18*(4)*,* 3483-3494. https://doi.org/10.1007/s11760-024-03014-6.

Atlas, L.G., Arockiam, D., Muthusamy, A., Balusamy, B., Selvarajan, S., Al-Shehari, T., & Alsadhan, N.A. (2025). A modernized approach to sentiment analysis of product reviews using BiGRU and RNN based LSTM deep learning models. *Scientific Reports*, *15*(1), 16642. https://doi.org/10.1038/s41598-025-01104-0.

Awati, C.J., Shirgave, S.K., & Thorat, S.A. (2025). An efficient top-n-items recommendation system using a hybrid optimal deep learning model. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-025-20858-3.

Dang, C.N., Moreno-García, M.N., & De la Prieta, F. (2021). An approach to integrating sentiment analysis into recommender systems. *Sensors*, *21*(16), 5666. https://doi.org/10.3390/s21165666.

Dang, N.C., Moreno-García, M.N., & De la Prieta, F. (2020). Sentiment analysis based on deep learning: a comparative study. *Electronics*, *9*(3), 483. https://doi.org/10.3390/electronics9030483.

Dhola, K., & Saradva, M. (2021). A comparative evaluation of traditional machine learning and deep learning classification techniques for sentiment analysis. *Proceedings of the Confluence 2021: 11th International Conference on Cloud Computing, Data Science and Engineering* (pp. 932-936). IEEE. Noida, India. https://doi.org/10.1109/confluence51648.2021.9377070.

Elahi, M., Kholgh, D.K., Kiarostami, M.S., Oussalah, M., & Saghari, S. (2023). Hybrid recommendation by incorporating the sentiment of product reviews. *Information Sciences*, *625*, 738-756. https://doi.org/10.1016/j.ins.2023.01.051.

Gao, Y., Zheng, H., & Cui, H. (2025). User preference modeling for movie recommendations based on deep learning. *Scientific Reports*, *15*(1), 16592. https://doi.org/10.1038/s41598-025-00030-5.

Gheewala, S., Xu, S., & Yeom, S. (2025). Deep attention dynamic representation learning networks for recommender system review modeling. *Big Data Research*, *40,* 100521. https://doi.org/10.1016/j.bdr.2025.100521.

Khapre, S., Singh, P., Shankar, A., Nayak, S.R., & Diwakar, M. (2023). Context-based intelligent recommendation by code reuse for smart decision support and cognitive adaptive systems. *International Journal of Intelligent Unmanned Systems*, *11*(1), 75-87. https://doi.org/10.1108/ijius-07-2021-0055.

Krishna, E.S.P., Ramu, T.B., Chaitanya, R.K., Ram, M.S., Balayesu, N., Gandikota, H.P., & Jagadesh, B.N. (2025). Enhancing E-commerce recommendations with sentiment analysis using MLA-EDTCNet and collaborative filtering. *Scientific Reports*, *15*(1), 6739. https://doi.org/10.1038/s41598-025-91275-7.

Kuang, M., Safa, R., Edalatpanah, S.A., & Keyser, R.S. (2023). A hybrid deep learning approach for sentiment analysis in product reviews. *Facta Universitatis, Series: Mechanical Engineering*, *21*(3), 479-500. https://doi.org/10.22190/fume230901038k.

Kumar, R.P., Mohan, G.B., Elakkiya, R., Kumar, M.C., & Rithani, M. (2023). Automated sentiment classification of amazon product reviews using LSTM and bidirectional LSTM. In *2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques* (pp. 1-6). IEEE. https://doi.org/10.1109/easct59475.2023.10393514.

Liu, B., Zeng, Q., Lu, L., Li, Y., & You, F. (2021). A survey of recommendation systems based on deep learning. *Journal of Physics: Conference Series*, *1754*(1), 012148. https://doi.org/10.1088/1742-6596/1754/1/012148.

Liu, Y., Lu, J., Yang, J., & Mao, F. (2020). Sentiment analysis for e-commerce product reviews by deep learning model of Bert-BiGRU-Softmax. *Mathematical Biosciences and Engineering*, *17*(6), 7819-7837. https://doi.org/10.3934/mbe.2020398.

Mahadevaswamy, U.B., & Swathi, P. (2022). Sentiment analysis using bidirectional LSTM network. *Procedia Computer Science*, *218*, 45-56. https://doi.org/10.1016/j.procs.2022.12.400.

Mu, R. (2018). A survey of recommender systems based on deep learning. *IEEE Access*, *6*, 69009-69022. https://doi.org/10.1109/access.2018.2880197.

Nilashi, M., Ibrahim, O., & Bagherifard, K. (2018). A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Systems with Applications*, *92*, 507-520. https://doi.org/10.1016/j.eswa.2017.09.058.

Ongsulee, P. (2017). Artificial intelligence, machine learning and deep learning. In *2017 15th International Conference on ICT and Knowledge Engineering* (pp. 1-6). IEEE. Bangkok, Thailand. https://doi.org/10.1109/ictke.2017.8259629.

Osman, N.A., Noah, S.A.M., Darwich, M., & Mohd, M. (2021). Integrating contextual sentiment analysis in collaborative recommender systems. *PLoS ONE*, *16*(3), e0248695. https://doi.org/10.1371/journal.pone.0248695.

Pandey, N.K., Singh, N., Kumar, K., Baruah, S.S., Diwakar, M., & Mishra, A.K. (2023). Online shows recommendations using machine learning in big data environment. In *2023 International Conference on Computer, Electronics and Electrical Engineering and their Applications* (pp. 1-6). IEEE. *Srinagar, Garhwal.* https://doi.org/10.1109/ic2e357697.2023.10262514.

Papadakis, H., Papagrigoriou, A., Panagiotakis, C., Kosmas, E., & Fragopoulou, P. (2022). Collaborative filtering recommender systems taxonomy. *Knowledge and Information Systems*, *64*(1), 35-74. https://doi.org/10.1007/s10115-021-01628-7.

Rehman, A.U., Malik, A.K., Basit, R., & Waqar, A. (2019). A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis. *Multimedia Tools and Applications*, *78*(18), 26597-26613. https://doi.org/10.1007/s11042-019-07788-7.

Seilsepour, A., Ravanmehr, R., & Nassiri, R. (2023). Topic sentiment analysis based on deep neural network using document embedding technique. *Journal of Supercomputing*, *79*(17), 19809-19847. https://doi.org/10.1007/s11227-023-05423-9.

Sharma, S., Diwakar, M., Joshi, K., Singh, P., Akram, S.V., & Gehlot, A. (2022). A critical review on sentiment analysis techniques. In *Proceedings of 3rd International Conference on Intelligent Engineering and Management* (741-746). IEEE. London, United Kingdom. https://doi.org/10.1109/iciem54221.2022.9853140.

Sultan, N. (2022). Sentiment analysis of amazon product reviews using supervised machine learning techniques. *Knowledge Engineering and Data Science*, *5*(1), 101-108. https://doi.org/10.17977/um018v5i12022p101-108.

Taher, G. (2021). E-commerce: advantages and limitations. *International Journal of Academic Research in Accounting Finance and Management Sciences*, *11*(1), 153-165. http://dx.doi.org/10.6007/ijarafms/v11-i1/8987.

Thomas, R., & Jeba, J.R. (2024). A novel framework for an intelligent deep learning based product recommendation system using sentiment analysis (SA). *Automatika*, *65*(2), 410-424. https://doi.org/10.1080/00051144.2023.2295148.