IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 August 2024, accepted 1 September 2024, date of publication 9 September 2024,
date of current version 19 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3456562

RESEARCH ARTICLE

A Novel Customer Review Analysis System Based on Balanced Deep Review and Rating Differences in User Preference

RAND ALMAHMOOD[1], MUHAMMED MUTLU YAPICI[2], AND ADEM TEKEREK[3]
[1]Department of Computer Engineering, Gazi University, 06560 Ankara, Türkiye
[2]Department of Computer Technologies, Elmadag Vocational School, Ankara University, 06780 Ankara, Türkiye
[3]Department of Computer Engineering, Faculty of Technology, Gazi University, 06500 Ankara, Türkiye

Corresponding author: Rand Almahmood (rjawad.almahmood@gazi.edu.tr)

This work was supported by NVIDIA Corporation.

ABSTRACT The rapid growth of mobile applications and online e-commerce websites has made it easy to gather information to create an enormous quantity of training data to aid consumers in making decisions about what to purchase. On online shopping sites, a helpful review analysis of user reviews can significantly increase users' loyalty. People may significantly influence the market value of goods and customer confidence in e-commerce decisions by using ratings, and reviews. One major issue with users' rating prediction models is that they ignore variations across users that fall inside the user's preferences or reviews. In this paper, we develop a new balanced helpful recommendation model with quantifying users' tendencies (BHRQUT)-based on personalized reviews and ratings to predict helpful reviews and improve recommendation accuracy by creating an auxiliary feature that is computed based on actual ratings and predicted ratings. Text sequence processing was acquired by experimental research on the influence of word vector embedding dimension and word frequency of review text, utilizing (NLP). These features were transformed into vectors based on the embedding layer to the balanced (CNN-BiLSTM) model. Experimental evaluations are performed on four review datasets from the 5-score Amazon domain and our model can significantly enhance the accuracy of helpful review text analysis by 97 percent. According to the experimental results when we compared with other deep recommendation approaches concerning multiple metrics and drew from the different experiments, the presented model can enhance the analyzability of user feedback by enhancing decision-making confidence without reducing accuracy.

INDEX TERMS Balanced deep model, natural processing language, recommendation system, tendencies-based collaborative filtering algorithm.

I. INTRODUCTION

The recommendation systems are predictive tools applied across several domains such as the e-commerce domain that aid users in choosing the right products and minimizing the effort required in the purchase process [1]. Reviews are composed by product users in text form [2]. Rating grade products on a spectrum from 1 to 5 (commonly known as stars) are the primary venues for such reviews to serve as a crucial information hub for consumers making purchase decisions [3].

The (fsQCA) model presented different components to view feedback with an abundance of valuable ratings as more credible compared to those with limited ratings [4]. However, a notable portion of digital customer feedback didn't capture positive comments, particularly for businesses or items with many reviews, and the most recent feedback hasn't yet collected positive responses [5]. Amazon.com, a prominent entity in digital commerce introduced a functionality enabling feedback assessment by prompting users with the question, "Was this review helpful?" after each

The associate editor coordinating the review of this manuscript and approving it for publication was Xianzhi Wang.

comment, empowering any reader to indicate its perceived value [6]. As a result, people find it challenging to read helpful evaluations before making a purchase. To determine if reviews aid in the decision-making process for purchases, Amazon has provided a review helpfulness voting module since 2007 [7].

Preferences for unrated items were predicted based on the similarity between consumers' ratings of an item using collaborative filtering and KNN algorithms, but consumers could give an extreme score to a product, those with the lowest and highest star ratings. It became possible to debate whether we should believe the star rating or not, which has almost no explanation. The rating must be supported by customer reviews. This allows other users to decide whether the reviews are fair or not [8].

Customer comments may contain information that does not help consumers make judgments about what to buy, such as advertising or reviews that are fictitious or phony. the analysis of users' comments are processed on a hybrid CNN-BiLSTM model with the impact of their votes without studying the effect of their ratings [9]. Additionally, there are fundamental issues that emerge when quality components are to be considered. Many existing real-world datasets such as the Amazon-5 dataset are imbalanced. This leads to the problem of overfitting associated with inconsistent and unequally sized samples, which impacts the efficiency of the system [10].

Based on the embedded CNN model, the additional features were improved and converted into feature vectors, and a BiLSTM-based classification was carried out for sentiment classification [11]. Other researchers have combined machine learning techniques with sentiment analysis and deep learning techniques to analyze reviews that are inconsistent with the value of ratings by training. a support vector machine (SVM) for sentiment classification by concatenating the review embedding, which is obtained from paragraph vectors, and the product embedding, which is generated from a recurrent neural network (RNN) with the gated recurrent unit (GRU) [12]. Depending on user tendencies, his/her ratings might be unreliable, the users with similar opinions rate items in a different way, some users mostly give positive ratings and rate really bad items negatively while others usually rate negative and give positive ratings to the best items only. To this end, it is valuable to study how well changing user consideration and balanced CNN-BiLSTM model-based prediction helpful reviews can work in improving recommendations.

In this paper, we propose a balanced helpful recommendation model that quantifies users' tendencies (BHRQUT) -based on users' reviews and ratings of of e-commerce products to analyze and predict the helpfulness of users' reviews. This model is constructed with two independent paths to predict helpful and unhelpful reviews, the first path is based on users' reviews and prediction average rating using different users' inclinations with TCF algorithm [13], and the second path is based on users' reviews and actual rating feature in the original dataset. In each path, the data augmentation technique as the over-sampling approach is adapted to balance the imbalanced (helpful, unhelpful) samples, and each path has a convolutional neural network (CNN) and a bidirectional long-short time memory network (BiLSTM) to capture helpful and unhelpful reviews. Four datasets, the Video Games, Musical Instruments, Books-5, and reviews-Clothing-Shoes dataset from Amazon 5-core, are used to train and test BHRQUT.

The contributions of this paper can be summarized as follows:

- We developed an enhanced recommendation system by predicted rating for users which is based on the user tendency using TCF algorithm with the help of the hybrid balanced CNN-BiLSTM model that can discriminate between users' varying opinions based on item preferences and the item's actual rating.
- We evaluated our experiments on four groups of Amazon review datasets. our proposed model is superior to the baseline studies with respect to RMSE, MAE, MSE, Accuracy, Recall, F1_Score, and Precision.
- Our proposed model (BHRQUT) allowed us to avoid the overfitting problem by utilizing a balanced model that significantly increased our accuracy compared to previous studies that did not use this specific strategy in predicting helpful reviews.
- We analyzed the influence of related factors such as the size of the helpfulness measured ratio on star score (3,4) with TCF and without it and the length of the input sentence on the performance of the model.

The rest sections of this study are as follows: Section II delves into the fundamental principles and theorems of the proposed e-commerce-compliant system. Section III offers a comprehensive description of the proposed system, while Section IV, elaborates on the experimental dataset, evaluation metrics, and the resulting outcomes. After that Section V analyzes the experiments of our model. Finally, Section VI concludes by addressing the discourse, its limitations, and prospects for future research.

II. PRELIMINARY LITERATURE REVIEW

This section summarizes a few publications that discuss various approaches to the recommendation system used in e-commerce in the field of review analysis and quantifying data based on users' tendencies from two aspects: review analysis based on deep CNN models and users' Tendencies-based recommendation systems.

A. REVIEWS ANALYSIS BASED ON DEEP CNN MODELS

CNN models have been the focus of numerous studies. applied to e-commerce platform recommendation systems. The three scenarios that are taken into account for feature selection are individual features, features within each category, and all features. Recommender systems are data mining systems that can effectively handle the task of selecting relevant information from large amounts of data supplied by a user based on their interests, preferences, or observed behavior with respect to a certain item [14]. Deep learning appreciates huge promotion and in over the past decade an accomplishment of intense learning in numerous application areas. The scholarly world and industry have been in a race to apply profound figuring out how to incorporate a more extensive scope of uses based on its ability to unveil numerous unpredictable undertakings [15].

Different approaches based on deep learning are put forth for recommendation systems. Researchers first attempt to create a deep learning system to extract latent attributes from incidental data. The researchers (2019) took into account the fact that the recommendation system will not be able to suggest movies to people who have removed their profiles or whose profiles are not present in the system [16]. Ge et al. [17] proposed HARR model for improving review-based recommendation systems by utilizing CNNs, attention mechanisms, and helpfulness score analysis [17].

Zheng et al. [18] shared two CNN networks to learn user behavior and item properties by exploiting review text [18]. A model developed by Krishnamoorthy [19] for predicting how helpful reviews will be is based on original linguistic components that were retrieved from the review text. The results of this study can provide e-commerce merchants with fresh perspectives on how to organize and rank online reviews and aid consumers in selecting better products.

In the study by Chen and colleagues [20], the prevalent methods used to determine the utility of reviews are based on categorized feedback and sometimes fall short. Certain sectors with a lack of thorough reviews and positive feedback do not accurately portray real-world circumstances. To address this gap, a CNN model predicts the relevance of reviews across various sectors. To analyze user/item reviews simultaneously, a model is constructed with two distinct paths, each of which has an attention mechanism-equipped bidirectional long short-term memory (BiLSTM) network and a convolutional neural network (CNN) to collect local aspect features and global aspect features independently [21].

Furthermore, consumers find it simpler to choose purchases for a range of products and services when they utilize a tailored suggestion service like this one. Sixun and Aonghus [22] produced a high-quality text that complies with the requirements of text-aware recommender systems by utilizing CNN-LSTM with RNN decoder-based tailored review-generating models. This increased recommendation accuracy. A method that predicts multi-criteria ratings from reviews and uses them to determine user priorities to generate recommendation candidates. Multi-criteria prediction ratings were obtained from reviews using a deep learning model based on convolutional neural network (CNN) and bidirectional long short-term memory (BiLSTM) [23]. Due to the large size of shopping data CNNs are powerful tools for feature learning and extraction from structured and unstructured data, and reduce dimensions of data, and BiLSTM algorithms offer significant advantages for modeling sequential data, including the ability to capture long-term dependencies, handle variable-length sequences, and effectively process noisy input making them widely used in various deep learning and artificial intelligence tasks.

B. USERS' TENDENCIES BASED RECOMMENDATION SYSTEMS

The majority of recommendation systems currently in use rely on collaborative filtering, which compares users' ratings of items based on how similar their past buying experiences were. This system was unable to suggest products that had recently been released and to offer recommendations to people with distinct or different tastes and the significance of user reviews in influencing purchase decisions as well as the variation in users' interpretations of those reviews. To solve this problem, Zhang et al presented a hybrid approach based on Probabilistic Matrix Factorization (PMF), presuming that users' preferences may be deduced from their attributes. The system uses item-specific latent characteristics to draw in diverse user types and user-specific latent features to record preferences. For users, a stacked denoising autoencoder is utilized to extract these features, and for items, a convolutional neural network (CNN) is used to capture semantic meaning in comments that people would find interesting. Furthermore, most of these studies are challenging to expand for comparable applications in the future [24]. As a result, research [25] verifies the findings using the recently suggested assessment techniques and examines the advantages of supplementing the original RS-predicted ideas with random suggestions. These methods enable the assessment of consumers' approval of the quality of RS services by computing their tendency toward diversity and novelty.

A sentiment analysis technique and feature selection algorithm (LDA) recognized user emotional inclinations toward product characteristics to enhance feature engineering [26]. The purpose of the study [27] is to take into account user preferences for various quality factors, such as diversity, by creating personalized recommendations through the use of Personalized Ranking Adaptation (PRA). However, this approach disregarded items with ratings lower than 5, which is problematic because these items can be crucial for suggesting useful items.

The proposed model in this paper differs from the existing works as it only recommends informative and useful items to improve customer satisfaction when choosing products and reduce the amount of time to research any product using a balanced CNN-BiLSTM model and cross-validation technique based on review and rating values that are predicted from tendencies collaborative filtering algorithm(TCF) that is dependent on difference of the users' evaluation for the products. The existing work mostly processed images and recommended items based on other actual ratings values and

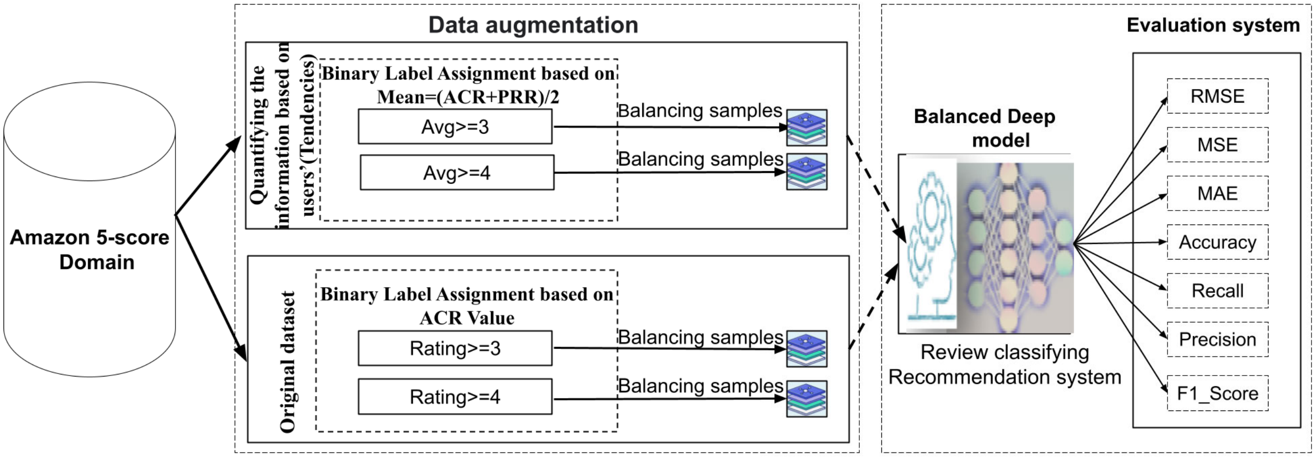


FIGURE 1. Overview of the BHRQUT model.

reviews regardless if the features were useful or not with just an imbalanced sample.

III. PROPOSED SYSTEM METHODOLOGY

In this part, we go into detail about the overview of the proposed BHRQUT model, which is seen in Figure 1.

To enhance the performance of review analysis on item reviews, we combine the advantages of TCF algorithm, CNN, Data Augmentation, and BiLSTM model. Four steps make up our overview. First, we use the TCF algorithm to enhance the review feature. In the second step, we assign a binary label (helpful, unhelpful) to each row based on the values of the average rating that is computed from the 'Actual Rating' and 'Predicted Rating' columns, then balancing the dataset with RoS (Random over sampling) an equal number of rows for each label. In the fourth step review representation, we train a CNN-BiLSTM model on 20 epochs iterations and a 5-fold cross-validation technique, and the model involves evaluating the user review content, which solely consists of rating interactions with reviews that the user has written on some measures as(RMSE, MSE, MAE, Accuracy, Recall, F1-Score, Precision, and confusion matrix). We begin by outlining each phase's specifics.

A. QUANTIFYING THE USER INFORMATION (TENDENCIES)

An analysis of the variations in users' inclinations to select the target element forms the basis of the TCF algorithm [13] concept to enhance the model's input data selection process. A certain amount of information is required to understand the linkages between sets of divergent data, as the algorithm's approach relies on matching user ratings even in cases when ratings are missing. The following main four steps are performed for each user and item tendency and as explained in Figure 2.

The first step is the input original features dataset (user-id, item-id, actual rating). In the second step (process step), we denote a user's tendency (Tu.) and Rui is the rating given by user u to item i, Ri is the mean (average) rating of item i across all users who have rated it, and Iu is the set of items rated by user u.as the average difference between the item mean and his/her evaluations as shown in equation 1.

$$\mathrm{Tu} = \sum_{i \in \mathrm{Iu}} \left(\frac{\mathrm{Rui} - \mathrm{Ri}}{\mathrm{Iu}} \right) \tag{1}$$

the equation 2 Determine the attitude of an item (Ti), i.e. whether users consider it to be a good or bad item that is, the rating with respect to the user mean. We denoted to Rui as the rating given by user u to item i, Uu as the mean (average) rating given by user u across all items they have rated, and Ui as the set of users who have rated item i.

$$\mathrm{Ti} = \sum_{u \in \mathrm{Ui}} \left(\frac{\mathrm{Rui} - \mathrm{Uu}}{|Ui|} \right) \tag{2}$$

In the Computation of the prediction rating(Pui) of item i by user u, parameter (σ) regulates the item and user mean contributions and we denoted the positive tendency as(>0) and negative tendency as(<0) This algorithm by testing four cases is constructed by the expression (3), as shown at the bottom of the next page.

The KNNBasic algorithm in the third step computes similarities based on four cases and collaborative filtering between user ID and item ID, and makes predicted ratings based on the k-nearest neighbors. Finally, the output features with predicted ratings feature.

B. BINARY LABEL ASSIGNMENT AND DATASET BALANCING TECHNIQUE

We can bring imbalanced data, often defined as a classification issue in which unequal representation among the classes exists. Several existing real-world datasets are

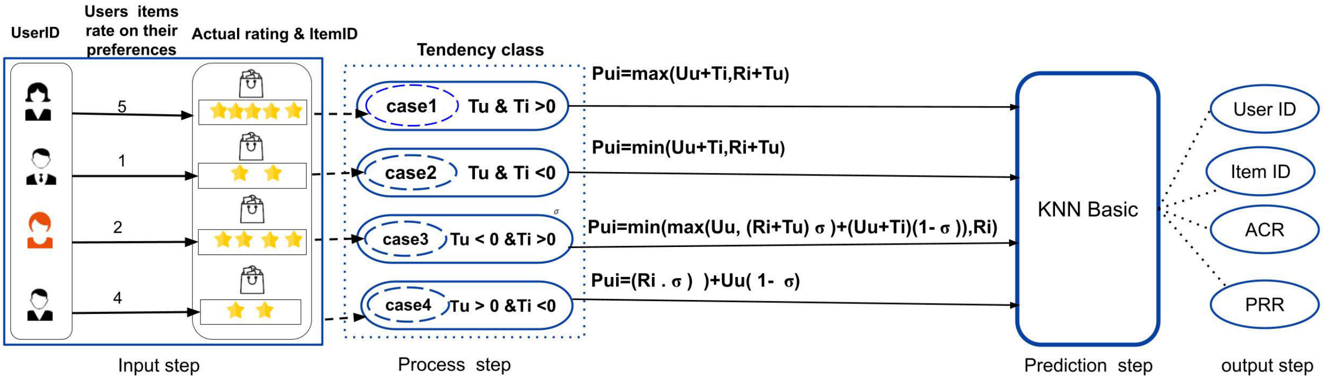


FIGURE 2. (TCF)Framework.

inherently imbalanced such as Amazon review dataset. There are several domains where the issue of class imbalance occurs, for example, computer science, recognizing fraud, health care diagnosis, and telecommunication and binary label classification [28].

Unbalanced training data have a significant detrimental effect on performance models of larger sample sizes. This data in classification problems can be handled in several methods, for instance, random over-sampling, synthetic minority oversampling (SMOTE), under-over-sampling, selecting the algorithm carefully, experimenting with the loss function, and resolving an issue with anomaly detection [29].

To handle imbalanced data in classification, prevent overfitting, and raise the accuracy value of our suggested model, we covered the random over-sampling strategy in this paper. We select the random over-sampling approach over other approaches for some causes: First, the under-sampling approaches reduce the dataset size by removing instances from the majority class, which can lead to the loss of potentially useful information, but random over-sampling avoids this issue by maintaining all data points. Second, while SMOTE generates new synthetic instances, which can be beneficial, it is more complex to implement. Random oversampling, on the other hand, is simpler and quicker, and in some cases, SMOTE can introduce noise or outliers if the synthetic examples do not accurately represent the true distribution of the minority class, however random oversampling by duplication avoids this risk [30].

Assuming that, when comparing the user's actual and predicted ratings of the product, both are more than or equal to 3 or 4 with two cases first with using TCF and second without it for binary classification and after extracting an extra feature with a helpful name, the samples are balanced based on the largest sample.

C. PROPOSED BALANCED CNN-BiLSTM ARCHITECTURE

Starting from structural features, such as review length, number of sentences, and word count, features are extracted. The structure of the content will have an impact on the quality of reviews. For example, too many words in long reviews may contain vague or ambiguous details that may affect the quality of the product's recommendation, making it of poor quality. The star rating and review summary constitute the metadata feature extraction. Customers will rate their star ratings for the product experience alongside each written review. It is the discrepancy between the true average star rating of items and the star rating given to products by consumers that defines a review so highly.

To enhance the accuracy of helpful review analysis on product reviews, we combined the advantages of TCF, the balanced CNN-BiLSTM model to propose the BHRQUT model. Initially, the data set is collected based on the user rating and information tendencies. After pre-processing using NLP tasks and optimizing the feature of the user profile balanced by binary label assignment and dataset balancing technique, the embedding layer captures the semantic meaning of words and sentences to allow for the prediction of user preferences for items. This layer converts categorical data into continuous vectors(0,1), enabling more efficient analysis, classification, and prediction tasks.

$$\mathrm{P_{ui}} = \begin{cases} \max(\mathrm{Uu} + \mathrm{Ti}, \mathrm{Ri} + \mathrm{Tu}) & \text{if Ti} > 0 \text{ and Tu} > 0 \\ \min(\mathrm{Uu} + \mathrm{Ti}, \mathrm{Ri} + \mathrm{Tu}) & \text{if Tu} < 0 \text{ and Ti} < 0 \\ \min\left(\max(\mathrm{Uu}, (\mathrm{Ri} + \mathrm{Tu})\sigma + (\mathrm{Uu} + \mathrm{Ti})(1 - \sigma))\,, \mathrm{Ri}\right) & \text{if Tu} < 0 \text{ and Ti} > 0 \\ (\mathrm{Ri} \cdot \sigma) + \mathrm{Uu}(1 - \sigma) & \text{if Tu} > 0 \text{ and Ti} < 0 \end{cases} \tag{3}$$

After that, we fed the feature vectors/ attributes to CNN for recommendation.

Convolutional Neural Network (CNN) layers are used for feature extraction on input-balanced data. In our model, we add four ConvID layers each layer output is computed based on its inputs as well as the word embedding size, max length size, kernel filter size, and biases of neurons. The mathematical expression of the output of the convolution layer, we constructed in equation (4)

$$Y_i^{(n)} = \sum_{j=0}^{h-1}\sum_{f=0}^{D-1} X_{(i.s+j-p)}K_{j,f}^{(n)} + b^{(n)} \tag{4}$$

We have two matrices, an input text sequence is represented as a matrix X of dimensions LxD, where L is the length of the sequence tokens and D is the dimensionality of the word embeddings and kernel matrix K of dimensions h× D, where h is the height of the filter and D matches the embedding dimension. Y is the output vector at position n, The stride determines(s) as the number of words that filter moves at each step on each element position i,j, where (i+j,f) is the position of the element of the input matrix X and (j,f) is the position the element the filter matrix K. The extra tokens as zeros are added to the beginning and end of the sequence to control the output dimensions with padding(p). b is biases associated with each layer can be regulated by equation 5.

$$\Delta b_i^{(n)} = -\frac{\beta}{n}\cdot\frac{\theta\cdot L}{Y_{i,j,f}} + m\cdot\Delta b_i^{(n-1)} \tag{5}$$

where b_i represents the bias of a neuron. The parameter for regularity is indicated by β, and the variables n and m represent the total number of samples used for training and testing, respectively. $\frac{\theta L}{Y_{i,j,f}}$ is the loss (MSE) function with respect to the output at position (i, j, f) with the f-th filter. After that, we set one of the max pooling layers to create a more abstract and compact representation of the input data, making subsequent layers focus on higher-level features. The max pooling operation for a given feature map can be mathematically described as Equation 6:

$$\mathrm{MP}_{i,n} = \max(Y_{i,s+f,n}) \quad \text{for} f = 0, 1, \ldots, m-1 \tag{6}$$

Where MP-i,n is the element in the output feature map at the position (i,n), Y-i,n is the element in the input feature map at position (i,n) and the maximum is taken over a window of pooling size (m) along the dimension L of length of the sequence tokens. We improved the network's ability to generalize to unseen data, these data are coupled with BiLSTMs in the CNN BiLSTM architecture to enhance sequence learning as in Figure 3. BiLSTM combines user interests in both directions to alleviate problems related to cold starts. BiLSTM deploys two LSTMs rather than one on the input sequence. One LSTM processes the input sequence in its original form, while the other processes a modified or reversed version [30]. This dual LSTM architecture enables bidirectional pattern recognition, considering both the forward and backward representations of the input sequence.

The mathematical representation of a BiLSTM involves forward and backward LSTM layers. The output of the BiLSTM at each time step of the forward and backward hidden states as the following equations, where we denote in equation 6 the htf is the forward hidden state at time t with hidden state $ht-1$ and cell state ct_{-1}, and htp in equation 7 is the backward hidden state at time t, Xt as the input at time t with hidden state ht+1and cell state ct+1.

$$h_{tf} = \mathrm{lstmf}(X_t, h_{t-1}, c_{t-1}) \tag{7}$$

$$h_{tb} = \mathrm{lstmb}(X_t, h_{t+1}, c_{t+1}) \tag{8}$$

The output of the BiLSTM at each time step t is the concatenation of the forward and backward hidden states written in the equation9.

$$\mathrm{BiLSTM} = (h_{tf}\cdot h_{tb})_l \tag{9}$$

Finally, the fully connected is merged from three (Flatten layer, Dense layer, and output layer) that combined from the width of the input sequence (L), the height of the input sequence (D),b is the bias term, the number of output y^n, and activation function sigmoid (S) with classification values(0,1) [31] as in equation (10)

$$Y = S(L\cdot D\cdot y^n) + b \tag{10}$$

D. FINE-TUNING AND EVALUATION

Fine-tuning is the process of making little adjustments to a process to get the desired outcome or performance. Fine-tuning deep learning is the process of programming a new deep learning algorithm using the weights of an older deep learning algorithm. Weights are used to connect every neuron in one layer of the neural network to every other neuron in the layer above it.

The fine-tuning technique significantly shortens the time required to create and process a new deep learning algorithm since it takes data from an existing deep learning algorithm. Deep learning is being advanced because it will make the process of creating new algorithms much simpler and faster. The evaluation of each feature's values and generating the quality score are done using a weighted sum. The three categories of properties that make up the criteria are structural, metadata, and readability. The reviews are more qualified, as seen by the higher score results. This phase results in a quality score and a ranking of the reviews according to quality.

Model evaluation is the process of using several evaluation metrics to understand how effectively a deep learning and machine learning model is performing, as well as its benefits and drawbacks. Early on in a research project, a model's efficacy must be assessed, and model assessment helps with model monitoring. Examples of classification performance measures include the RMSE, MAE, MSE, Accuracy, Precision, Recall, and F1-Score.

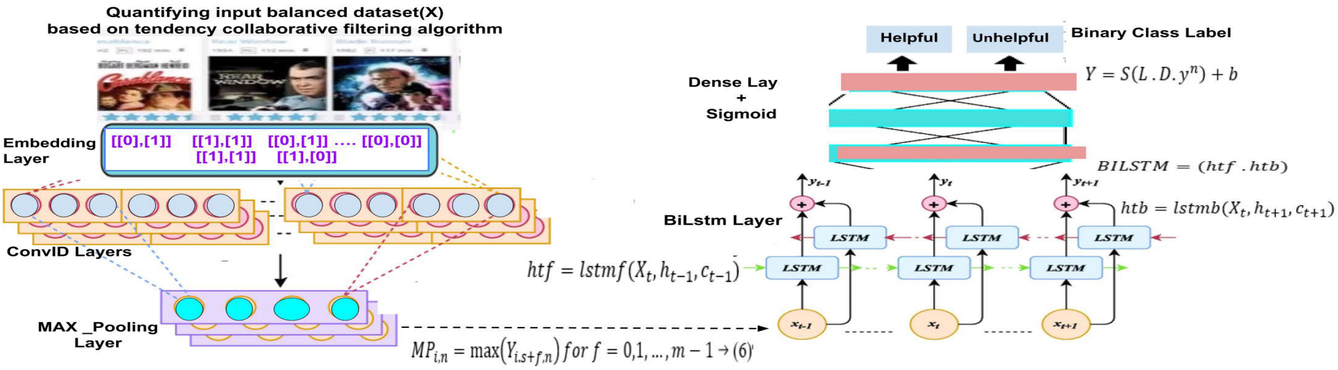


FIGURE 3. Formulation of BHRQUT architecture.

IV. EXPERIMENTS

This section presents extensive experiments on four categories of Amazon review datasets to assess BHRQUT's performance. Details are provided regarding the datasets, baseline procedures, experiment design, the parameters that were set to design our study, and outcomes.

A. BASELINES

The performance of this paper is evaluated using six recommendation systems, three of them rating-review based dep recommender systems (SLCABG, AFRAM,Deep Feature Extractor), two of them vote-review based dep recommender systems(machine-Review Generation Model, RHRM), and one prediction rating based on analysis of user's review and different users' tendencies(DUPIA). The explanation of these models was provided as follows:

1) **RHRM** [9]: Review of the helpfulness-based recommendation methodology is based on a hybrid CNN-BiLSTM model to classify helpful reviews.
2) **AFRAM** [21]: An aspect-based fashion recommendation model with an attention mechanism is developed to extract the latent features in customers' reviews and ratings in two paths each of which has CNN and LSTM neural networks.
3) **machine-Review Generation Model** [22]: Machine generation review-based context and attention models on two levels, human write review level and machine generation review level with DNN and RNN is investigated for the explanation of helpful review.
4) **DUPIA** [24]: Differentiating Users' Preferences and Items' Attractiveness model is proposed to predict users' ratings by estimating users' reviews and different users' considerations by adopting CNN and attention mechanisms.
5) **SLCABG** [32]: Sentiment lexicon combines CNN and attention-based BiGRU model is improved to enhance review sentiment analysis with deep learning algorithms on extracting the sentiment review good or bad.
6) **Deep Feature Extractor** [33]: Features matrix is produced by using LDA analysis to extract item characteristics from customer reviews. This matrix is transformed by DNN into a dense users-deep features matrix to predict rating and helpful recommendations using MF.

TABLE 1. Amazon score-5 items number of review and ratings.

Item Category	Number of Reviews (5-score)	Number of Reviews (5-score) with our model
Books-5(C3)	27,164,983	1000000
reviews-Clothing-Shoes and Jewelry(C4)	11,285,464	1000000
Musical-Instruments-5(C2)	231,392	231,392
Video-Games-5(C1)	497,577	497,577

B. EXPERIMENTS SETTINGS

1) DATASET

In recent years, the availability of large-scale datasets has become crucial for advancing research in various domains. Notable among these are datasets such as those provided by Amazon review data (2018). These datasets are a priceless resource for practitioners and researchers alike, facilitating the creation and assessment of algorithms and models for a variety of uses. In our experiments, we adopted four groups from Amazon review data [33]. For two categories of small samples (Video Games(C1), Musical Instruments(C2)), we chose all reviews from both of them, and we chose (1000000) reviews from two categories of large samples (Books-5(C3), reviews-Clothing-Shoes and Jewelry(C4)) as displayed in Table 1.

The data collection includes product reviews and data from Amazon, totaling more than 142.8 million reviews between May 1996 and July 2018. This dataset consists of links (also viewed/also purchased graphs), product metadata (descriptions, category details, price, brand, and image features), and reviews (ratings, text, helpfulness votes).

TABLE 2. **The illustration of amazon review composition.**

Attributes Name	Value
overall	0.5
vote	67
verified	True
reviewerID	AAP7PPBU72QFM
asin	0151004714
style	Hardcover
reviewerName	D. C. Carrad
reviewText	This is the best novel I have read in 2 or 3 y...
summary	A star is born
unixReviewTime	937612800
image	NaN

Each review includes the following information: (1) reviewer ID and name; (2) review item ID; (3) Boolean value for review verified; (4) rating item based on individual user reviews (1)-(5), (5) review text summary; (6) the date the review was published, (7) Review text that is related user's evaluation of the product, (7) Style is some item specifications, (9) total feedback for reviewer's review(Vote), (10) image of item The attribute data utilized in the Amazon Review data CSV file is listed in Table 2. with 11 columns.

2) DATA PREPROCESSING

In the preprocessing datasets stage, we achieve the following tasks: (i) Training the data in two cases both of them with two paths, in case 1 we do a re-rating of items based on looking at the differences between users and items by calling the Tendencies class from the surprise package, and filtering the reviews in two paths case (1_1) and case (1_2), in case (1_1) is produced when the average rating (avg) whose avg= (ACR+ PRR)/2 is greater than or equal to 3 when the highly helpful reviews (avg >=0.3) and unhelpful reviews (avg= < 0.2) as the training dataset, and case (1_2) is created when the avg is greater than or equal to 4 highly helpful reviews (avg >= 0.4) and unhelpful reviews (avg= < 0.3) as the training dataset according to what was suggested in [34].

In case 2 we filter the reviews also in two paths case (2_1) and case (2_2) based on the actual rating of the dataset without using the Tendencies class. In case (2_1), when the actual rating(ACR) is greater than or equal to 3, the highly helpful reviews (ACR >= 0.3) and unhelpful reviews (ACR= < 0.2) as the training dataset according to what was suggested in [32]. In case (2_2) when the Actual rating is greater than or equal to 4 highly helpful reviews (ACR >= 0.4) and unhelpful reviews (ACR= < 0.3) as the training dataset.

(ii) In both cases after the filtering approach, we get two imbalanced samples (helpful, and unhelpful) that require a balancing of samples as shown by the performance accuracy of the model, we augmented each dataset using the randomly oversampling technique based on majority sample size(helpful) [35].

(iii) After the balancing data we prepared the dataset using the pipeline techniques of text preprocessing, these techniques were applied with the NLTK (Natural Language Toolkit) and kearas packages to remove missing values, Stem words, stop words, and tokenize the sentences into vectors of words.

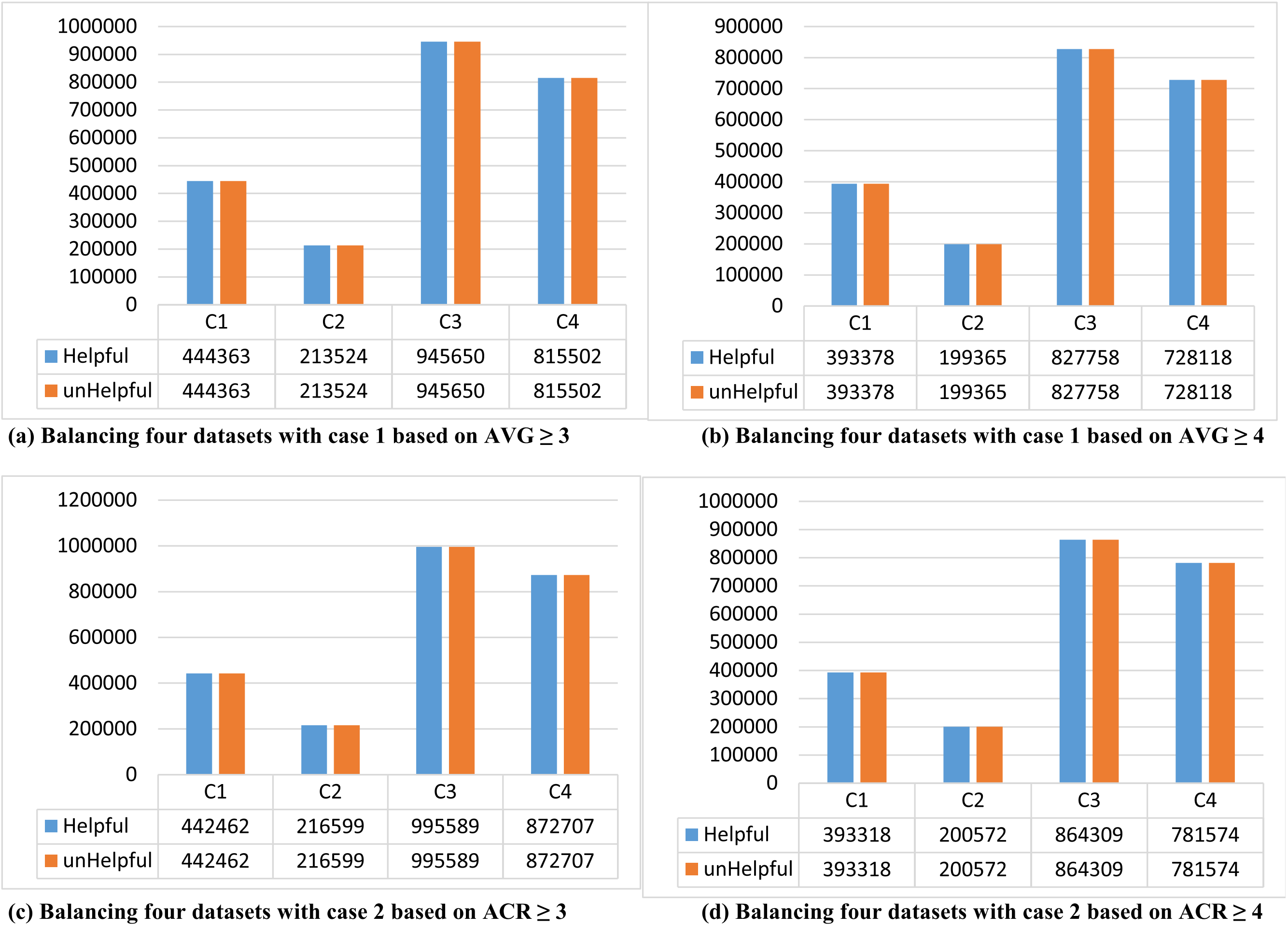
(iv) The dataset's maximum number of words in each review after pre-processing is counted, because the maximum length of reviews helps set a uniform sequence length when tokenizing text, ensuring that all reviews are either padded to this length or truncated, leading to consistent input sizes for models.

(v) We used 5-fold cross-validation to assure the accuracy of the model evaluation, which divides the dataset into training and testing sets, with 80% and 20% of each, respectively.

3) REPRODUCIBLE RESULTS

To utilize a summary of the review text and helpful features with values(0,1) that are filtered from the average rating attribute computed from the tendencies-based collaborative filtering algorithm, we preprocessed the filtered features using natural language processing techniques. As the input layer for training the balanced CNN-BiLSTM model, we learned the words using a word embedding approach by mapping various word indices to embedding vectors during training data. The input dimension 50,000 is the most frequent word representing each token by a unique integer from 0 to 49,999. The output embedding dimension is the size of the dense vector space in which words are embedded 50 to reduce the model complexity and computational cost, which can be useful with a large dataset. These vectors capture semantic information about the words in the context of employed data. This means the embedding layer is created as a lookup table where each word of the top 50,000 frequent words in the vocabulary is associated with a 50-dimensional vector. Since we adopted four categories of the dataset, the maximum length of the text statement in the dataset is different, we observed (21,22,23,24,25,30,38,44) values of max length sequence text for different categories.

The input text sequence data feature was passed through the kernel size with 2× 2. The model can capture and interpret local word patterns without losing crucial information by utilizing a 2 × 2 kernel because the text sequences are relatively short (summary of review text). In this case, predicting helpfulness may depend on identifying brief but significant word pairs. Subsequently, four ConvID layers were utilized to extract the feature maps with a first filter size of 100 to filter the wide variety of features from the



	C1	C2	C3	C4
🟦 Helpful	444363	213524	945650	815502
🟧 unHelpful	444363	213524	945650	815502

(a) Balancing four datasets with case 1 based on AVG ≥ 3

	C1	C2	C3	C4
🟦 Helpful	393378	199365	827758	728118
🟧 unHelpful	393378	199365	827758	728118

(b) Balancing four datasets with case 1 based on AVG ≥ 4

	C1	C2	C3	C4
🟦 Helpful	442462	216599	995589	872707
🟧 unHelpful	442462	216599	995589	872707

(c) Balancing four datasets with case 2 based on ACR ≥ 3

	C1	C2	C3	C4
🟦 Helpful	393318	200572	864309	781574
🟧 unHelpful	393318	200572	864309	781574

(d) Balancing four datasets with case 2 based on ACR ≥ 4

FIGURE 4. Balancing data for four categories of dataset in case1 and case2.

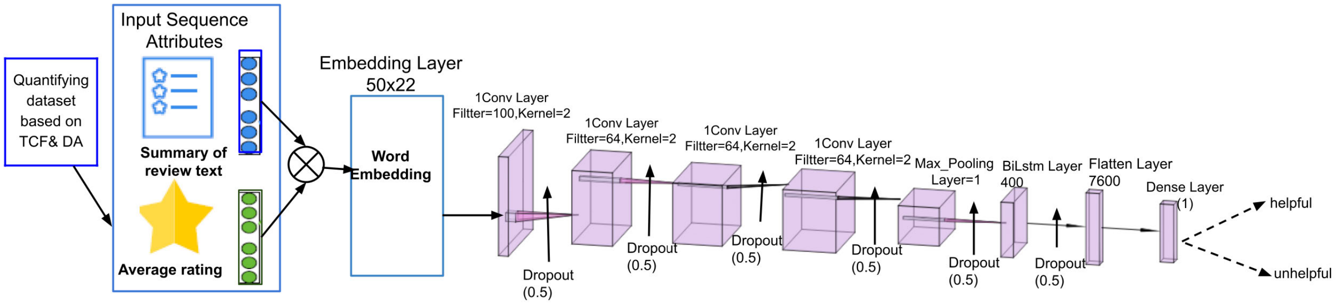


FIGURE 5. Construction of the proposed model.

input data and balance the model's complexity. the other three layers of ConvID each of them 64 filters, especially in deeper models the dimensionality reduction is important to focus on more abstract and higher-level features. In addition to the specified input sequences' spatial dimensions, the max pooling layer down scales. It takes into account the highest value of every input feature in each filter kernel's pool. The kernel in this layer is 1 × 1 doing so, the feature maps' spatial dimensions—height and width are decreased but crucial characteristics are maintained, which reduces computational complexity and guards against overfitting. After that, the outputs of the max-pooling layer are input to one layer of BiLSTM with (200) hidden units. Since this is a bidirectional LSTM, having a substantial number of units ensures that both forward and backward contexts are well-captured, enriching the overall representation of the

TABLE 3. Hyper parameters setup.

Hyper-parameters	Value
Batch Size	256
Number of epochs	20
Activation function	Relu
Optimizer	Adam
Loss	mse
Dropout	0.5
Learning rate	0.01
MAX-NB-WORDS	50000
EMBEDDING-DIM	50
maxLen	21,22,23,24,25,30,38,44
Dense Activation function	sigmoid

sequence, therefore the 200 hidden units give the model a large capacity to learn and remember patterns over time. This is particularly useful in NLP tasks, where the model needs to understand the context of words across a sequence. After each layer, we set the dropout rate at 0.5 to make the model more resilient to variations in the data and the batchNormalization layer to solve the issue of overfitting. Because of the relu function's representational sparsity and computing efficiency, it was employed as an activation function in every layer.

5-folds are created from the dataset, each fold uses 80% of the data to train the model and the remaining 20% to validate it. This technique eliminates the possibility of overfitting to a particular subset of the data and aids in evaluating the model's capacity for generalization. The number of iterations of model with 20 epochs and 5-fold cross-validation is a balanced and efficient approach that allows the model to train and evaluate datasets effectively. These iterations ensured that the model had enough training time to learn meaningful patterns while minimizing the risks of overfitting and ensuring reliable performance evaluation across different subsets of the data.

By setting the model with an initial learning rate of 0.01 and Adam optimizer with the loss function known as MSE, accurate results by minimizing the MSE. A layer that is fully connected, made up of three layers: flatten layer, dense layer with a sigmoid function for output class because in binary classification, where each input sample can only belong to one of two classes as (helpful or unhelpful), and output layer. The structure of the proposed model and the parameters setup are shown in Figure 5 and Table 3. All experiments are implemented using Tensorflow, Keras, and Surprise packages on Python 3 and Nvidia titan xp GPU.

4) ASSESSMENT CRITERIA

We employed Accuracy, Precision, Recall, F1_score, MAE, RMSE, and MSE as metrics to assess the classification performance of our model with the four categories of the Amazon review dataset, and the equations (11)-(17) are shown in the formulation of these matrices respectively. In our study, the confusion matrix consists of two rows, the first row (TN, FP) and the second row (FN, TP). True Positive (TP) is the number of actual helpful reviews that were correctly predicted as helpful, True Negative (TN) is the number of actual unhelpful reviews that were correctly predicted as unhelpful, False Positive (FP) is the number of actual helpful reviews that were incorrectly predicted as helpful, and False Negative (FN) is the number of actual helpful reviews that were incorrectly predicted as unhelpful. The most widely used method among evaluation methods is accuracy. Accuracy (A) can be briefly interpreted as the ratio of all helpful reviews to all the reviews. The total classification findings calculate the ratio of accurate classifications to helpful and unhelpful reviews.

$$A = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{11}$$

Precision(P) is the proportion of genuinely favorable evaluations to those that the model has classified as favorable.

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{12}$$

The percentage of predictive helpful reviews to all truly helpful reviews was one of the model's Recall(R) metrics.

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{13}$$

The F1_score is an average that weighs accuracy and recall equally. A higher F1score indicates that the recommender system is more capable of classifying information.

$$\mathrm{F1\_SCORE} = 2 \cdot \frac{P \cdot R}{P + R} \tag{14}$$

The MAE and RMSE are statistical evaluation metrics that evaluate prediction performance by comparing the difference between actual target label value and predicted target label value. The average of the values of the derived magnitudes of errors, which is also called the mean-absolute-error, can be obtained by utilizing the absolute value operation, which is described below,

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| \hat{y}_i - y_i \right| \tag{15}$$

RMSE (Root Mean Squared Error) is a common way to measure error for a given actual and predicted values, the Root Mean Square Error (RMSE) provides a value weight that is comparatively high. The matching suggestion prediction is more accurate when the value is low., RMSE is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2} \tag{16}$$

TABLE 4. Validation and training loss results of our model for each category (C1, C2, C3, and C4) in each case name.

Case Name	Category of the Dataset	Loss value	
		Val	Train
case (1_1)	C1	0.04	0.03
case (1_1)	C2	0.03	0.02
case (1_1)	C3	0.05	0.03
case (1_1)	C4	0.05	0.04
case (1_2)	C1	0.07	0.05
case (1_2)	C2	0.05	0.03
case (1_2)	C3	0.08	0.06
case (1_2)	C4	0.07	0.05
case (2_1)	C1	0.05	0.03
case (2_1)	C2	0.03	0.02
case (2_1)	C3	0.05	0.04
case (2_1)	C4	0.05	0.04
case (2_2)	C1	0.07	0.04
case (2_2)	C2	0.05	0.03
case (2_2)	C3	0.09	0.06
case (2_2)	C4	0.07	0.06

where $\hat{y}_i - y_i$ is the difference between the predicted target value and actual target value, and n is the number of values. We use MSE as a loss function [36]. To learn the parameters of our model, the objective function, F, can be expressed as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} \hat{y}_i - y_i \tag{17}$$

C. EXPERIMENTAL RESULTS

The effectiveness of our proposed model was assessed on four categories of datasets (C1, C2, C3, AND C4) with two cases each case with two paths, therefore there are (case (1_1), case (1_2), case (2_1), case (2_2) for each category aimed at classifying two classes (helpful and unhelpful). We prepared the data using the pipeline of text preprocessing. To investigate the effect of our model on each scale, we used both small and large datasets as we displayed in section (A.1). Furthermore, in Table 4. Based on how far the prediction deviates from the true value, a loss function, known as a cost function, considers the probability or uncertainty of a prediction. This enables us to have a more thorough understanding of how the model is performing. We displayed validation and training loss function values and experimental results prove that the case (1_1) has the lowest error value on each category name.

The most widely used method among evaluation methods is accuracy. We conducted the training accuracy as shown in Table 5 to assess the prediction performance of the proposed recommendation framework. Numerous graphical measures can be used to express a model's predictive ability. the proposed model achieved a high accuracy of 97% and a low loss value of 0.03 with the C2 dataset in case (1_1) and the lowest accuracy of 88% and a high loss value of 0.09 in case (2_2) with C3 dataset. We found the performance worsened in case(1_2) when avg >=4 and in case (2_2) when the rating >=4. Thus, the optimal helpful ratio should be avg>=3 to improve the recommendation performance.

A confusion matrix is a commonly structured matrix used to evaluate the performance of the model in classification techniques. In our model, this matrix shows the predicted classes along the horizontal axis and the actual classes along the vertical axis. This configuration made it easier to distinguish between positive and negative predictions. The diagonal line of the matrix indicates how many true positive and true negative results the classifier detected; this is a useful tool for computing the contents of the classification report for various training data categories. This matrix's results demonstrated a rise in the number of true positives (TP) for each case and class, improving the suggested model's overall quality and helping to raise the degree of model accuracy. Figure 6 illustrates subfigures of the classification confusion matrixes generated during the testing phase for our proposed model. Subfigures (a, b, c, d) provide the confusion matrices for the four cases on four datasets (C1, C2, C3, C4) respectively.

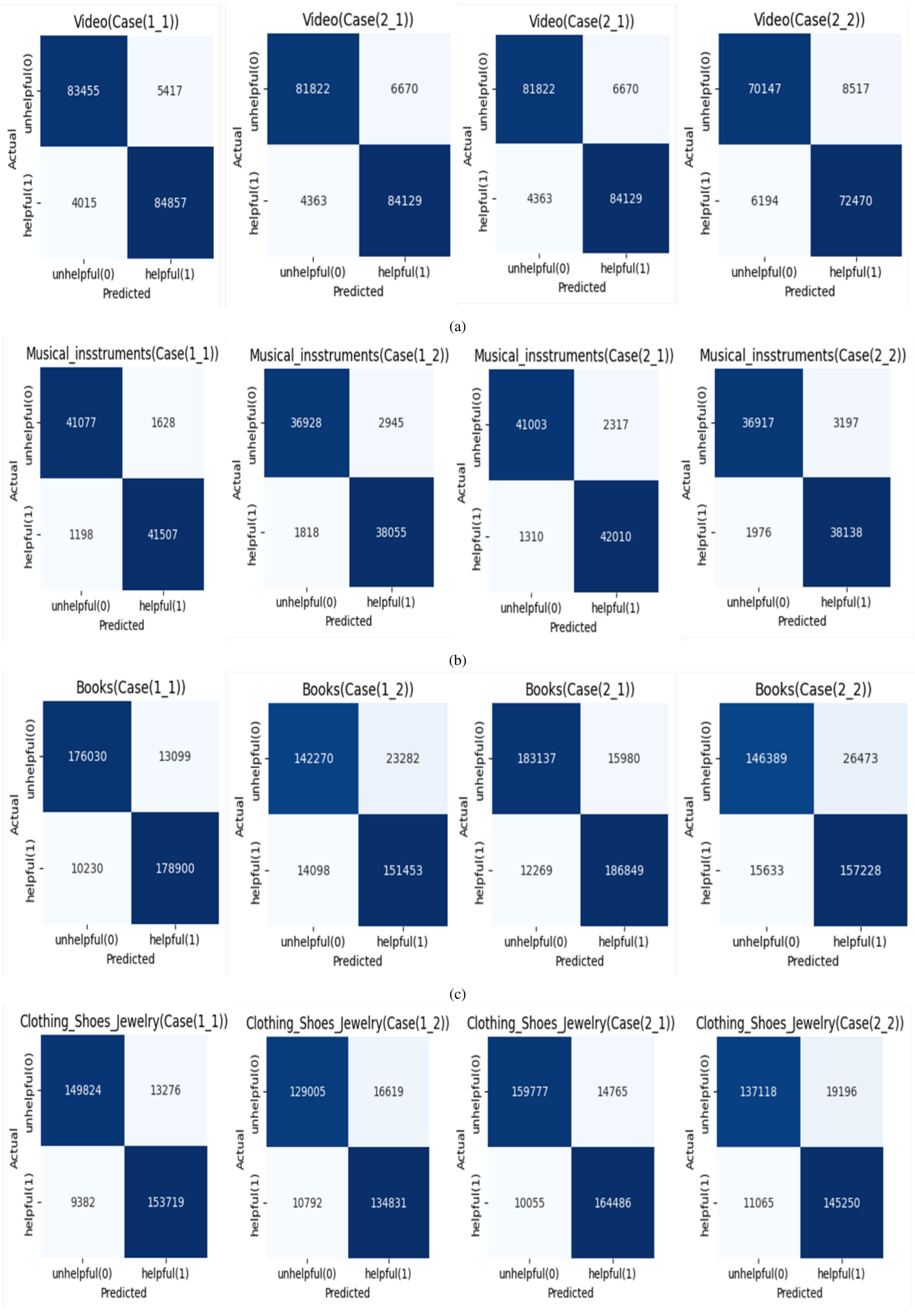
We assess the predictive accuracy of the our(BHRQUT) model by comparing it to other models, including RHRN [9], AFRAM [21], machine-generated reviews [22], DUPIA [24], and Deep Feature Extractor [33], across four Amazon review datasets. We used RMSE, MSE, and MAE to measure the effects of different users' opinions as in Table 6.

Additionally, we investigate the impact of the max length of review text and we set the optimal sequence length to compare the classification performance with the RHRM [9] and SLCABG [32] models in the book category using Precision, Recall, F1-Score, and Accuracy metrics. As a result, the optimal max length was 24 with case (1_1) of our model, and SLCABG model with max text length 648 is performing better than the RHRM model with length 2817, as detailed in Table 7.

We select the maximum text sentence length in the dataset to conduct the effect of maximum length sequences on the performance of our model with different cases and four datasets as shown in Table 8. We find with max-length (21) that is the less length with the best result in accuracy and loss function.

V. DISCUSSION

This paper presents a new model to analyze the review of users on products to enhance the decision-making on purchasing. Before inputting the text sequences into the deep model, the dataset is quantified using the tendency class to predict ratings based on different opinions to improve the helpful features and accuracy of the model. Then we use data augmentation with a randomly over-sampling technique to balance the imbalanced sample (helpful and unhelpful), we use NLP tools to analyze the word summary of the review text. We selected the maximum input sentence length for the text of the summary review text feature in the data



(a)

(b)

(c)

(d)

FIGURE 6. Confusion matrices of the model in case1 and case2 with two paths for C1, C2, C3, C4 datasets.

TABLE 5. Validation and Training (Precision, Recall, F1-score, and Accuracy) results of our model for each category (C1, C2, C3, and C4) in each case name.

category of Amazon Score-5	Case Name	Precision		Recall		F1-Score		Accurecy	
		Val	Train	Val	Train	Val	Train	Val	Train
C1	case (1_1)	94%	96%	95.4%	98%	95%	97%	95%	97%
	case (1_2)	89.4%	93%	92%	95%	90.6%	94%	90.4%	94%
	case (2_1)	92.6%	94%	95%	98%	94%	96%	94%	96%
	case (2_2)	89.8%	93%	92.2%	96%	91%	95%	90.8%	94%
Average								92.55%	95.25%
C2	case (1_1)	96.2%	97%	96.8%	99%	97%	98%	97%	98%
	case (1_2)	92.8%	96%	95.4%	98%	94%	97%	94%	97%
	case (2_1)	94.8%	97%	97%	99%	96%	98%	96%	98%
	case (2_2)	92.4%	95%	95%	98%	94%	97%	94%	97%
Average								95.25%	97.5%
C3	case (1_1)	93.4%	95%	94.4%	96%	94%	96%	94%	96%
	case (1_2)	86.6%	90%	91.6%	95%	89%	93%	88.8%	92%
	case (2_1)	92%	95%	93.8%	96%	93%	95%	93%	95%
	case (2_2)	85.6%	89%	91%	96%	88%	92%	88%	92%
Average								90.95%	93.75%
C4	case (1_1)	92.2%	93%	94.2%	97%	93%	95%	93%	95%
	case (1_2)	89.2%	91%	92.6%	95%	91%	93%	90.8%	93%
	case (2_1)	91.8%	94%	94.2%	96%	93%	95%	93%	95%
	case (2_2)	88%	91%	93%	95%	90.8%	93%	90%	93%
Average								91.7%	94%

TABLE 6. Performance comparison of helpfulness review analysis models based on RMSE, MSE, and MAE.

Models	RMSE				MSE				MAE			
	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
RHRM [9]			0.592								0.468	
AFRAM [21]								1.098				0.801
machine-generated reviews [22]	0.881											
DUPIA [24]		0.833										
Deep Feature Extractor[33]					1.581	1.475	1.221	1.211				
BHRQUT on case (1_1) with avg >=3	0.204	0.161	0.218	0.223	0.041	0.026	0.047	0.050	0.071	0.038	0.089	0.091
BHRQUT on case (1_2) with avg >=4	0.265	0.219	0.290	0.264	0.070	0.048	0.084	0.069	0.133	0.073	0.154	0.139
BHRQUT on case (2_1) with ACR >=3	0.217	0.184	0.233	0.225	0.047	0.034	0.054	0.051	0.076	0.047	0.102	0.101
BHRQUT on case (2_2) with ACR >=4	0.263	0.226	0.301	0.268	0.069	0.051	0.089	0.072	0.126	0.083	0.179	0.145

TABLE 7. compression of the impact of the max length of the input sentences on the (BHRQUT) model with other models.

Models	measured score for helpfulness	Max-length	Precision	Recall	$F1\_score$	Accuracy
RHRM [9]	vote	2817	85.54%	88.73%	87.10%	86.14%
SLCABG [32]	rating	648	92.9%	93.8%	93.3%	93.4%
Ours(BHRQUT) case (1_1)	Average of rating	24	93.4%	94.4%	94%	94%

sets as we fixed the maximum number of words in all sentences. 5-folds are created from the dataset, each fold uses 80% of the data to train the model and the remaining 20%a to validate it. This technique eliminates the possibility of overfitting to a particular subset of the data and aids in evaluating the model's capacity for generalization. The number of iterations of model with 20 epochs and 5-fold cross-validation is a balanced and efficient approach that allows the model to train and evaluate datasets effectively. It ensured that the model had enough training time to learn meaningful patterns while minimizing the risks of overfitting and ensuring reliable performance evaluation across different subsets of the data. The words (tokens) are transformed into dense vectors of a set size by the embedding layer, where words with comparable semantic content have similar vector representations. This reduces the dimensionality of the input and enables the model to capture links between words that have a closer meaning. This saves computing power and helps the model to concentrate on important properties. Then CNN is adopted to assign different weights to different information contained in the user profile, followed by the bidirectional representations obtained using the BiLSTM attention

TABLE 8. Effect of the max sequence length of the input sentences on the performance of the (BHRQUT) model.

Dataset name	Case Name	Max-length	Accuracy	Loss
C1	case (1_1)	30	95%	0.04
	case (2_1)	30	94%	0.05
	case (1_2)	30	91%	0.07
	case (2_2)	38	91%	0.07
C2	case (1_1)	21	97%	0.03
	case (2_1)	23	96%	0.03
	case (1_2)	24	94%	0.05
	case (2_2)	25	94%	0.05
C3	case (1_1)	24	94%	0.05
	case (2_1)	23	93%	0.05
	case (1_2)	22	89%	0.08
	case (2_2)	23	88%	0.09
C4	case (1_1)	23	93%	0.05
	case (2_1)	23	93%	0.05
	case (1_2)	44	91%	0.07
	case (2_2)	44	90%	0.07

network that is combined from forward LSTMs and backward LSTMs networks. bidirectional LSTMs are instrumental in increasing the amount of information available to the network and improving the context available to the algorithm for data storage to improve the performance of current helpful review analysis models in the users' reviews analysis domain of the product.

We applied the proposed BHRQUT model to four Amazon review categories datasets and obtained the best RMSE, MSE, MAE, Accuracy, Recall, Precision, and F1-score for our model and compared these results with our baselines as shown in Tables(6,7). In four cases, the BHRQUT model with RMSE measure shows better performance up 76.84 percent improvements compared with RHRM [12], machine-generated reviews [22], and DUPIA [24] including Video Games(C1), Musical Instruments(C2), and Books-5(C3) categories. in four cases, the BHRQUT model with RMSE measure shows the best performance up 76.84 percent improvements compared with RHRM [12], machine-generated reviews [22], and DUPIA [24] including Video Games(C1), Musical Instruments(C2), and Books-5(C3) categories. Also, MSE and MAE measures best performance up 98.35 and 95.26 percent improvements respectively compared with RHRM [12], AFRAM [21], and Deep Feature Extractor [33] including Video Games(C1), Musical Instruments(C2), Books-5(C3), reviews-Clothing-Shoes and Jewelry(C4) categories.

We selected the maximum input sentence length for the text of the summary review text feature in the data sets as we fixed the maximum number of words in all sentences. The proposed strategy works better for most of the product categories, as shown in Tables 7-8 on the impact of the maximum input text statement length on the performance of the model. We found that the performance of the model is the best max text length when the input length is 21 tokens and when we compared the effect of the maximum sentence length on all our model cases, we noticed that as the length decreased, the accuracy of the model increased. Also, we observed Our proposed BHRQUT model was more efficient when the max text length was 24 for the book category on RHRM [9] model with a max text length of 2817 according to the Precision, Recall, F1-Score, and Accuracy metrics with (9.19%, 6.39%, 7.92%, and 9.12%) respectively, and also on SLCABG [32] model with max text length 648 according of the Precision, Recall, F1-Score, Accuracy metrics with (0.54%, 0.64%, 0.75%, and 0.64%) respectively. This is evidence that shorter sequence lengths can improve performance through several means, including reducing noise, improving accuracy, preventing overfitting, increasing computational efficiency, and ensuring that the model focuses on the most relevant data. As a result, models often generalize more successfully and perform better in real-world applications.

We employ the small size dataset and large size dataset in this study and then we explored the results of our model are better than other helpfulness analysis models in each dataset size. in predicting helpful reviews with case (1_1) for all four datasets of our models. The experimental results show that the classification performance of the deep learning model (balanced CNN-BiLSTM) in case (1_1) is significantly better than all cases of our proposed model, but case (1_2) and case(2_2) in all datasets are less than case (1_1)and case (2_1) cases results. also, we observed Musical Instruments(C2) dataset is better than other datasets in (1_1)and(2_1) cases with all performance metrics results, this means the measured ratio (avg>=3) or (ACR>=3) are better than ratio based on (avg>=4) or (ACR>=4), and the experiments show that predicting reviews is useful for making a purchase decision using our proposed model better than the rest of the competing methodologies in this paper with helpfulness ratio dependent on rating(star) and SLCABG model performance based on rating level is better than RHRM model which is predicted the helpfulness review based on vote's value when using metrics Accuracy, Precision, Recall, and F1-score as shown in the Table7. Adding the classification performance of the BHRQUT model proposed by our comprehensive TCF, DA, and balanced CNN-BiLSTM also achieved a high performance 97% compared with the commonly used deep learning model.

Furthermore, we investigated how the TCF and DA approaches affected the model's performance under the condition that (avg>=3). Section IV-A.3's classification report parameters and confusion matrix experimental results demonstrated that the proposed performance was higher when classifying helpful reviews. Additionally, users' different tendencies are influenced by subjective such as the quantity of helpfulness ratings in reviews more than users' reviews. We used the balanced CNN-BiLSTM model to categorize the review helpfulness data. Next, we conducted the limitation of the proposed recommendation system to evaluate the recommendation performance. The following are the study's limitations: First, the Amazon review dataset domain was the only one we used. Second, based only on rating scores to categorize the review's quality. Third, we found the condition (ACR >= 4) and (avg>=4) with case (2_2) is a reduction in the performance of the model.

VI. CONCLUSION

In this paper, we propose a novel approach to enhance e-commerce recommendation systems by analyzing the helpfulness of customer reviews using deep learning techniques. First of all, our proposed model collects the four groups of Amazon review datasets (Amazon-5) based on the TCF algorithm which works on differences between users or items tendencies to eliminate the sparsity problem with the relation users or items-based algorithm. Having a balanced dataset, we balanced the datasets by adopting random over-sampling to equally the majority helpful class with the minority unhelpful class. We used NLP techniques to preprocess a summary of the human review attribute related to each product category to prepare the word(token). Having the tokens of each review, the transforming into a dense vector in the embedding layer is done. Finally, the dense vector is created as input to the balanced CNN-BiLSTM model to accurately predict whether a review is helpful by leveraging both local feature extraction and contextual understanding. We have also assessed the effectiveness of the proposed recommendation architecture in this study. First, the CNN (Convolutional Neural Network) part of the model identifies important phrases or patterns in the review text, much like a spotlight that highlights key information. This helps the model focus on significant words or phrases that might indicate helpfulness. Then, the BiLSTM (Bidirectional Long Short-Term Memory) network comes into play, which understands the sequence and context of the entire review. By processing the text in both forward and backward directions, the BiLSTM ensures that the context is fully captured.

We validated the performance of our model using the large samples and and small sample dataset from the Amazon Review dataset. We compared the RMSE, MAE, MSE, of our model with five baseline models from the stae of arte models, including RHRM, AFRAM, machine-generated reviews, DUPIA, and Deep Feature Extractor models. we also assessed Accuracy,Recall, Precision, and F1-Score of our model compared with the (RHRM, and SLCABG) models based max-text length. Our model outperforms these state-of-the-art models for most datasets. Our proposed model demonstrates impressive performance improvements with the best classification result with 97%, 0.161, 0.026,0.038 on all cases of our proposed with(accuracy, RMSE, MSE, MAE) respectively and we get 94% accuracy better than [9] and [32] models, and our results 0.161 RMSE, 0.026 MSE, 0.038 MAE outperform than [9], [21], [22], [24], and [33] models. In addition, according to experimental results, product purchase decisions can be influenced by user preference assessments with useful review information. Recommendation systems can be used effectively in many applications such as Amazon, Alibaba, Netflix, etc. We also aim to enlighten readers by adding the further explanation in terms of its theoretical contribution and practical application.

However, the approach proposed in this paper can only use rating to classify the review as a helpful and unhelpful class, which is not suitable in areas with high requirements for recommendation systems. Therefore, the next step is to use rating and vote in the same condition to get the most accurate recommendations. The performance of the deep learning algorithms of the recommendation system in the field of e-commerce will be improved by suggesting other techniques to enhance its ability to discover reliable recommendations. Future research must thus assess the effectiveness of various deep learning algorithms. The review that was written earlier received more favorable comments than the review that was published later. The dates of the written review should be considered in future studies because they could result in a sequential bias problem.

LIST OF ABBREVIATIONS

NLP	**N**atural **L**anguage **P**rocessing.
CNN	**C**onvolution **N**eural **N**etwork.
BiLSTM	**Bi**directional **L**ong **S**ort **T**erm **M**emory.
LSTM	**L**ong **S**ort **T**erm **M**emory.
TCF	**T**endencies **C**ollaborative **F**Filtering **T**erm **M**emory.
DA	**D**ata **A**ugmentation.
avg	**A**verage.
ACR	**AC**tual **R**ating.
PRR	**PR**edicted **R**ating.
RMSE	**R**oot **M**ean **S**quare **E**rror.
MSE	**M**ean **S**quare **E**rror.
A	**A**ccuracy.
P	**P**recision.
R	**R**ecall.
KNN	**K**- **N**earest **N**eighbors **N**etwork.
SLCABG	**S**entiment **L**exicon **C**onvolutional **A**attention **B**idirectional. Gated Recurrent Unit.
RHRM	**R**eview **H**elpful **R**ecommendation **M**ethodology.
RNN	**R**ecurrent **N**eural **N**etwork.
Deep Feature Extractor	Latent **D**irichlet Allocation + **D**eep **N**eural Network + **M**atrix **F**actorization.

ACKNOWLEDGMENT

All experimental studies were carried out on TITAN XP graphics cards donated by NVIDIA.

REFERENCES

[1] R. J. K. Almahmood and A. Tekerek, "Issues and solutions in deep learning-enabled recommendation systems within the e-commerce field," *Appl. Sci.*, vol. 12, no. 21, p. 11256, Nov. 2022, doi: 10.3390/app122111256.

[2] L. Martin and P. Pu, "Prediction of helpful reviews using emotions extraction," in *Proc. 28th AAAI Conf. Artif. Intell.*, Québec City, QC, Canada, vol. 28, no. 1, 2014, pp. 1551–1557.

[3] R. Hayes and A. Downie, "What is e-commerce," IBM Topics, IBM, Feb. 2024. [Online]. Available: https://www.ibm.com/topics/ecommerce

[4] Y. Zhu, M. Liu, X. Zeng, and P. Huang, "The effects of prior reviews on perceived review helpfulness: A configuration perspective," *J. Bus. Res.*, vol. 110, pp. 484–494, Mar. 2020.

[5] S.-M. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti, "Automatically assessing review helpfulness," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2006, p. 423.

[6] R. A. Hendrawan, E. Suryani, and R. Oktavia, "Evaluation of e-commerce product reviews based on structural, metadata, and readability characteristics," *Proc. Comput. Sci.*, vol. 124, pp. 280–286, Jan. 2017.

[7] K. Kaushik, R. Mishra, N. P. Rana, and Y. K. Dwivedi, "Exploring reviews and review sequences on e-commerce platform: A study of helpful reviews on Amazon.in," *J. Retailing Consum. Services*, vol. 45, pp. 21–32, Nov. 2018.

[8] K. Xue and J. Wang, "Collaborative filtering recommendation algorithm for user interest and relationship based on score matrix," in *Proc. Int. Conf. Math., Modeling, Simulation Algorithms (MMSA)*. Amsterdam, The Netherlands: Atlantis Press, 2018, pp. 217–221.

[9] Q. Li, X. Li, B. Lee, and J. Kim, "A hybrid CNN-based review helpfulness filtering model for improving e-commerce recommendation service," *Appl. Sci.*, vol. 11, no. 18, p. 8613, Sep. 2021.

[10] M. Martin, "Predicting ratings of Amazon reviews: Techniques for imbalanced datasets," M.S. thesis, Univ. Liège, Liège, Belgium, 2017.

[11] Z. Ahmed and J. Wang, "A fine-grained deep learning model using embedded-CNN with BiLSTM for exploiting product sentiments," *Alexandria Eng. J.*, vol. 65, pp. 731–747, Feb. 2023, doi: 10.1016/j.aej.2022.10.037.

[12] N. Shrestha and F. Nasoz, "Deep learning sentiment analysis of Amazon.com reviews and ratings," *Int. J. Soft Comput., Artif. Intell. Appl.*, vol. 8, no. 1, pp. 1–15, Feb. 2019.

[13] F. Cacheda, V. Carneiro, D. Fernández, and V. Formoso, "Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems," *ACM Trans. Web*, vol. 5, no. 1, pp. 1–33, Feb. 2011.

[14] G. Kim, I. Choi, Q. Li, and J. Kim, "A CNN-based advertisement recommendation through real-time user face recognition," *Appl. Sci.*, vol. 11, no. 20, p. 9705, Oct. 2021.

[15] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for YouTube recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 191–198.

[16] M. Faisal, A. Hameed, and A. S. Khattak, "Recommending movies on user's current preferences via deep neural network," in *Proc. 15th Int. Conf. Emerg. Technol. (ICET)*, Dec. 2019, pp. 1–6.

[17] S. Ge, T. Qi, C. Wu, F. Wu, X. Xie, and Y. Huang, "Helpfulness-aware review based neural recommendation," *CCF Trans. Pervasive Comput. Interact.*, vol. 1, no. 4, pp. 285–295, Dec. 2019.

[18] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, Feb. 2017, pp. 425–434.

[19] S. Krishnamoorthy, "Linguistic features for review helpfulness prediction," *Expert Syst. Appl.*, vol. 42, no. 7, pp. 3751–3759, May 2015.

[20] C. Chen, Y. Yang, J. Zhou, X. Li, and F. S. Bao, "Cross-domain review helpfulness prediction based on convolutional neural networks with auxiliary domain discriminators," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 602–607.

[21] W. Li and B. Xu, "Aspect-based fashion recommendation with attention mechanism," *IEEE Access*, vol. 8, pp. 141814–141823, 2020.

[22] S. Ouyang and A. Lawlor, "Improving explainable recommendations by deep review-based explanations," *IEEE Access*, vol. 9, pp. 67444–67455, 2021.

[23] J. Kim, J. Park, M. Shin, J. Lee, and N. Moon, "The method for generating recommended candidates through prediction of multi-criteria ratings using CNN-BiLSTM," *J. Inf. Process. Syst.*, vol. 17, no. 4, pp. 707–720, 2021.

[24] X. Zhang, H. Liu, X. Chen, J. Zhong, and D. Wang, "A novel hybrid deep recommendation system to differentiate user's preference and item's attractiveness," *Inf. Sci.*, vol. 519, pp. 306–316, May 2020.

[25] M. Kshour, M. Ebrahimi, S. Goliaee, and R. Tawil, "New recommender system evaluation approaches based on user selections factor," *Heliyon*, vol. 7, no. 7, Jul. 2021, Art. no. e07397.

[26] Y. Zhang and L. Zhang, "Movie recommendation algorithm based on sentiment analysis and LDA," *Proc. Comput. Sci.*, vol. 199, pp. 871–878, Jan. 2022.

[27] M. Jugovac, D. Jannach, and L. Lerche, "Efficient optimization of multiple recommendation quality factors according to individual user tendencies," *Expert Syst. Appl.*, vol. 81, pp. 321–331, Sep. 2017.

[28] N. Purnawirawan, P. De Pelsmacker, and N. Dens, "Balance and sequence in online reviews: How perceived usefulness affects attitudes and intentions," *J. Interact. Marketing*, vol. 26, no. 4, pp. 244–255, Nov. 2012.

[29] C. Zhang, G. Wang, Y. Zhou, and J. Jiang, "A new approach for imbalanced data classification based on minimize loss learning," in *Proc. IEEE 2nd Int. Conf. Data Sci. Cyberspace (DSC)*, Jun. 2017, pp. 82–87.

[30] P. M. Vieira and F. Rodrigues, "An automated approach for binary classification on imbalanced data," *Knowl. Inf. Syst.*, vol. 66, no. 5, pp. 2747–2767, May 2024.

[31] A. S. Alhanaf, M. Farsadi, and H. H. Balik, "Fault detection and classification in ring power system with DG penetration using hybrid CNN-LSTM," *IEEE Access*, vol. 12, pp. 59953–59975, 2024.

[32] L. Yang, Y. Li, J. Wang, and R. S. Sherratt, "Sentiment analysis for e-commerce product reviews in Chinese based on sentiment lexicon and deep learning," *IEEE Access*, vol. 8, pp. 23522–23530, 2020.

[33] B. M. Shoja and N. Tabrizi, "Customer reviews analysis with deep neural networks for e-commerce recommender systems," *IEEE Access*, vol. 7, pp. 119121–119130, 2019, doi: 10.1109/ACCESS.2019.2937518.

[34] S. Datta, J. Das, P. Gupta, and S. Majumder, "SCARS: A scalable context-aware recommendation system," in *Proc. 3rd Int. Conf. Comput., Commun., Control Inf. Technol. (C3IT)*, Feb. 2015, pp. 1–6.

[35] M. Bach, A. Werner, J. Żywiec, and W. Pluskiewicz, "The study of under- and over-sampling methods' utility in analysis of highly imbalanced data on osteoporosis," *Inf. Sci.*, vol. 384, pp. 174–190, Apr. 2017.

[36] J. Lv, B. Song, J. Guo, X. Du, and M. Guizani, "Interest-related item similarity model based on multimodal data for top-N recommendation," *IEEE Access*, vol. 7, pp. 12809–12821, 2019.

RAND ALMAHMOOD was born in Iraq, in March 1981. She received the B.S. degree in computer science from the Department of Mathematics, University of Thi-Qar, Iraq, in 2001, and the M.S. degree in software engineering from the Politehnica University of Bucharest, Bucharest, Romania, in 2017. She is currently pursuing the Ph.D. degree with the Department of Computer Engineering, Gazi University, Ankara, Türkiye. From 2004 to 2016, she was a Research Assistant with the Computer Laboratory, Faculty of Engineering. From 2017 to 2019, she was an Assistant Lecturer with the Department of Oil Engineering, College of Engineering, University of Thi-Qar. She prepared many computer technology courses for university employees and a number of workshops, and participated in scientific conferences held at the University of Thi-Qar. She has skills and expertise, such as software development, object-oriented programming, java programming, web development, SQL, software programming, and C++.

MUHAMMED MUTLU YAPICI received the M.Sc. degree from the Department of Computer and Electronic Education, Informatics Institute, Gazi University, Türkiye, in 2012, and the Ph.D. degree from the Department of Computer Engineering, Gazi University.

From 2014 to 2021, he was a Lecturer with the Department of Computer Technologies, Ankara University, Türkiye. From 2022 to 2023, he was a Visiting Scholar with the School of Computing, Newcastle University, U.K. Currently, he is an Assistant Professor with the Department of Computer Technologies, Ankara University. His research interests include artificial intelligence, machine learning, deep learning, image processing, optimization techniques, and their applications in medical research, signature verification, and object recognition.

ADEM TEKEREK received the M.Sc. degree from the Department of Electronics and Computer Education, where he developed a web content management system for enterprise structures, and the Ph.D. degree from the Department of Electronics and Computer Education, Gazi University, in 2016. His Ph.D. thesis was on the implementation of a new real-time web application firewall algorithm for web-based intrusion prevention. He is currently an Associate Professor with the Department of Computer Engineering, Faculty of Technology, Gazi University. He has made significant contributions to the academic literature with his publications in leading journals, such as *Expert Systems with Applications*, *Computers*, *Security*, and *Wireless Personal Communications*. He has taught courses, such as database management systems, data structures and algorithms, software engineering, artificial intelligence, business intelligence, data analysis, and game programming. His research interests include software development, artificial intelligence, machine learning, deep learning, data mining, information security, content management systems, distance education, AI applications in healthcare, assistive technology, and software architecture. He is skilled in multiple programming languages, such as database management, web design, and operating systems. He is involved in Erasmus+ projects. He is also a member of several professional and academic organizations and contributes to youth education and cultural activities. He has received publication incentive awards from Gazi University and TÜBİTAK.